

tional imaging of the motor system that bring to bear novel mathematical techniques and extends the scope of functional imaging experimentation.

Mapping the Formation of Internal Models

The notion of internal model formation predicting future motor requirements has emerged as a dominant concept from *in vivo* studies of the human brain with functional neuroimaging. Reaching, grasping, and tracking objects requires the construction and execution of an internal model of the movements needed for performing the action (Desmurget and Grafton, 2000; Imamizu et al., 2000). Neuroimaging studies of visuomotor tracking, for instance, have been particularly useful for elucidating the development of internal model formation. Several such studies have identified significant changes in regional activity in a network of regions including primary and supplementary motor cortices, basal ganglia, and cerebellum during motor tasks (Grafton, Fagg, and Arbib, 1998; Grafton, Hazeltine, and Ivry, 1998; Turner et al., 1998). Each of these brain areas is directly involved in carrying out motor movements. Turner and colleagues (1998), however, examined rCBF PET images obtained while subjects moved a handheld joystick to track the movement of a target at three different rates of sinusoidal movement. Increases in rCBF during arm movement (relative to an eye tracking only baseline condition) were seen in a distributed pattern of regions, including primary sensorimotor, dorsal and mesial premotor, and dorsal parietal cortices in the left hemisphere and, though not as prominently, the sensorimotor and superior parietal cortices in the right hemisphere. Subcortical activations were observed in left putamen, globus pallidus, and thalamus, in the right basal ganglia, and in the right anterior cerebellum. Left primary motor, left globus pallidus, and right anterior cerebellum had changes in rCBF that correlated positively with the rate of movement. A particularly unique finding was the activation of the globus pallidus with increasing movement velocity. On the other hand, studies of Parkinsonian patients (Nakamura et al., 2001) have revealed increased patterns of rCBF activity in these regions, suggesting a compensation for defective basal ganglia functioning and a failure to correct errors online. This supports the notion that the basal ganglia motor circuit may be involved preferentially in controlling or monitoring the scale and/or dynamics of limb movements needed to minimize movement error. These findings hint at interdependency between distributed brain areas needed for online correction of motor errors leading to internal movement representation.

The Emergence of Motor Automaticity

As subjects gain increased experience with motor tasks they typically display continued improvement in motor execution until those movements have become automatic. In general, motor automaticity is most likely to occur in tasks where performance errors may be readily anticipated and corrected online. Brain imaging studies have demonstrated differential changes in activity in limb motor areas during early motor skill learning, consistent with functional reorganization occurring at the level of motor output. Extensive practice and the emergence of skill automaticity resulted in decreases in the amount of activity in motor and SMA, accompanied by increases in activity in inferior parietal cortex as well as in basal ganglia. These alterations may be further modified over time presumably due to neuronal efficiency and optimization. Therefore, internal models of movements and movement automaticity are tightly linked.

On a simple level, automaticity in the motor system may be indirectly measured when contrasting performance of a motor task using the dominant and nondominant hands. Nondominant hand

movements, perhaps being less automatic, appear to require greater cortical BOLD signal activity similar to complex tasks with the dominant hand, and result in greater activation of ipsilateral cortical motor areas and striatum. However, automaticity may be more rigorously examined and manipulated through the use of sensorimotor compatibility task paradigms. Experiments by Grafton, Salidis, and Willingham (2001), for example, assessed motor learning under compatible and incompatible perceptual-motor conditions to identify brain areas involved in different perceptual-motor transformations. Subjects tracked a continuously moving target that moved in a repeating sequence embedded within random movements to block sequence awareness. Psychophysical studies of behavioral transfer from incompatible (joystick and cursor moving in opposite directions) to compatible tracking established that incompatible learning was occurring with respect to target location. rCBF imaging during compatible learning identified increasing activity throughout the precentral gyrus, maximal in the arm area. Incompatible learning also led to increasing rCBF activity in the precentral gyrus, maximal in the putative frontal eye fields. When the incompatible task was switched to a compatible response and the previously learned sequence was reintroduced, there was an increase in activation of the arm region of the motor cortex. These findings indicate that learning-related increases of brain activity leading to motor automaticity are dynamic, with recruitment of multiple motor output areas, contingent on task demands.

Feedback Monitoring

The cerebellum appears to play a critical role in the coordination of movement, being essential in the processing of motor feedback (see CEREBELLUM AND MOTOR CONTROL). Miall, Reckess, and Imamizu (2001) assessed cerebellar involvement using fMRI during visually guided tracking tasks requiring varying degrees of eye-hand coordination. BOLD signal in the cerebellum indicated greater activity during independent rather than coordinated eye and hand tracking. In subsequent tasks, they observed parametric increases in cerebellar activity as eye-hand coordination increased. This demonstrates a nonmonotonic relationship of the cerebellar BOLD signal with tracking performance, showing high activity during both coordinated and independent conditions. In another example, using $H_2^{15}O$ PET, Blakemore, Frith, and Wolpert (2001) examined neural responses to parametrically varied degrees of discrepancy between the predicted and actual sensory consequences of movement. Subjects used their right hand to move a robotic arm. The motion of this robotic arm determined the position of another robotic arm, which made contact with the palm of the subject's left hand. Using this interface, computer-controlled delays were introduced between the movement of the right hand and the tactile stimulation on the left. Activity in the right lateral cerebellar cortex was positively correlated with stimulation delay. These data provide provocative evidence that the cerebellum plays a key role in signaling the sensory discrepancy between the predicted and actual sensory consequences of movements, supporting motor coordination.

Sensorimotor Integration

Sensorimotor integration is the process by which sensory input and motor output signals are combined to provide an internal estimate of the state of both the world and one's own body. Although a single perceptual and motor snapshot can provide information about the current state, computational models show that the state can be optimally estimated by an iterative process in which an internal estimate is maintained and revised by the current sensory and motor signals (see SENSORIMOTOR LEARNING). These theoretical models predict that an internal state system is, indeed, stored

in the brain. Reports on patients with lesions of the superior parietal lobe have shown both sensory and motor deficits consistent with an inability to maintain such an internal representation between updates (Wolpert, Goodbody, and Husain, 1998). Such behavioral findings predict that the superior parietal lobe is critical for sensorimotor integration, by maintaining an internal representation of the body's state.

Neuroimaging studies, too, have lent support to this notion. Grafton and co-workers (1992) studied visually guided movements subjects performing visuomotor tracking tasks during PET. Tracking a moving target with the index finger showed a network of focal responses of rCBF observed in the primary motor cortex, dorsal parietal cortex, precuneate cortex, SMA, and ipsilateral anterior cerebellum relative to visual tracking alone. When the temporal complexity of the tracking task was altered by introducing a "no-go" contingency that allowed for greater time for movement preparation, there was a significant increase of rCBF in the SMA. When the spatial complexity was altered by adding a secondary target that provided directional cues for the primary target, there were additional significant increases of rCBF in bilateral dorsal parietal cortex and precuneus. Performing the tracking task with different body parts produced somatotopically distributed responses in only the motor cortex. The results of this study suggest that the SMA plays a role in the sequencing of movements and that medial and dorsal parietal cortices participate in the integration of spatial attributes during movement selection.

Measuring and Modeling the Motor System

PET and fMRI studies have provided ample evidence that activation of motor cortices and online movement error correction repeated over time result in action automaticity and sensorimotor integration. Initial attempts at movements result in widespread activation when the brain is drawing upon numerous systems to approximate the required movement, force, velocity, etc. Successive presentations of the same movements offer the motor system opportunity to tune internal model parameters on the basis of behavioral error and subsequent model updates. A process of reinforcement learning and iterative motor control optimization in this vein has been discussed extensively (see REINFORCEMENT LEARNING IN MOTOR CONTROL). During this process, as the roles of supporting brain systems appear to be no longer required, the pattern of observed activation may be altered, diminishing in extent as the model becomes more accurate, integrated, and automatic. Ultimately, only a minimal set of brain regions necessary for carrying out the needed movement indicate BOLD response activation. Therefore, areas previously involved in the tuning of internal model parameters are now free to devote their neural resources to other cognitive problems.

From a computational point of view, this process resembles that of a control optimization problem (see IDENTIFICATION AND CONTROL). Initially, the parameters governing the system are ill-defined. But through an iterative process of taking errors into account, future errors are minimized as the contribution to performance accuracy from some parameters are minimized and those of others accentuated. At which time, the minimal number of parameters have been identified that minimize overall system error in the presence of system noise (i.e., model equilibrium), thereby indicating that the model has been augmented from that of a purely causal model to that of one also having the ability to forecast the next model state and required motor output (see OPTIMIZATION PRINCIPLES IN MOTOR CONTROL). This, then, forms what is often more broadly referred to in the motor control literature as a "forward model" (Desmurget and Grafton, 2000). From signal processing, such systems may be used to predict system behavior in which the parameter estimation problem involves the identification of com-

plex poles and zeros, along with system gain terms (see also FORECASTING). For example, a simplistic, finite impulse response (FIR) model is presented in Figure 2. This could be used to simulate the forward modeling of visuomotor task performance of the limb, in the presence of system noise, through the use of a switched connectivity between ocular and motor mechanisms.

To achieve the richness of functional imaging data necessary for such modeling purposes, greater sophistication in behavioral paradigms used in the scanning environment is needed. Such paradigms will involve an increased utilization of finely sampled behavioral measurements of motor speed, acceleration and higher derivatives. For example, behavioral paradigms such as that employed by Novak, Miller, and Houk (2000) might be studied with fMRI to measure the neural concomitants of rapid hand and joint movements. These authors attempted to identify overlapping submovements during a rotational target capture task by examining the zero crossings of subject acceleration traces and its derivatives, jerk and snap (the third and fourth derivatives, respectively). Movements without overlapping submovements had, on average, near symmetric, bell-shaped velocity profiles that were independent of speed and consistent with a theoretical minimum jerk velocity

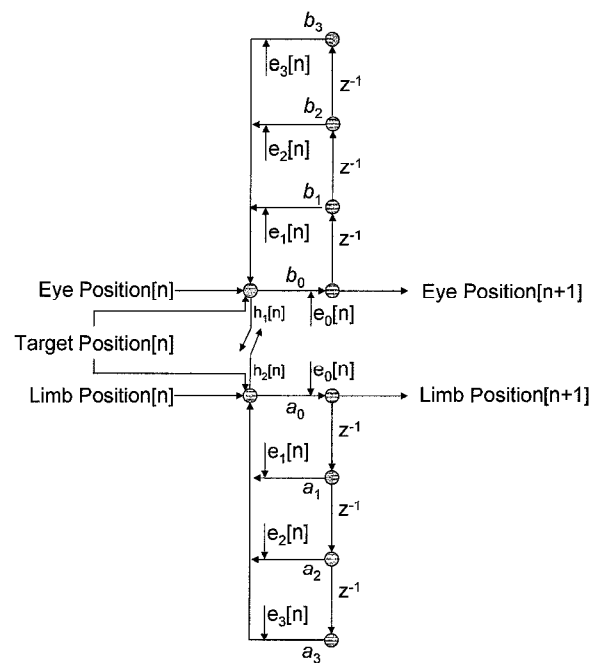


Figure 2. Visuomotor tracking may be modeled computationally using predictive FIR models. This figure shows a FIR model that incorporates separate, butterfly components for visual—and for motor—tracking, connected via a switched line tap. This permits both halves of the model to work independently, when the switch is open, or together using an appropriate cross-system transfer function (h), when the switch is closed. The current state of the limb position enters into the system, is scaled, delayed accordingly using discrete complex exponential delay terms (e.g., $\exp(-i2\pi/N) = z^{-1}$, where $i = \sqrt{-1}$) and aggregated to form the output for the next time step. System error is present at each delay term, which when the parameter estimates have not reached equilibrium may dominate system output. Linear systems theory combined with methods for the analysis of functional and effective connectivity from neuroimaging data permit the construction of dynamic models of brain function in visuomotor control as well as other neural systems. Additionally, these models may be useful in the synthetic simulation of fMRI data as has been previously with PET (see SYNTHETIC FUNCTIONAL BRAIN MAPPING).

model. The authors propose a nonlinearly dampened mass-spring (second-order derivative) model of the wrist as a suitable model governing knob turning. Motor tasks paradigms like this, or ones that utilize a joystick, trackball, or other continuous input device permitting estimation of higher order derivatives, when conducted using whole brain fMRI would help in understanding what brain regions are dynamically involved in the construction of internal models and how they may relate to limb kinematics.

In pursuing this line of investigation, however, new thinking in fMRI experimental methods is needed. Unlike epoch- or event-related paradigms, the use of continuous performance fMRI holds considerable promise for investigation of the dynamic process of motor functioning (Figure 3). Most important in this neuroimaging framework is the performance of the subject in relation to maintaining positional, velocity, and acceleration accuracy rather than the presence or absence of perceptual stimuli. For example, Figure 4 shows the results of a reduced GLM model analysis containing only subject-generated tracking variables indicating significant effect involving visual areas (V1 and V2), cerebellum, primary and supplementary motor cortex with absolute target position; visual areas (V1 and V2), cuneus, and superior frontal gyrus with target velocity; and superior frontal and cingulated gyri with target ac-

celeration. Activation of these visual areas has been previously implicated in attentional networks (Friston and Buchel, 2000; see also VISUAL ATTENTION), underscoring the possible role for these components in visuomotor tracking performance. Subject positional error was significantly correlated with visual areas as well as activation in DLPFC, suggesting a role for the frontal cortex in the organization of action (see PREFRONTAL CORTEX IN TEMPORAL ORGANIZATION OF ACTION). The velocity of subject positional error was significantly correlated with activity in primary motor region, consistent with the aforementioned results of Turner and co-workers (1998). These findings clearly indicate a specific collection of dynamically involved brain regions correlating with target position, velocity, acceleration, and subject-generated variables. Continued investigations of this type can examine more closely how the variables pertaining to visuomanual tracking performance are optimized by the formation of internal models of continuous movement, for instance, over a period of several weeks.

Many cognitive and behavioral models born out of human brain imaging data are often focused on the isolated brain areas associated with statistically high blood flow or BOLD signal response, rather than analyzing the cooperative computation between multiple brain regions. By contrast, the analysis of functional connec-

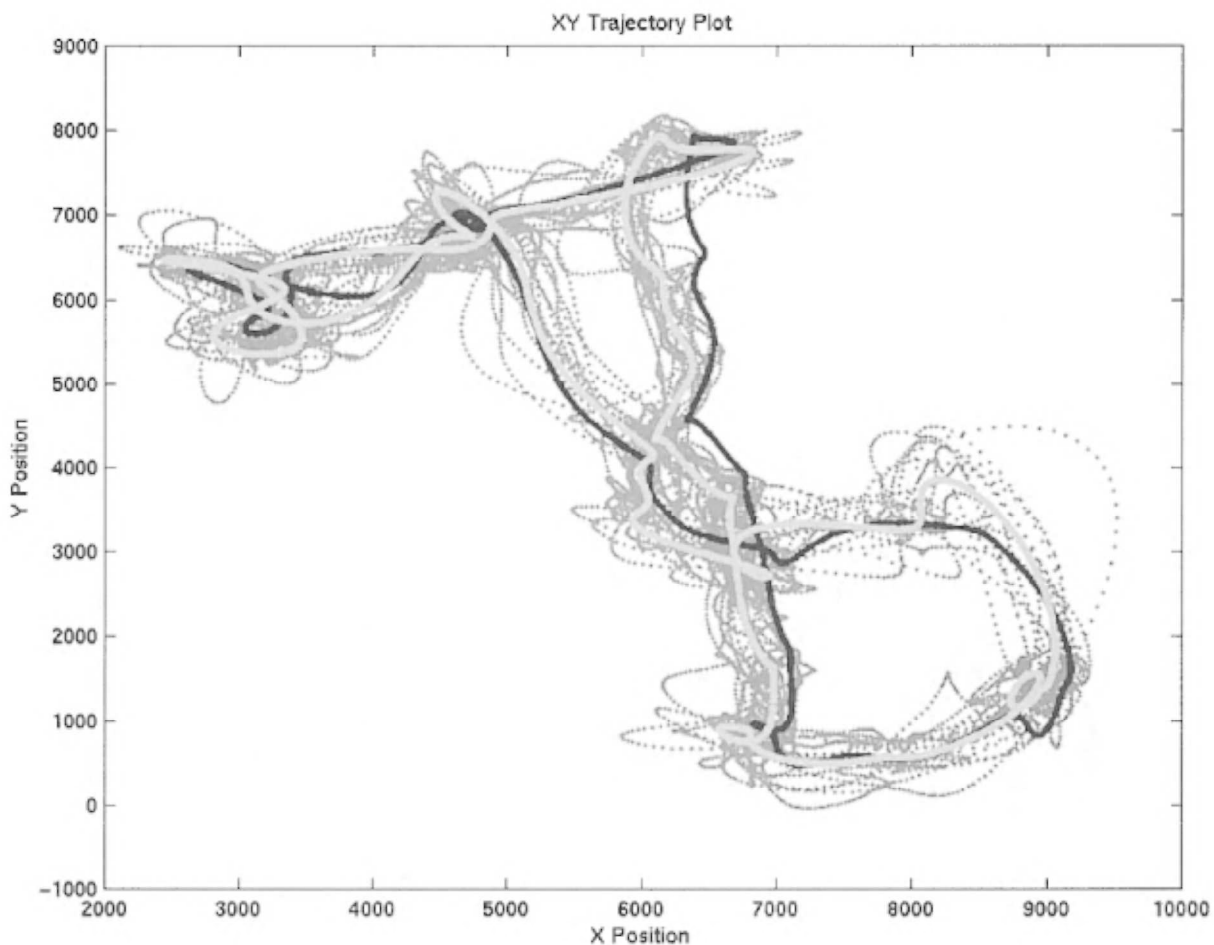


Figure 3. Visuomotor pursuit tracking performance obtained during fMRI in an example subject. The target trajectory presented here was constructed using a 32-point-based complex Fourier spectra having randomized phase-components. Subjects performed six versions of this trajectory, in which four were rotated versions of the same trajectory (0° , $+90^\circ$, 180° , and -90°); one run was a repeat of the 0° rotated trajectory but in which a 10-

time-point temporal lag was imposed on the joystick cursor; and, finally, a different trajectory, comprised of the same frequency magnitudes but having randomized phase relative to the other trajectories. The light gray dotted line represents 16 cycles of an example subject's performance, the black line represents the trajectory followed by the target, and the medium gray line the subject's mean pursuit trajectory taken over the 16 cycles.

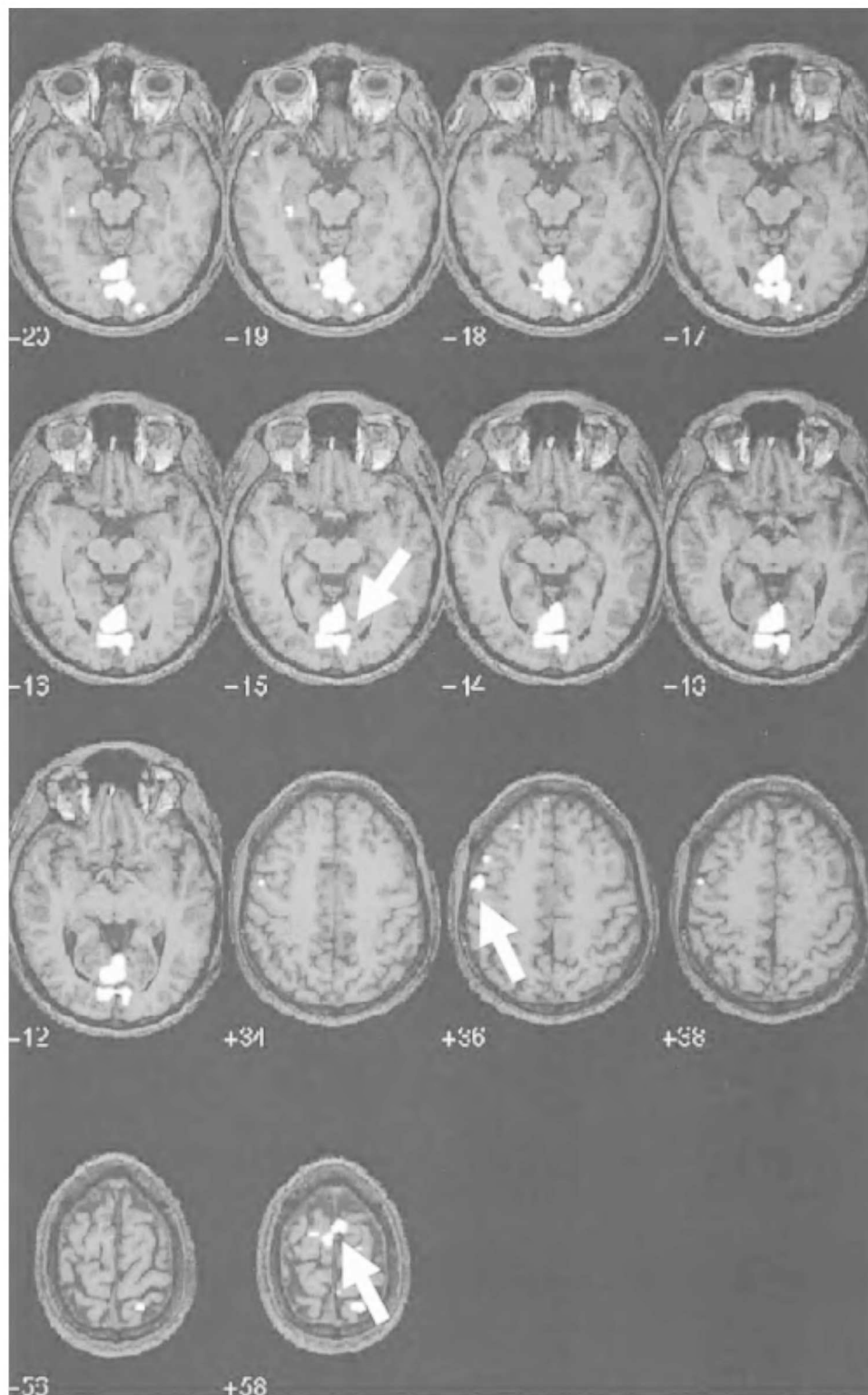


Figure 4. Talairach-space overlay plot of the regression of all continuous visuomotor tracking variables (Reduced Model $F(12, 823)$, $p \leq 0.005$, uncorrected) for an example subject after the removal of run-to-run, linear trend, eye movement, and physiological effects. An all-plastic fiberoptic joystick was specially fabricated for use in the MRI scanner. Subject visuomotor tracking performance, eye position, heart rate, and respiration was continuously measured during collection of BOLD EPI time series (General Electric Horizon 1.5T scanner, TR = 2000 ms, TE = 500 ms, FOV = 24 cm). Scanner and task timing were synchronized using the acquisition of the MR scanner unblank TTL signal from which slice acquisition information was obtained. This information was used to sort both the subject's tracking performance data, as well as physiological monitoring and eye tracking data, into a slice-based experimental design matrix. Additional variates were included in the design matrix to account for run-to-run shifts in baseline as well as within-run linear trends. Voxels from each slice were then subjected to linear regression via the GLM and effects were tested against the Wilk's Lambda criterion and converted to F-statistics. A reduced regression model of only those variables related to subject task performance was obtained after regressing out the effects of physiological, eye tracking, and run-to-run effects. Principle effects of the performance-related variables alone are noted in visual and motor cortices (arrows) as well as in middle frontal gyrus. These results demonstrate the successful activation of principle neural systems during continuous visuomotor tracking in fMRI. Further detail on the relative roles of each of the individual performance variables is the subject of a manuscript in preparation.

tivity provides insight into the functional relationships between distributed brain areas (see COVARIANCE STRUCTURAL EQUATION MODELING). Figure 5 shows a three-dimensional representation of the pattern of inter-regional correlations between BOLD time course activity during visuomotor tracking, indicative of strong

connectivity between visual, motor, and subcortical regions. Such strong interaction between these areas would be expected from current models of internal model formation. The combination of functional connectivity modeling methods for neuroimaging may be combined with techniques for forecasting system behavior thereby

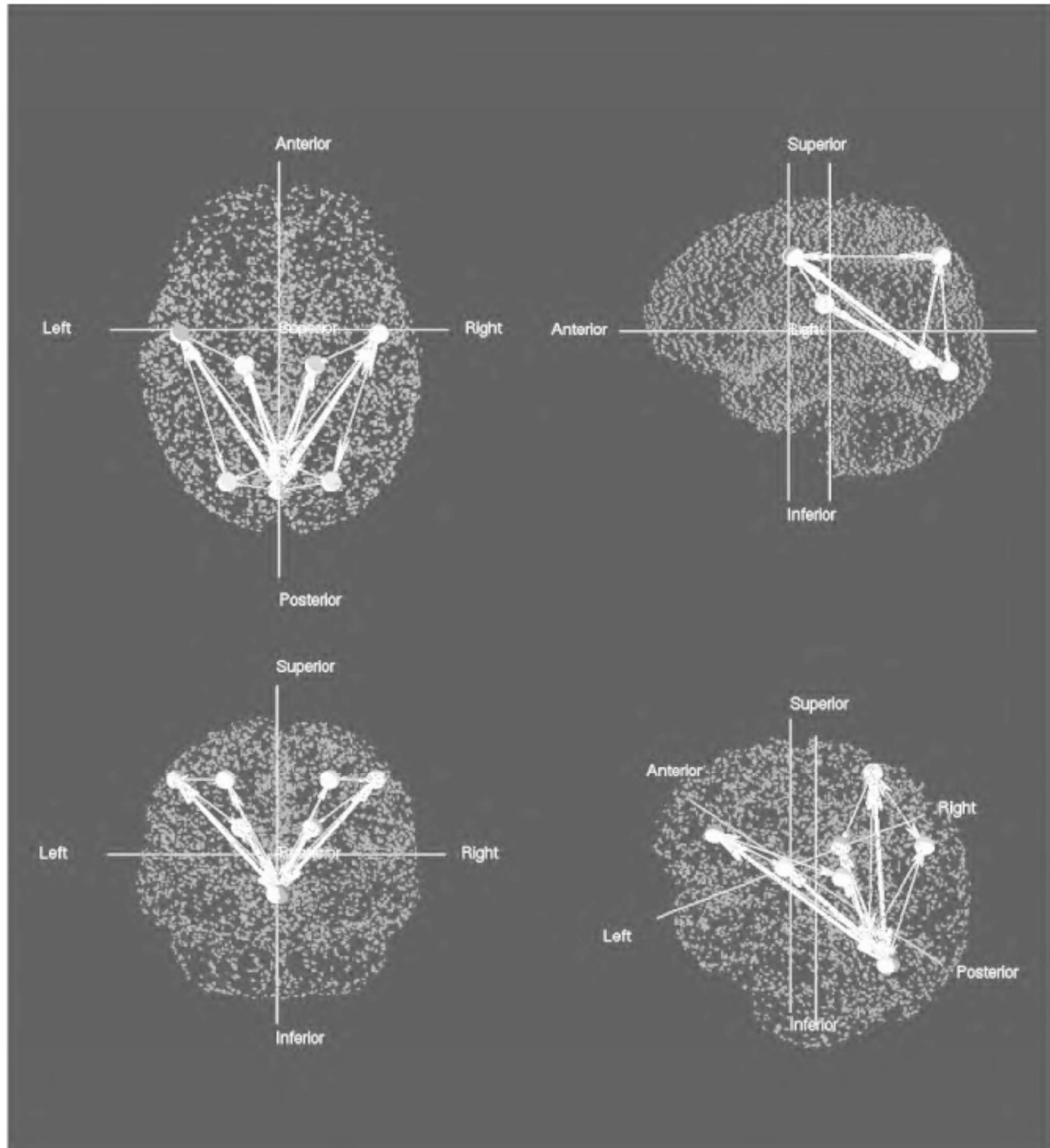


Figure 5. In this figure, multiple views (axial, sagittal, coronal, and perspective) of brain regions identified during continuous visuomotor tracking are shown as spheroids in 3D standardized Talairach space. The strength of the correlation is indicated by line thickness and no connectivity is evident where a path connecting two brain regions is not present. The resulting paths indicate bidirectional paths connecting the visual regions, subcortical, parietal, and primary motor regions, with stronger correlations existing between primary visual and motor regions as well as from visual to subcortical regions. Further decomposition of the correlational structure between regions (PCA, structural equation modeling, etc.) can be used to identify the independent contributions that each connection contributes to the region-wise covariance matrix. Inclusion of causal and anticausal lag terms (e.g., as in Figure 2) increases the generality of the model giving it a temporal

component similar to that of the hemodynamic lag between stimulus presentation and the BOLD response in fMRI. Moreover, in a connected network it is often useful to identify optimal routes between nodes by which to propagate information through the system. Analytical techniques are available for the assessment of the minimum-cost path (the path in which information loss is minimal or that propagation delay is smallest) between a node in a connected network to other surrounding nodes. For example, Dijkstra's minimum-cost path algorithm can be employed to examine the minimum-cost path from V1 to, for instance, the primary motor cortex. The minimum cost path may be further constrained by restricting the number of other regions (nodes) through which it must pass (e.g., the number of hops).

providing an empirically based model for the temporal characteristics of inter-regional connectivity involved during tracking and online error correction. Since the resulting FIR-connectivity model includes temporal delay terms, both causal and anticausal, then feedback and feedforward connections may be estimated. The examination of how the strength of connectivity is altered between regions over time as internal models of motor behavior are formed would also be possible. The analysis of how signals are propagated through the network would identify optimum paths of information flow (see figure caption for discussion). In this manner, the responsibilities of the parietal lobe and cerebellum (Desmurget et al., 2001), as well as the role the basal ganglia (see BASAL GANGLIA), in the formation of internal models and automaticity could be assessed via this modeling process.

Conclusions

The framework of online error correction, automaticity, and integration, gaining support from in vivo functional brain imaging, is helping to explain changes in motor system activation magnitude and spatial extent often accompanying practice and increased skill. In visuomotor tracking, in particular, it is difficult to envision how such behavioral alterations can result in the absence of a predictive internal model. Empirical results from brain imaging have given credence to previously postulated theoretical models of visuomotor coordination (Gauthier et al., 1988) that anticipated such dynamic interaction between motor and visual systems. One can readily expect that the levels of sophistication for examining the domain of the motor system using neuroimaging will continue to improve. In presenting these key examples from the field of neuroimaging, it is clear that there has been and will continue to be much gained from studies imaging the motor brain.

Acknowledgments. The author is grateful to Dr. Scott T. Grafton for his comments on earlier versions of this chapter. This work was funded by a grant from the National Science Foundation (NSF 01-41, 0121905).

Road Maps: Cognitive Neuroscience; Mammalian Motor Control

Related Reading: Hemispheric Interactions and Specialization; Imaging the Grammatical Brain; Imaging the Visual Brain; Statistical Parametric Mapping of Cortical Activity Patterns; Synthetic Functional Brain Mapping

References

- Blakemore, S. J., Frith, C. D., and Wolpert, D. M., 2001, The cerebellum is involved in predicting the sensory consequences of action, *Neuroreport*, 12(9):1879–1884.
- Desmurget, M., and Grafton, S., 2000, Forward modeling allows feedback control for fast reaching movements, *Trends Cogn. Sci.*, 4(11):423–431. ♦
- Desmurget, M., Grea, H., Grethe, J. S., Prablanc, C., Alexander, G. E., and Grafton, S. T., 2001, Functional anatomy of nonvisual feedback loops during reaching: A positron emission tomography study, *J. Neurosci.*, 21(8):2919–2928.
- Friston, K. J., and Buchel, C., 2000, Attentional modulation of effective connectivity from V2 to V5/MT in humans, *Proc. Natl. Acad. Sci. USA*, 97(13):7591–7596.
- Gauthier, G. M., Vercher, J. L., Mussa-Ivaldi, F. A., and Marchetti, E., 1988, Oculo-manual tracking of visual targets: Control learning, coordination control and coordination model, *Exp. Brain Res.*, 73:127–137. ♦
- Grafton, S. T., Fagg, A. H., and Arbib, M. A., 1998, Dorsal premotor cortex and conditional movement selection: A PET functional mapping study, *J. Neurophysiol.*, 79(2):1092–1097.
- Grafton, S. T., Hazeltine, E., and Ivry, R. B., 1998, Abstract and effector-specific representations of motor sequences identified with PET, *J. Neurosci.*, 18(22):9420–9428. ♦
- Grafton, S. T., Mazziotta, J. C., Woods, R. P., and Phelps, M. E., 1992, Human functional anatomy of visually guided finger movements, *Brain*, 115(Pt 2):565–587.
- Grafton, S. T., Salidis, J., and Willingham, D. B., 2001, Motor learning of compatible and incompatible visuomotor maps, *J. Cogn. Neurosci.*, 13(2):217–231.
- Imamizu, H., Miyauchi, S., Tamada, T., Sasaki, Y., Takino, R., Putz, B., Yoshioka, T., and Kawato, M., 2000, Human cerebellar activity reflecting an acquired internal model of a new tool, *Nature*, 403(6766):192–195.
- Miall, R. C., Reckess, G. Z., and Imamizu, H., 2001, The cerebellum coordinates eye and hand tracking movements, *Nat. Neurosci.*, 4(6):638–644. ♦
- Nakamura, T., Ghilardi, M. F., Mentis, M., Dhawan, V., Fukuda, M., Hacking, A., Moeller, J. R., Ghez, C., and Eidelberg, D., 2001, Functional networks in motor sequence learning: abnormal topographies in Parkinson's disease, *Hum. Brain. Mapp.*, 12(1):42–60.
- Novak, K. E., Miller, L. E., and Houk, J. C., 2000, Kinematic properties of rapid hand movements in a knob turning task, *Exp. Brain Res.*, 132(4):419–433. ♦
- Turner, R. S., Grafton, S. T., Votaw, J. R., Delong, M. R., and Hoffman, J. M., 1998, Motor subcircuits mediating the control of movement velocity: A PET study, *J. Neurophysiol.*, 80(4):2162–2176. ♦
- Wolpert, D. M., Goodbody, S. J., and Husain, M., 1998, Maintaining internal representations: the role of the human superior parietal lobe, *Nat. Neurosci.*, 1(6):529–533. ♦

Imaging the Visual Brain

Robert L. Savoy

Introduction

This article describes some of the progress made in functional brain imaging of visual processes and highlights the challenges and opportunities for future progress, especially in the context of testing models and theories. A few key attributes of modern brain-imaging tools are compared. A subset of the many applications to visual processing is described. The primary technology used in the selected studies is functional magnetic resonance imaging (fMRI), which is currently the dominant volumetric imaging technology for studying human brain function. But the emphasis is to describe, independent of the imaging technology, the relevance of particular experimental designs to problems in visual perception and associated issues in higher-level (i.e., necessarily top-down) processing of visual information.

The key features of modern brain-imaging tools as presented here are those features that determine the potential strengths and weaknesses of each tool when the tool is applied to problems in human brain theory. These features include current and likely future limits in spatial and temporal resolution, constraints on subject participation, and trade-offs in experimental design. Within a given imaging modality, such attributes of the various imaging tools are not independent of each other. Lesion data will be mentioned briefly, but as it is not a usable experimental tool with humans, it will not be emphasized, despite its great theoretical and historical importance.

Functional imaging tools have been applied to a wide range of problems associated with low-level and higher-level visual processing. This article will focus on retinotopy (a low-level aspect of

the brain's visual architecture), visual motion perception and visual object representation (intermediate-level aspects of vision), and voluntary modulation of attention and visual imagery (higher-level processes that necessarily have top-down components). This list does not come close to covering the full range of applications to the visual system, but it emphasizes some of the areas where modeling and brain theory might be testable using current imaging tools.

Imaging Technologies

Functional imaging technologies can be divided into three categories based on the kind of physical phenomena they can measure. In one group are technologies (EEG, MEG, TMS/rTMS, intracranial electrode recordings) that measure, disrupt, or stimulate the human nervous system by interacting with the electrical properties of active neurons. In a second group are technologies (SPECT, PET, fMRI, NIRS) that measure aspects of blood flow and blood chemistry that change in response to local neural activity. The third category (DOT) may ultimately be able to respond to both aspects of human brain function. For a general overview, see Savoy (2001).

Spatial and Temporal Resolution Limits

Table 1 summarizes the approximate spatial and temporal resolutions for various brain imaging modalities. However, it should be understood that one-number summaries are highly misleading, for at least three reasons. First, the physical resolution limits of the *tool* may be different from the resolution limits implied by the biophysical phenomenon being measured. For example, MRI can collect images in a few milliseconds, and optical techniques like NIRS can collect information in microseconds, but both of these technologies, when applied to functional brain mapping, are constrained by the slower temporal resolution of the neuronally triggered hemodynamic changes. Similar statements can be made about spatial resolution. The second way in which one-number summaries can be misleading is that, for several modalities (notably fMRI), the numbers are moving targets that are changing quickly as improvements are made to the hardware and associated analysis software. Finally, there are often explicit trade-offs available when using these technologies. For example, if one is willing to spend an entire MRI session collecting one brain volume, it is possible to improve the spatial resolution. More generally, there are trade-

offs between imaging time, spatial resolution, and signal-to-noise ratio in all modalities, but most dramatically in MRI.

Table 1 lists the current resolutions for typical uses of the modalities in functional brain imaging. At least one specific application is mentioned in the section on retinotopy within which substantially greater spatial resolution is achieved.

Experimental Design and Data Analysis

In addition to the trade-offs with respect to imaging hardware, there are related constraints on available experimental designs. Table 1 includes several columns related to these issues. Most important is the column associated with multiple testing of a single subject. The ability to return, again and again, to the same brain with additional tests is one of the great practical strengths of the minimally invasive technologies that do not use ionizing radiation. This attribute is likely to be of particular importance in testing theories and models in the future.

The information on temporal design types is included in the table to indicate some constraints imposed on experimental design by the various modalities. In block design tests ("BLK"), the subject performs a given task for an extended period of time (say, 1 minute) to get the brain and its associated blood flow in a given state before an image is collected. Other design types refer to the use of single trials. In spaced single trials ("SST"), the responses to individual stimulus presentations are collected and averaged together. SST designs in the domain of blood-based imaging have intertrial intervals on the order of 10–20 s; SST designs in the domain of electrical modalities have intertrial intervals that are much shorter (between 250 and 2,000 ms). Rapid single trial ("RST") designs refer to the use of stimuli that elicit overlapping hemodynamic changes, which are separated via deconvolution during data analysis. This technique is specific to fMRI. Its advantages are the more efficient use of imaging time and the more rapid presentation of stimuli to keep subjects awake and engaged in the tasks; its disadvantages are greater complexity in specifying the design and analyzing the data, as well as less sensitivity. Note that the electrical modalities actually use stimuli that are more "rapid" in the sense that they can be presented at shorter intertrial intervals, but the electrical signals thus stimulated are over much faster than the hemodynamic changes, so they are not generally overlapping, and therefore the term SST is perhaps more appropriate. Both SST and RST, collectively called *event-related designs*, involve the combining of data from different trials within a given trial type. The ultimate, in terms of experimental resolution, is the individual single trial ("IST"), in which data from each individual stimulus presentation are analyzed independently from the rest. In theory, any modality could accomplish this; in practice, only those with sufficient signal-to-noise ratio can be used this way.

Applications in Vision

Retinotopy

The first application of fMRI-based research was in the domain of the early stages of visual processing. Indeed, the very first human fMRI study involved the demonstration that a region of the brain associated with early visual processing, occipital cortex in the calcarine fissure, yielded an NMR signal that varied as flashing lights were presented (or not) to a subject.

This demonstration was exciting, but the excitement was limited, for several reasons. First, nothing new had been demonstrated about human visual cortex. Second, there were a host of technical concerns which, had they been correct, would have meant that the spatial resolution obtainable with fMRI would be seriously compromised. And finally, most of the next simple advances would not

Table 1. Critical Attributes of Current Imaging Modalities

Modality	Spatial Resolution	Temporal Resolution	Temporal Design Types				Many Sessions with Single Subject
			BLK	SST	RST	IST	
O ¹⁵ -PET	8–12 mm	~30–60 s	✓	—	—	—	No
SPECT	3–8 mm	~20 s	✓	—	—	—	No
TMS, rTMS	~cm	10 s	✓	✓	—	—	Yes
MEG/EEG	~cm	1 ms	✓	✓	NA	—	Yes
NIRS	~cm	1 ms	✓	✓	✓	✓	Yes
iEEG	μm	1 ms	✓	✓	NA	✓	No
fMRI	1–8 mm	1,000 ms	✓	✓	✓	✓	Yes

Abbreviations: BLK: block design; SST: spaced single trials; RST: rapid single trials; IST: individual single trials; PET: positron emission tomography; SPECT: single-photon emission computed tomography; TMS, rTMS: (rapid) transcranial magnetic stimulation; NIRS: near infrared spectroscopy; MEG/EEG: magneto- and electroencephalography; iEEG: intracranial EEG; fMRI: functional magnetic resonance imaging; NA: not applicable.

go beyond what we already know from (invasive) single-cell recordings in nonhuman primates.

However, the development of fMRI in the ensuing years for the study of early visual processing addressed all these concerns, and went far beyond them. First, retinotopy was demonstrated for area V1 at a level of spatial resolution that exceeded any previously demonstrated with a noninvasive technique. Second, retinotopy was used to delineate multiple visual areas. Differences between the layout of human visual areas as compared with other primate species were demonstrated, and new visual areas apparently unique to humans were described.

The ability to routinely map retinotopically defined regions of the human brain has enabled progress in number of areas. In one clinically relevant application (Hadjikhani et al., 2001) the visual auras of a migraine headache were mapped for an individual subject as the headache progressed and the auras moved through different portions of the visual field. The ability to objectively observe the physiology underlying what had previously been considered to be purely subjective effects is of obvious practical consequence, especially in the context of evaluating drug treatments and other therapies. In a context perhaps more directly relevant to testing theories and models, several groups have claimed to detect ocular dominance columns in human visual cortex (Cheng, Waggoner, and Tanaka, 2001; Menon et al., 1997). This achievement uses MRI techniques that push the spatial resolution (with associated trade-offs in temporal resolution). If functional imaging of cortical columns in other brain areas is achieved (which is not a trivial extension, as the ocular dominance columns are some of the largest in cerebral cortex), the ability to test increasingly rich and detailed models will be likely.

Motion Processing and the Motion Aftereffect

One of the most robust findings in functional brain imaging is the activation of the cortical area known as V5 or MT by the visual presentation of moving stimuli. This area has been important in at least two types of studies: the popular visual illusion known as the motion aftereffect (MAE), and the documentation of changes in cortical activity in response to voluntary changes of attention.

Virtually concurrent with the earliest studies of retinotopy using fMRI, a classic psychological effect, the MAE, was seen to be associated with detectable brain activity localized to specific parts of the cortex associated with visual motion processing. When subjects looking at moving patterns reported a subjective MAE, specific brain areas—notably area MT/V5—showed increased activity. This initial finding has been extended using behavioral variants of the basic effect to demonstrate particularly tight correlation of MT activity and the subjective perception of the effect. Two studies made use of the fact that the MAE lasts longer (after adaptation) if it is not elicited by the presentation of a stationary test pattern. In one study, subjects were kept in the dark after adaptation. Activity in MT decreased when the subject was in the dark, and returned during subsequent viewing of a stationary target that elicited the subjective impression of motion. In another study, the MAE was generated in only part of the visual field, and MT activity was elicited only when the stationary target was presented to the adapted portion of the field (see Moore and Engel, 1999, for a summary). This collection of work gives some idea of the potential for hypothesis testing associated with functional brain imaging in vision.

Visual Object Representation: General Issues, and Are Faces a Special Case?

One of the most active areas of experimental research and theoretical analysis in the context of functional brain imaging of vision

has to do with visual object representation. Numerous imaging and clinical reports have documented localization of cortical function associated with particular classes of visual objects. The category of human faces has received by far the most attention in this regard, but many other categories (from general classes such as living versus inanimate or tools versus animals to highly specific classes such as cows versus horses) have been studied.

One focus of the debate is the meaning of these localized activations per se. Specifically, it has been documented that, for example, the “fusiform face area” responds best to faces in a number of contexts, but the area responds (statistically significantly) to numerous other categories of objects (e.g., Ishai et al., 1999). The question of whether it is best to think of this area as somehow face specific or, alternatively, as specific to any overpracticed category is one of the heated debates in the field. Lesion data are particularly relevant here. There is the relatively rare but well-documented phenomenon of prosopagnosia (a specific impairment of the visual perception of faces), and there is the even rarer case study of an otherwise healthy subject who lost the visual face-processing area of cortex in infancy and continued to show face-specific deficits as an adult. But disentangling the specificity for faces from the possible specificity for a more general overpracticed category has not yet been achieved.

A growing number of quantitative studies are attempting to understand and utilize this cortical specificity to study visual object representation. Discriminating the brain activation responses to different categories has been refined to demonstrating varying degrees of specificity even in areas that are not the best ones for the individual categories. So, for example, it is possible to discriminate houses from scissors even when the imaging data are restricted to areas that are maximally responsive to faces (Haxby et al., 2001).

Independent of the ultimate resolution of these concerns, these cortical areas of reasonable specificity can be exploited to test longstanding questions of theoretical interest in cognitive neuroscience and brain modeling. The following two sections discuss two aspects of visual processing that are necessarily top-down phenomena: voluntary modulation of attention, and imagery. In both cases, the ability to functionally specify an a priori region of interest for analysis (based on the object- or motion-selective specificity outlined above) is crucial for tight experimental and statistical tests.

Visual Attention

Attention is an intensively studied area of cognitive psychology and has been very popular for imaging studies. There are many reviews of visual attention in the context of imaging (e.g., Kanwisher and Wojciulik, 2000). The following discussion focuses on one aspect of this area.

A classic PET study (Corbetta et al., 1990) contrasted the activation of different cortical areas depending on whether the subject needed to attend to a single attribute of the stimuli (size/shape, color, or speed of motion) or to *any* of those three attributes when performing a discrimination task. A more recent study of visual attention took advantage of the known localization of function associated with three categories (motion processing, face processing, and place/building processing) to test an explicit theoretical question of long standing in cognitive psychology and neuroscience. Both of these studies required changes in voluntary attention, with (in some cases) associated changes in behavioral performance measures. But, independent of whether there were detectable changes in behavioral performance on the associated tasks, there were detectable and statistically significant changes in the activity of specific cortical areas. By dint of clever experimental design, these changes were relevant to specific theoretical questions, as elaborated below.

One early fMRI-based study demonstrated that the use of voluntary attention (deciding whether to attend to a subset of moving dots or to a subset of stationary dots in a field of moving and stationary dots) caused detectable changes in MR signals associated with a visual motion-processing area in cortex (O'Craven et al., 1997). This study did not have an overt behavioral measure to provide external evidence that subjects were actually performing their assigned tasks. But the data were sufficiently clean and unambiguous that this study was published and gained considerable attention.

Over the ensuing years the study was replicated and extended in a number of ways by different laboratories around the world. The initial basic demonstration of attentional modulation became the starting point for more subtle experiments, experiments that were more tightly tied to behavioral measures. Importantly, both the qualitative and the quantitative measures of attentional modulation were replicated. For instance, the motion-processing area was active whenever visual movement was present, but that activity increased by about 50% when the subject was attending to the movement, in contrast to when the subject was not attending to the movement. The studies that used analogous tasks as part of their design found quantitatively similar changes.

Taking advantage of the findings that other parts of the brain showed localized activity in response to certain classes of objects (e.g., the so-called fusiform face area [FFA] and parahippocampal place area [PPA]) one study (O'Craven, Downing, and Kanwisher, 1999) addressed the question of whether there was increased processing of the irrelevant attributes of attended objects relative to unattended objects. Stimuli were designed that mixed the attributes of faces, places, and motion on each trial. The discrimination task on a given trial depended on only one of these attributes. Not surprisingly, the three relevant cortical areas (MT/V5, FFA, and PPA) each showed increased activation when the discriminating attribute (motion, faces, or places) was the relevant one for that cortical area. More interestingly, contrasting the imaging data from those trials distinguished by an irrelevant attribute of the stimulus (e.g., whether or not an image was moving in a face discrimination task) led to the demonstration of increased activity in the area responsible for that attribute (e.g., the motion-processing area in the example above), even though the presence or absence of that attribute (motion, in this example) was irrelevant to the discrimination task. This finding was interpreted as supporting models of attentional processing that had an "object-based" component: once the subject is attending to an object for whatever reasons, all attributes of the attended object are enhanced in processing, including those that are not functionally relevant at the time. Thus, brain imaging experiments are being applied to theoretical questions of longstanding interest in cognitive psychology.

Visual Imagery

There have been numerous studies of visual imagery attempting to document changes in brain activity while subjects "imagine," i.e., recreate in their minds, some approximation to the brain state elicited by the physical presentation of various visual stimuli, in the absence of those stimuli. Continuing the thread started with the studies of voluntary visual attention in the previous section, *imagery* shares the property of being necessarily a top-down phenomenon. In at least one study involving the visual imagery of familiar faces or familiar locations, the strength and functional localization of the imagined visual images were sufficient to permit a data analyst looking only at two regions of the cortex (previously localized as being particularly responsive to faces or places, respectively) to estimate, with far greater than chance accuracy, whether the subject was imagining one or the other of the two classes of stimuli *on individual stimulus presentation trials*. That is, it was not necessary

to average across a collection of trials of a given type (e.g., imagining a face) in order to be able to do the discrimination (O'Craven and Kanwisher, 2000). Studies such as this one have obvious potential for both practical applications and the testing of theoretical models of brain function.

Discussion

The present article described some of the lowest-level applications (using the regular retinotopic mapping of multiple visual areas to distinguish those areas), some of the middle-level applications (connected to functional localization associated with object and motion processing), and some of the highest, necessarily top-down applications associated with voluntary attention and imagery. Visual stimulus processing has probably been the most intensively pursued application of functional brain imaging. One reason for this popularity is the wealth of data on the primate visual system, obtained largely through invasive, single-cell recordings and lesion studies. The early dependence on connection to known primate neurophysiology is, in recent times, being turned around. Several laboratories are now developing functional MRI suites designed specifically to study nonhuman primates. The idea is to use the invasive technologies like single-cell recording, adapted for the MR environment, and get a deeper understanding of both the functional brain structures and the relationship between neural activity and hemodynamics, using methods that would be unethical in human subjects.

This article has reviewed some applications in functional brain imaging of the visual modality in which *quantitative* measures of activity were obtained. In analogy with a classic observation in experimental psychology by Paul Meehl (1967), it should be understood that increasing the statistical power of the experimental tool (whether by increasing the number of subjects or by increasing the strength and sensitivity of the imaging hardware) is generally not sufficient to distinguish interesting models of brain function and organization. The fact that brain area *X* shows a different level of activity in response to task 1 than in response to task 2 has been amply demonstrated for many *X*s and pairs of tasks. Increasing the power of the tools will only increase the set of *X*s and task pairs for which those differences become statistically significant (Savoy, 2001). Quantitative and parametric observations must be obtained to help functional brain imaging live up to its potential, specifically, the potential to create tight, quantitative tests of theory and causal models. Many experimentalists understand this challenge and the associated dilemmas, and an increasing number of studies go beyond the "boxology" sometimes disparagingly applied to functional imaging studies. Increased communication between brain theorists and functional neuroimagers can only improve the situation.

Road Map: Cognitive Neuroscience; Vision

Related Reading: Covariance Structural Equation Modeling; Statistical Parametric Mapping of Cortical Activity Patterns; Synthetic Functional Brain Mapping

References

- Corbetta, M., Miezin, F. M., Dobmeyer, S., Shulman, G. L., and Petersen, S. E., 1990, Attentional modulation of neural processing of shape, color, and velocity in humans, *Science*, 248:1556–1559.
- Cheng, K., Waggoner, R. A., and Tanaka, K., 2001, Human ocular dominance columns as revealed by high-field functional magnetic resonance imaging, *Neuron*, 32:359–374.
- Hadjikhani, N., Sanches del Rio, M., Wu, O., Schwartz, D., Bakker, D.,

- Fischl, B., Kwong, K. K., Cutrer, M. F., Rosen, B. R., Tootell, R. B. H., Sorensen, A. G., and Moskowitz, M. A., 2001, Mechanisms of migraine aura revealed by functional MRI in human visual cortex, *Proc. Natl. Acad. Sci. USA*, 98:4687–4692.
- Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., and Pietrini, P., 2001, Distributed and overlapping representations of faces and objects in ventral temporal cortex, *Science*, 293:2425–2430.
- Ishai, A., Ungerleider, L. G., Martin, A., Shouten, J. L., and Haxby, J. V., 1999, Distributed representation of objects in the human ventral visual pathway, *Proc. Natl. Acad. Sci. USA*, 96:9379–9384.
- Kanwisher, N., and Wojciulik, E., 2000, Visual attention: Insights from brain imaging, *Nature Rev. Neurosci.*, 1:91–100. ♦
- Meehl, P. E., 1967, Theory-testing in psychology and physics: A methodological paradox, *Philos. Sci.*, 34:103–115.
- Menon, R. S., Ogawa, S., Strupp, J. P., and Ugurbil, K., 1997, Mapping ocular dominance columns in V1 using fMRI, *J. Neurophysiol.*, 77:2780–2797.
- Moore, C., and Engel, S. A., 1999, Visual perception: Mind and brain see eye to eye, *Curr. Biol.*, 9:R74–R76.
- O'Craven, K. M., Downing, P. E., and Kanwisher, N., 1999, fMRI evidence for objects as the units of attentional selection, *Nature*, 401:584–587.
- O'Craven, K. M., and Kanwisher, N., 2000, Mental imagery of faces and places activates corresponding stimulus-specific brain regions, *J. Cognit. Neurosci.*, 12:1013–1023.
- O'Craven, K. M., Rosen, B. R., Kwong, K. K., Treisman, A., and Savoy, R. L., 1997, Voluntary attention modulates fMRI activity in human MT/MST, *Neuron*, 18:591–598.
- Savoy, R. L., 2001, History and future directions of human brain mapping and functional neuroimaging, *Acta Psychol.*, 107:9–42. ♦

Imitation

Aude G. Billard

Introduction

Imitation—the ability to recognize and reproduce others' actions—is a powerful means of learning and developing new skills. Species endowed with this capability are provided with fundamental abilities for social learning. In its most complex form, imitation provides fundamental capabilities for social cognition, such as the recognition of conspecifics, the attribution of others' intentions, and the ability to deceive and to manipulate others' states of mind.

Research on imitation builds a bridge between biology and engineering, and between the study and use of imitation. Biology seeks to better understand the cognitive and neural processes behind the different forms of animal imitation, and how these relate to the evolution of social cognition. Engineering uses studies of the biological processes of human imitation to design robot controllers and computational algorithms enabling learning and imitative skills similar in robustness and flexibility to human skills.

There are three major levels of modeling of imitation. *Theoretical modeling* derives models of the cognitive mechanisms behind imitation based on behavioral studies of humans' and other animals' imitation. *Computational modeling* builds models of the neural mechanisms, and their brain correlates, behind imitation learning in human and other animals. *Robotics modeling* designs algorithms for imitation learning, implementable in hardware systems, that allow a robot to be taught by demonstration. Next we briefly describe key findings and issues faced by each of the three levels of imitation modeling.

Theoretical Modeling

The study of animal imitation encompasses a large range of disciplines, including ethology, neuroscience, psychology, and linguistics.

For ethologists, the major issue is to define what behaviors the term *imitation* refers to and in which species these behaviors are exhibited (for reviews, see Whiten, 2000; Heyes, 2001). Animal imitation seems best described in terms of levels of complexity. Imitation (or “true” imitation) is contrasted to mimicry or copying. True imitation is the ability to replicate and, by so doing, learn skills that are not part of the animal's prior repertoire, by observation of those performed by others. Mimicry, in contrast, is the ability to replicate a behavior that is usually part of the usual animal repertoire.

Simple forms of imitation that probably require no understanding of intention or theory of mind are found in, e.g., rats and monkeys. These species' copying ability is considered to be an instance of *social facilitation*, in which the correct behavior is prompted by the social context. This simple imitative behavior relies on a form of associative learning that accepts temporal delays, imprecise timing, and incomplete cues (Heyes, 2001). In this form of imitation, the act of observing enhances learning of a skill by reducing the number of incorrect associations.

More complex forms of imitation are demonstrated by apes and dolphins. Chimpanzees and orangutans can master simple sequential, manipulatory tasks. They are capable of replicating part of the observed behavior in a different context than that in which it was observed. Dolphins can be trained to copy long sequences of body movements following human demonstration, showing an ability to map different body structures to their own (they respond to the demonstrator's movements of the legs and arms with similar movements of their tail and fins, respectively).

These more complex forms of imitation are set apart from simpler ones because they encompass the ability to reproduce *sequences* of actions and the ability to *transform* the actions so as to produce variations (subparts) of the observed behavior in the same or a different context (see, e.g., Byrne and Russon, 1998).

The ability to imitate reaches its fullest complexity in humans. Humans can imitate any actions of the body based on a variety of purposes or goals, such as the goal of reproducing the aesthetic (e.g., dance), efficiency (e.g., sport), or precision (e.g., surgery) aspect of the movement. Imitation can be *immediate* or *deferred*, depending on whether the replication occurs within a short (few minutes) or long (hours, days) time after the demonstration. It may be partial or selective (when only part of the imitative behavior is replicated), goal-directed (when only the means-end of the demonstration is perfectly reproduced), or exact (Bekkering and Prinz in Dautenhahn and Nehaniv, 2002). Imitation in humans extends to verbal and facial expression, and from there to high-level cognitive and behavioral skills. It is a fundamental means to relate socially to others, and people who are impaired in their imitative skills, such as people with autism, also show general impairment in other social skills.

For psychologists, imitation is crucial to the child's growing capacity for representation and symbolization. Meltzoff and colleagues' work contributed to redefining the developmental stages of children's imitation proposed by Piaget in *Play, Dreams and*

Imitation (see Meltzoff and Moore, 1999). In infants, immediate imitation of facial expression appears soon after birth, suggesting an “innate” kinesthetic-visual mapping.* Deferred imitation appears as early as 9 months, implying a growing capacity for internal representation of others’ movements. Generalized imitation involving numerous modalities, such as vocal and verbal imitation and the ability to imitate a great variety of actions, begins around 15 to 18 months.

An important body of research in linguistics studies vocal and verbal imitation in birds, with the goal of understanding the role that hearing plays in tuning speech production and how this can relate to similar developmental processes in human infants (see Doupe and Kuhl, 1999). Young birds’ songs mature in the presence of a tutor (usually the parent bird) and are species and region specific. Parrots and mynah birds are particularly intriguing because of their ability to reproduce segments of human speech.

Studies of animal imitation show that imitation results not from a single mechanism but from several cognitive mechanisms that are multimodal (audiomotor, visuokinesthetic-motor) in essence and are used for other (nonimitative) behaviors. Visuomotor imitation is better understood at this stage than is vocal imitation, as it can profit from the large body of literature on perception and production of motion. Findings from these studies directly relevant to the study of imitation are briefly summarized next.

Motion Perception

Since Johansson’s landmark study in 1973, an abundance of literature has demonstrated the capacity of humans to recognize biological (especially human) motions from a limited number of cues (these studies use point-light display techniques that allow the viewer to see only one point for each moving limb) (for a review, see Dittrich, 1999). Humans can easily make out the general features of the motion, distinguishing the type of gait or the type of action, as well as specific features, such as the weight of an object being lifted or the age and sex of the walker. More important, humans are quite capable of distinguishing between biological and nonbiological motions. This ability relies on powerful visual mechanisms for quickly extracting relevant features from the kinematics of multiple-joint motion. Some of these features are the phase or relationship across limb motion, the orientation, and the speed of limb movement.

Motor Control

Although there is evidence that the brain can recognize motion from a limited number of clues, it is not yet understood which information is used to recognize and to reproduce the motion. Because of the redundancy of multiple-joint motion, the information offered in point-light display experiments is usually not sufficient to lead to a single plausible solution. It seems, therefore, that the mechanisms humans use to assist in visual reconstruction of motion rely on models of the structure of the human body and the dynamics of its possible motion.

Evidence that the central nervous system (CNS) uses models of body dynamics to direct motion also comes from purely motor control studies (see MOTOR CONTROL, BIOLOGICAL AND THEORETICAL). The idea is that, rather than relying on sensory feedback (which is too slow to reach the CNS in time for the next motor command), the CNS uses *feedforward control* to control movements; that is, it uses *inverse forward* models to predict the ex-

pected outcome of a command as well as to estimate the current position and velocity of the moving limbs.

In summary, evidence from psychophysical studies of motion perception and from motion studies suggest that, to achieve a good replication of movements from a paucity of visual cues, the brain uses models of human kinematics and dynamics of motion. Moreover, it is likely that visual and motor representation of movements bear a close relationship for the mapping to be immediate and precise. It is not yet understood how the CNS builds these representations.

Computational Modeling

The challenge faced by computational modeling is to construct a model that can account for all the instances of imitation reported in the literature. The model should provide a means of naming and distinguishing animal imitative abilities, following a list of fundamental cognitive components. Ideally, this hierarchical representation of animal imitation should follow the evolutionary tree, such that the different cognitive processes can be linked to the evolution of specific neural structures. We review next the evidence for neural structures specific to imitation.

Neural Structures Behind Visuomotor Imitation

For a long time imitation has been a topic of research primarily in the cognitive and psychological sciences; only recently has imitation become the explicit topic of a number of neuroscience studies. This new trend started with the discovery of the *mirror neuron* system (Rizzolatti et al., 1996), a neural circuit in F5 area of monkey premotor cortex that is active both when the monkey observes another monkey or a human grasping or manipulating objects and when the monkey performs the same manipulation. The mirror neuron system has been proposed as the link between visual and motor representation that is necessary to learn from the observation and imitation of others’ actions. Evidence from brain imaging studies (e.g., Decety et al., 2002) suggests the existence of a similar system in humans involving predominantly Brodmann’s areas 44 and 45 (Broca’s areas), 40 (parietal lobe), and 21 (superior temporal sulcus).

Evidence that specific areas of the human brain contribute to imitation also comes indirectly from lesion studies. Studies of abnormal imitative behavior can be separated in two groups:

1. Patients suffering from a lack of or strong deficiency in the ability to imitate. Patients with ideomotor apraxia after parietal lesion are unable to make symbolic gestures or to act out the use of an object in response to an oral request (De Renzi, Motti, and Nichelli, 1980). It is unclear whether ideomotor apraxia results from a deficit in motor imagery mechanisms or in motor execution. Apraxic patients are sometimes also incapable of recognizing a correctly produced gesture when given a stationary (photograph) or moving visual presentation. This suggests that the parietal lobe provides the locus of a neural network responsible for the translation of mental representation into movement production. However, the absence of systematic co-occurrence of ideomotor apraxia and impairment in gesture recognition indicates that motor imagery and motor execution remain two separate processes, even if closely interconnected.

2. Patients displaying obstinate imitation behavior, that is, a compulsive imitation behavior that cannot be stopped easily by command. Patients with frontal lobe damage sometimes display imitation behavior in which they imitate the examiner’s gestures without being so instructed (Lhermite, Pillon, and Serdaru, 1986). This type of disorder supports the view that the frontal lobe modulates (mainly inhibits) a subcircuit that continually interprets vi-

*Unsuccessful replications of the work led to a large debate that seems now quasi-resolved, thanks to several consecutive successful replications.

sual observation of movements through the activation of motor patterns that would produce the same movements (a typical mirror neuron circuit).

Taken together, evidence from lesion studies and brain imaging suggests a major role for parietomotor connectivity as a basic circuit (possibly the mirror neuron system) behind movement imitation, and it also highlights the importance of frontoparietal connectivity in regulating this circuit.

Since its discovery, the mirror neuron system has led to a number of speculations about its role in imitation. However, evidence to support these hypotheses is still lacking. Research on the human mirror system is still in its early stage and has addressed only simple actions of the arms and hands (fingers). It remains to be shown that mirror neurons exist for driving motion of other limbs, and to understand their role in driving imitation and imitation learning of complex actions (so as to qualify as “true imitation”).

Computational modeling investigates some of the possible implications of a high-level representation of movements common to both visual and motor systems (a mirror neuron system) for imitation learning. In this quest, Oztop and Arbib developed a computational model of monkey mirror neuron system (see Arbib et al., 2002, and *LANGUAGE EVOLUTION: THE MIRROR SYSTEM HYPOTHESIS*). The model accounts for the role of the parietal lobe and F5 area in recognition and control of grasping. In particular, it gives a description of how, through learning of performing grasps, visuomotor (from parietal lobe to F5) connectivity can be built.

At a higher level of abstraction, computational models of the neural and cognitive correlates to human imitation are developed. Demiris and Hayes’s model (in Dautenhahn and Nehaniv, 2002) gives an account of the cognitive processes behind imitation, in which the motor system is either active (active imitation) or passive (passive imitation) during perception. The active imitation mode encompasses a motor imagery mechanism (a type of mirror system) in which the same motor structures used in producing motion are used during visual perception for classification and recognition of motion.

Billard’s (1999) model gives a high-level, comprehensive, but simplified representation of the visuomotor pathway behind learning by imitation, from processing real video data to directing a dynamic simulation of a humanoid or an actual robot (Figure 1). The model has composite modules whose functionalities were inspired by those of specific brain regions, incorporating abstract models of the superior temporal sulcus (STS), the spinal cord, the primary motor cortex (M1), the dorsal premotor area (PMd), and the cerebellum. Each part is implemented at a connectionist level,



Figure 1. Robota, a minihumanoid, doll-like robot, can mirror the arm and head motion of a human demonstrator by visual tracking of the optical flow. Researchers are investigating its use as an educational toy for normal and handicapped children. (From Billard in Dautenhahn, K., and Nehaniv, C., Eds., 2002, *Imitation in Animals and Artifacts*, Cambridge, MA: MIT Press, Reproduced with permission.)

where the neuron unit is modeled as a *leaky integrator*. Neurons in the PMd module respond both to visual information (from STS) and to corresponding motor commands produced by the cerebellum. The STS-PMd-M1 interconnection is a simplified model of a mirror neuron system. The biological plausibility of the model was validated against kinematic recording of human motion (Billard, 1999) and functional magnetic resonance imaging (fMRI) data of human imitation of finger motion (Arbib et al., 2000).

Robotics Modeling

Robotics investigates the potential of imitation learning as a user-friendly means of human-robot interaction. The goal is to provide robots with the capacity for being reprogrammed in a nonexplicit fashion, that is, through demonstration. The challenge is to determine learning algorithms that are flexible across tasks and across platforms (robots).

An important issue dealt with by computational and robotic modeling is that of determining a measure of the similarity across demonstrator and imitator motions (Schaal, 1999; Dautenhahn and Nehaniv, 2002). For instance, when imitating grasping an object, one can reproduce one, a few, or all characteristics of the movements, and one can in principle reproduce (1) the goal of the movement (grasping the object with any effector following any path), (2) the goal of the movement and the correct effector (grasping the object with the correct hand), and (3) the detail of each joint movement, the motion of subsegments, and even the overall speed of movement. In each case, a different measure of the similarity between demonstrator and imitator movements must be used to account for the correctness of the reproduction. The measure should, in some cases, be qualitative, comparing the relationships across objects (which hand, which object), whereas in other cases it is quantitative, comparing the paths followed by each hand or comparing the angular trajectories of each joint.

In construction imitations of joint motion, the problem is how to transfer human motions into robot motions, insofar as humans and robots have very different dynamics. In other words, the problem is how to compute the inverse kinematics (if working in eccentric coordinates, such as when using visual tracking) or the inverse dynamics (when working in intrinsic coordinates such as when using manipulandum; see *ROBOT LEARNING* and *ROBOT ARM CONTROL*).

A large part of robotics research follows a purely engineering perspective, solving assembly task learning from observation (e.g., Friedrich et al., 1996). Typically, the demonstrator’s movements are measured either as torques and joint angle displacements through the use of a manipulandum or from visual tracking. The robot is then controlled using classical planning techniques.

More recent efforts, inspired by computational modeling of human imitation, are oriented toward analyzing the underlying mechanisms of imitation in natural systems and modeling those mechanisms in artificial ones. The goal here is to design robot controllers showing similar robustness and adaptability as natural systems. Biologically inspired models of the ability to imitate have been tested in experiments in which the robot could replicate movements of the head and arms of a human (see Schaal, 1999, for a review).

Discussion

Imitation is a concept heavily debated in the biological literature. Modeling can eliminate some of the debate by defining what minimal computation is necessary for each type of imitation. Several theoretical models have been proposed to distinguish between each level of computation, e.g., by differentiating between purely associative imitation (low-level imitation) and sequential imitation (high-level imitation). Although conceptual distinctions are impor-

tant, they are hard to validate through behavioral studies only. Computational models play a key role in validating these theories by offering an explicit functional description of the computation required for each level of imitation. Realistic modeling that uses real data as input (e.g., video recording of human or animal motion) and physical devices (e.g., robots) or realistic simulation as output is essential to gain a fuller understanding of the mechanisms underlying sensorimotor coordination in imitation.

At this point, there are very few computational or robotic models of imitation. However, the field is currently popular and is bound to grow rapidly within the next years. Its popularity is in part due to recent technological development in robotics that have allowed the design of humanoid robots whose joint complexity approaches that of humans. Modeling of imitation has also benefited from a recent spate of neurological data on human and monkey imitation. Computational and robotic modeling are expected to fill in the gaps between modeling of low-level information (from neurological studies) and modeling of high-level information (from behavioral studies). Modeling of imitation should lead to a better understanding of the neural mechanisms at the basis of social cognition and offer new perspectives on the evolution of animals' ability for social representation.

Road Map: Cognitive Neuroscience

Related Reading: Action Monitoring and Forward Control of Movements; Grasping Movements: Visuomotor Transformations; Language Evolution: The Mirror System Hypothesis; Motor Primitives; Reaching Movements: Implications for Computational Models; Sequence Learning

References

- Arbib, M., Billard, A., Iacoboni, M., and Oztop, E., 2000, Mirror neurons, imitation and (synthetic) brain imaging, *Neural Netw.*, 13:953–973.
- Billard, A., 1999, Learning motor skills by imitation: A biologically inspired robotic model, *Cybern. Syst.*, 32:155–193.
- Byrne, R. W., and Russon, A. E., 1998, Learning by imitation: A hierarchical approach, *Behav. Brain Sci.*, 21:667–721.
- Dautenhahn, K., and Nehaniv, C., Eds., 2002, *Imitation in Animals and Artifacts*, Cambridge, MA: MIT Press.
- Doupe, A. J., and Kuhl, P. K., 1999, Birdsongs and human speech: Common themes and mechanisms, *Annu. Rev. Neurosci.*, 22:567–631. ♦
- Decety, J., Chaminade, T., Grezes, J., and Meltzoff, A. N., 2002, A PET exploration of the neural mechanisms involved in reciprocal imitation, *Neuroimage*, 15:265–272.
- DeRenzi, E., Motti, F., and Nichelli, P., 1980, Imitating gestures: A quantitative approach to ideomotor apraxia, *Arch. Neurol.*, 36:6–10.
- Dittrich, W. H., 1999, Seeing biological motion: Is there a role for cognitive strategies? in *Lecture Notes in Artificial Intelligence* (A. Braffort et al., Eds.), Berlin: Springer-Verlag, 1739, pp. 3–22.
- Friedrich, H., Munch, S., Dillmann, R., Bocionek, S., and Sassini, M., 1996, Robot Programming by demonstration (RPD): Supporting the induction by human interaction, *Machine Learn.*, 23:163–189.
- Heyes, C. M., 2001, Causes and consequences of imitation, *Trends Cogn. Sci.*, 5:253–261. ♦
- Lhermite, F., Pillon, B., and Serdaru, M., 1986, Human autonomy and the frontal lobes: Part I. Imitation and utilization behavior: A neuropsychological study of 75 patients, *Annu. Neurol.*, 19:326–334.
- Meltzoff, A. N., and Moore, M. K., 1999, Resolving the debate about early imitation, in *Reader in Developmental Psychology* (A. Slater and D. Muir, Eds.), Oxford: Blackwell, pp. 151–155. ♦
- Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L., 1996, Premotor cortex and the recognition of motor actions, *Cogn. Brain Res.*, 3:131–141.
- Whiten, A., 2000, Primate culture and social learning, *Cogn. Sci.*, 24, 2000. ♦
- Schaal, S., 1999, Is imitation learning the route to humanoid robots?, *Trends Cogn. Sci.*, 3:233–242. ♦

Independent Component Analysis

Anthony J. Bell

Introduction

Independent component analysis (ICA) is a linear transform of multivariate data designed to make the resulting random vector as statistically independent (factorial) as possible. Despite its relatively short history, it is rapidly becoming a standard technique in multivariate analysis. In signal processing it is used to attack the problem of the blind separation of sources (Haykin, 2000), for example of audio signals that have been mixed together by an unknown process. In the area of neural networks and brain theory, it is an example of an information-theoretic unsupervised learning algorithm, and one that provides one of the most compelling accounts of how early sensory processing may self-organize. That is, when an ICA network is trained on an ensemble of natural images, it learns localized oriented receptive fields (see FEATURE ANALYSIS) qualitatively similar to those found in area V1 of mammalian visual cortex (Bell and Sejnowski, 1997). Finally, in the increasingly important area of analyzing multivariate brain data (multi-electrode recordings, electroencephalography, functional magnetic resonance imaging), ICA has been used to pull recordings apart into components of interest to researchers attempting to understand task-related spatial and temporal brain dynamics (Jung et al., 2000; BRAIN SIGNAL ANALYSIS).

Thus we have the pleasingly ironic situation in which the same neural network algorithm is being used both as an explanation of brain properties and as a method of probing the brain.

The idea is as follows. We are given an N -dimensional random vector, \mathbf{x} , which could be the instantaneous output of N microphones, N time points of an audio signal, N pixels of an image, the output of N electrodes that record brain potentials, or any other multidimensional signal. Typically there will be many correlations between the elements of the vector \mathbf{x} . ICA, like PRINCIPAL COMPONENT ANALYSIS (PCA) (q.v.), is a method of removing those correlations by multiplying the data by a matrix, as follows:

$$\mathbf{u} = \mathbf{W}\mathbf{x} \quad (1)$$

(Here, we imagine the data are zero-mean; see the next section for details on preprocessing.) But whereas PCA merely uses second-order statistics (the covariance matrix), ICA uses statistics of all orders. PCA attempts to decorrelate the outputs (using an orthogonal matrix \mathbf{W}), while ICA attempts to make the outputs statistically independent, and places no constraints on the matrix \mathbf{W} . Statistical independence means the joint probability density function (p.d.f.) of the output *factorizes*:

$$p(\mathbf{u}) = \prod_{i=1}^N p_i(u_i) \quad (2)$$

while decorrelation means only that $\langle \mathbf{u}\mathbf{u}^T \rangle$, the covariance matrix of \mathbf{u} , is diagonal ($\langle \cdot \rangle$ means average).

Another way to think of the transform in Equation 1 is as follows:

$$\mathbf{x} = \mathbf{W}^{-1}\mathbf{u} \quad (3)$$

In this, the data are formed by linear superposition of *basis functions* (columns of \mathbf{W}^{-1}), each of which is activated by an independent component, u_i . We call the rows of \mathbf{W} *filters* because they extract the independent components. In orthogonal transforms such as PCA, the Fourier transform, and many wavelet transforms, the basis functions and filters are the same (because $\mathbf{W}^T = \mathbf{W}^{-1}$), but in ICA they are different.

The usefulness of a nonorthogonal transform sensitive to higher-order statistics can be seen in Figure 1. Here we plot PCA and ICA basis functions (axes) for a two-dimensional data distribution. Clearly, the ICA axes capture the structure of the data much better. Although this data distribution may look strange, it is actually very common in natural data, much more common than those who like to model data with Gaussians, or mixtures of Gaussians, might suppose. It comes from the nonorthogonal “mixing together” of

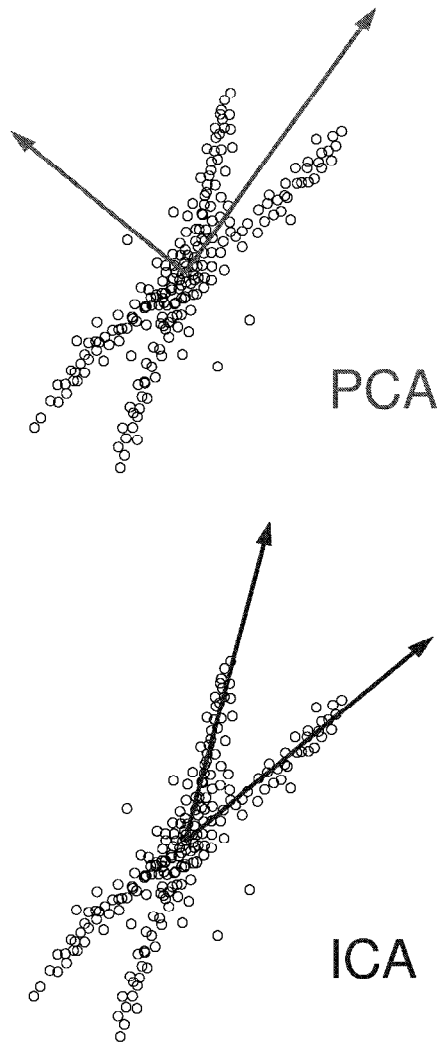


Figure 1. The difference between PCA and ICA on a nonorthogonal mixture of two distributions that are independent and highly sparse (peaky with long tails). An example of a sparse distribution is the Laplacian $p(x) \propto e^{-|x|}$. PCA, looking for orthogonal axes that are ranked in terms of maximum variance, completely misses the structure of the data. Although these distributions look strange, they are very common in natural data. (Courtesy of T.-P. Jung.)

highly sparse independent components, where by *sparse* we typically mean much peakier than a Gaussian, with longer tails. A more technical term for sparse is super-Gaussian. The ICA algorithm we will describe is ideally suited for extracting these sparse independent components.

Before describing the algorithm, we make some remarks regarding the origin of the various algorithms and the relations between them, with pointers to tutorial literature. This is necessary for a balanced treatment, as there is no one ICA algorithm.

The ICA problem was introduced by Herault and Jutten (1986). The results of their algorithm were poorly understood and led to Comon’s 1994 paper defining the problem, and to his solution using fourth-order statistics. Much work took place in this period in the French signal processing community, including the maximum likelihood approach (Pham, Garrat, and Jutten, 1992), which subsequently formed the basis of Cardoso and Laheld’s (1996) EASI method. These methods are very close to the Infomax approach (Bell and Sejnowski, 1995) so we will refer to this algorithm as Infomax/ML-ICA. Cichocki, Unbehauen, and Rummert (1994) had proposed an algorithm that motivated Amari (1997) and colleagues to show that its success was due to its relation to a “natural gradient” modification of the Infomax/ML-ICA gradient. This modification greatly simplified the algorithm, and made convergence faster and more stable.

This algorithm, which we might, rather clumsily, call NatGrad-Infomax/ML-ICA, is thus derived from multiple authors and is the most widely used adaptive, or on-line (i.e., stochastic gradient), method for ICA. It is also the most neural-network-like. It is the one we will describe in the body of this article, at the risk of underrepresenting other approaches. Useful batch algorithms also exist, such as Hyvärinen’s FastICA and many cumulant-based techniques.

Helpful review papers comparing the different algorithms are Lee et al. (1998) and Hyvärinen (1999). The edited collection of Haykin (2000) contains excellent survey papers from many of the authors mentioned above, and the edited collection of Girolami (2000) contains more recent theoretical work and examples of many applications. There have been two international workshops on the topic (ICA 1999 and 2000), the proceedings of which contain much additional material.

An Algorithm for ICA

This section provides enough information for the reader to implement ICA. To follow the full derivations, readers will have to consult the original sources. There is only space here to highlight the mathematics.

Preprocessing

To start with, the linear transform of Equation 1 should actually be an *affine* transform: $\mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{w}$, where \mathbf{w} is an $N \times 1$ “bias” vector that centers the data on the origin. But if we assume the independent component p.d.f.s, $p_i(u_i)$, are roughly symmetric, then it is simpler to make the data zero-mean beforehand. An additional preprocessing step, one that speeds convergence, is to “sphere” the data beforehand, that is, to diagonalize its covariance matrix. These preprocessing steps are achieved as follows:

$$\mathbf{x} \leftarrow 2\langle \mathbf{x}\mathbf{x}^T \rangle^{-1/2}(\mathbf{x} - \langle \mathbf{x} \rangle) \quad (4)$$

by which we mean that we subtract the mean of the data, then multiply by twice the inverse square root of its covariance matrix. This yields a decorrelated data ensemble whose covariance matrix satisfies $\langle \mathbf{x}\mathbf{x}^T \rangle = 4\mathbf{I}$, where \mathbf{I} is the identity matrix. This is a useful starting point for further training with the *logistic*-ICA algorithm, which we will describe below. Note that this sphering method is

not PCA. It is another decorrelation method called zero-phase whitening, which we have empirically found to be a better starting point for training than PCA-style decorrelation. In fact, there are many decorrelating, or second-order independent transforms, and extra constraints are needed to choose one of them. Zero-phase whitening constrains the matrix \mathbf{W} to be symmetric, while PCA constrains it to be orthogonal. ICA, also a decorrelation technique, but without any constraints on \mathbf{W} , finds its constraints in the higher-order statistics of the data.

Natural-Gradient Infomax/ML ICA

Most neural network algorithms have an objective function. ICA is no different. Its objective is to minimize the *redundancy* between the outputs. This is a generalization of the mutual information and is written as follows:

$$I(\mathbf{u}) = \int p(\mathbf{u}) \log \frac{p(\mathbf{u})}{\prod_{i=1}^N p_i(u_i)} d\mathbf{u} \quad (5)$$

It is easily verified that this redundancy measure has the value of 0 when the p.d.f. $p(\mathbf{u})$ factorizes as in Equation 2.

This is actually a difficult function to minimize directly. The insight that led to the Infomax-ICA algorithm was that $I(\mathbf{u})$ is related to the joint entropy, $H(\mathbf{g}(\mathbf{u}))$, of the outputs passed through a set of sigmoidal nonlinear functions, \mathbf{g} . The relation is as follows:

$$I(\mathbf{u}) = -H(\mathbf{g}(\mathbf{u})) + E \left[\sum_i \log \frac{|g'_i(u_i)|}{p_i(u_i)} \right] \quad (6)$$

Thus, if the absolute values of the slopes of the sigmoid functions, $|g'_i(u_i)|$, are the same as the independent component p.d.f.s, $p_i(u_i)$, then Infomax (maximizing the joint entropy of the $\mathbf{g}(\mathbf{u})$ vector) will be the same as ICA (minimizing the redundancy in the \mathbf{u} -vector). The principle of matching the g 's to the p 's is illustrated in Figure 2, where a single Infomax unit attempts to match an input Gaussian

distribution to a logistic sigmoid unit, for which

$$g(u) = \frac{1}{1 + e^{-u}} \quad (7)$$

The match cannot be perfect, but it does approach the maximum entropy p.d.f. for a distribution bounded between 0 and 1—in other words, the unit distribution—and this is done by maximizing the expected log slope, $E[\log |g'(wx)|]$. The generalization of this idea to N dimensions leads to maximizing the expected log determinant of the Jacobian matrix $[\partial g_i(u_i)/\partial x_{ij}]_{ij}$. This optimization attempts to map the input vectors uniformly into the unit N -cube (assuming that the g -functions are still 0–1 bounded). Intuitively, we can see the following: If our outputs are spread evenly in a cube, then telling you the value along one axis does not tell you anything about the values along the other axes. This is the intuition behind statistical independence.

To cut a long story short, the exact stochastic gradient descent algorithm that maximizes $H(\mathbf{g}(\mathbf{u}))$ is:

$$\Delta \mathbf{W} \propto \mathbf{W}^{-T} + \mathbf{f}(\mathbf{u})\mathbf{x}^T \quad (8)$$

where $-T$ denotes inverse transpose, and the vector function, \mathbf{f} , has elements

$$f_i(u_i) = \frac{\partial}{\partial u_i} \ln g'_i(u_i) \quad (9)$$

When $g'_i(u_i) = p_i(u_i)$ for all i , then, according to Equation 6, we have an exact ICA algorithm.

This leaves us with the tricky problem. Either we have to estimate the functions \mathbf{g} on-line during training or we have to hope that the final term in Equation 6 does not interfere with Infomax performing ICA. This is one occasion where hoping actually works, because of the following *robustness conjecture*: Any super-Gaussian prior, $p_i(u_i)$, will suffice to extract super-Gaussian in-

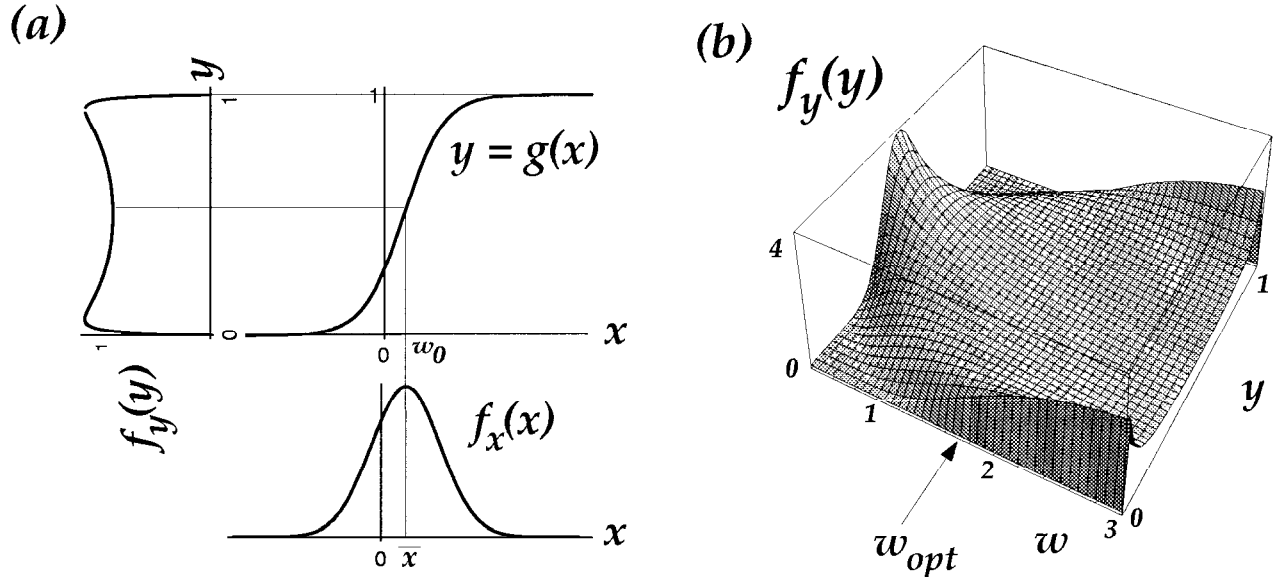


Figure 2. Optimal information flow in sigmoidal neurons. *A*, Input x , having probability density function $f_x(x)$ (note: this is $p(x)$ in the text), in this case a Gaussian, is passed through a nonlinear function $g(x)$. The information in the resulting density, $f_y(y)$, depends on matching the mean and variance of x to the threshold, w_0 , and slope, w , of $g(x)$ (Nicol Schraudolph, personal

communication). *B*, $f_y(y)$ is plotted for different values of the weight w . The optimal weight, w_{opt} , transmits most information. (From Bell, A. J., and Sejnowski, T. J., 1995, An information maximization approach to blind separation and blind deconvolution, *Neural Computat.*, 7:1129–1159. Reproduced with permission.)

dependent components. Any sub-Gaussian prior will suffice to extract sub-Gaussian independent components. Super-Gaussian means peakier than Gaussian, sometimes called *sparse*, and usually signifying positive kurtosis. There are no solid proofs yet of this conjecture; it is more something that has been empirically observed, but it leads to generally successful “extended ICA” algorithms (see Lee et al., 1998) that do on-line switching of the priors, $\hat{p}_i(u_i)$, between super- and sub-Gaussian functions. In practice, because of this robustness, this switching may be all the estimation we need to do. This is also the insight behind “negentropy” approaches to ICA, which maximize the distance of the $p_i(u_i)$ from Gaussian, as described in Hyvärinen (1999) and Lee et al. (1998).

For most natural data (images, sounds, etc.), the independent component p.d.f.s are all super-Gaussian, and many good results can be achieved with what is called logistic ICA, in which our super-Gaussian prior is the slope, $g'(u_i)$, of the common logistic sigmoid function (Equation 7) so often used in neural networks. For this choice of g , the function f in Equation 8 evaluates as $f(u) = 1 - 2g(u)$.

The Infomax-ICA algorithm is almost identical to the maximum likelihood approach (Pham et al., 1992). In maximum likelihood density estimation, we maximize a parameterized estimate of the log of the p.d.f. of the input, $\log \hat{p}(\mathbf{x}|\mathbf{W}, \mathbf{g})$. A simple argument shows that the determinant of the Jacobian matrix, $\det [\partial g_i(u_i)/\partial x_i]_{ij}$, is exactly such a density estimate (for much the same reason that $|g'_i(u_i)|$ is a density estimate for $p_i(u_i)$ in Equation 6). Infomax maximizes this log likelihood, and therefore inherits the useful properties of maximum likelihood methods, while preserving an information-theoretic perspective on the problem.

The final twist in the Infomax/maximum likelihood ICA algorithm comes from Amari and colleagues (Amari, 1997). They observed that a simpler learning rule with much faster and more stable convergence was obtained by multiplying the Infomax gradient of Equation 8 by $\mathbf{W}^T \mathbf{W}$ to obtain the following much simpler rule:

$$\Delta \mathbf{W}_{\text{NatGrad}} = (\Delta \mathbf{W}) \mathbf{W}^T \mathbf{W} \propto (\mathbf{I} + \mathbf{f}(\mathbf{u}) \mathbf{u}^T) \mathbf{W} \quad (10)$$

Since $\mathbf{W}^T \mathbf{W}$ is positive definite, it does not change the minima and maxima of the optimization, it just scales the gradient. As luck would have it, it scales the gradient in an optimal fashion, which may be explained by an appeal to information geometry (Amari, 1997) or to equivariance: the gradient vector local to \mathbf{W} is normalized so that it always behaves as it does when \mathbf{W} is close to \mathbf{I} . For explanations, see the relevant chapters in Haykin (2000).

Both interpretations reflect the fact that the parameter space of \mathbf{W} is not truly Euclidean, since its axes are entries of a matrix. Technically speaking, the parameter space has the structure of a Lie group.

Equation 10 can be clearly seen to be a nonlinear decorrelation rule, stabilizing when $\langle -\mathbf{f}(\mathbf{u}) \mathbf{u}^T \rangle = \mathbf{I}$ (the minus sign is there because the \mathbf{f} functions are typically decreasing). The Taylor series expansion of the \mathbf{f} functions provide information about higher-order correlations necessary to perform the ICA task.

Applications

The blind source separation problem in signal processing has often been considered synonymous with the ICA problem, but ICA can be applied in many situations where there is no clear notion of what constitutes a source. Some of the most interesting results have been achieved in such situations. Here we consider a few.

ICA on Natural Images

From the point of view of computer vision and computational neuroscience, perhaps the most interesting result was the ICA basis vectors obtained for a data set of small image patches drawn from natural images (Bell and Sejnowski, 1997). These basis vectors

consisted of oriented, localized, contrast-sensitive functions, sometimes referred to as edges (though an edge is really something to do with object boundaries).

Figure 3 shows a selection of basis functions (columns of \mathbf{W}) and filters (rows of \mathbf{W}) obtained from training on 18×18 patches.

The reason why this is interesting is that both the classic experiments of Hubel and Wiesel on orientation-selective neurons in visual cortex and several decades of theorizing about feature detection in vision have left open a question most succinctly phrased by Horace Barlow: “Why do we have edge detectors?” In other words, were there any coding principles that could have predicted the formation of localized, oriented receptive fields?

Barlow was the first to propose that our visual cortical feature detectors might be the end result of a *redundancy reduction* process, in which the activation of each feature detector is supposed to be as statistically independent from the others as possible. Algorithms based on second-order statistics had failed to give clear, robust local filters. In particular, the principal components of natural images are Fourier filters ranked in frequency, quite unlike oriented localized filters.

Several authors proposed projection pursuit-style approaches, culminating in Olshausen and Field’s (1997) demonstration of the self-organization of local, oriented receptive fields using a sparseness criterion.

By identifying sparseness with super-Gaussianity (which Olshausen and Field implicitly did), we can readily see why an Infomax/ICA net with the logistic nonlinearity for its $g_i(u_i)$ s would produce the filters that produced the sparsest activation distributions when passed over the images. These distributions, furthest from Gaussian on the super-Gaussian side, were the most likely to be as statistically independent as possible, through the central limit theorem argument that any mixture of two independent distributions produces a distribution that is closer to Gaussian. As stated before, it is remarkable that none of the independent components of natural images are sub-Gaussian. This has been verified by using the extended ICA algorithm.

The assumption implicit in both approaches has been that the first layer of visual processing should attempt to invert the simplest possible image formation process, in which the image is formed, just as in Equation 3, by linear superposition of basis vectors (columns of \mathbf{W}^{-1}), each activated by independent (or sparse) causes, u_i .

Impressive results have been obtained by van Hateren and Ruderman (1998) on the *basis movies* of moving images. This 1,728-dimensional transform (viewable at <http://hlab.phys.rug.nl/demos/ica>) is one of the largest ICA applications to date. The resulting spatiotemporal bases are localized, oriented, and moving perpendicular to their orientation direction, just as in monkey visual cortex. And significantly, there are many more with the much lower spatial frequency required to match the profile of monkey visual receptive fields. This helps to answer the complaint that the ICA bases of natural images were clustered around high-frequency (sharp) contrast filters.

Biomedical Applications

This section reports briefly on the work of S. Makeig, T.-P. Jung, M. McKeown, and their colleagues (see Jung et al., 2000, and BRAIN SIGNAL ANALYSIS).

Electroencephalographic (EEG) data are a measure of the brain’s electric fields, which are linearly mixed by volume conduction at scalp electrodes with negligible time delays and thus are perfect for ICA analysis. If there were 14 electrical dipoles in the brain, each with independently fluctuating charges, and 14 noiseless electrodes on the scalp, then the dipole signals could be perfectly recovered by ICA. This is, of course, not the case. But the results are interesting nonetheless.

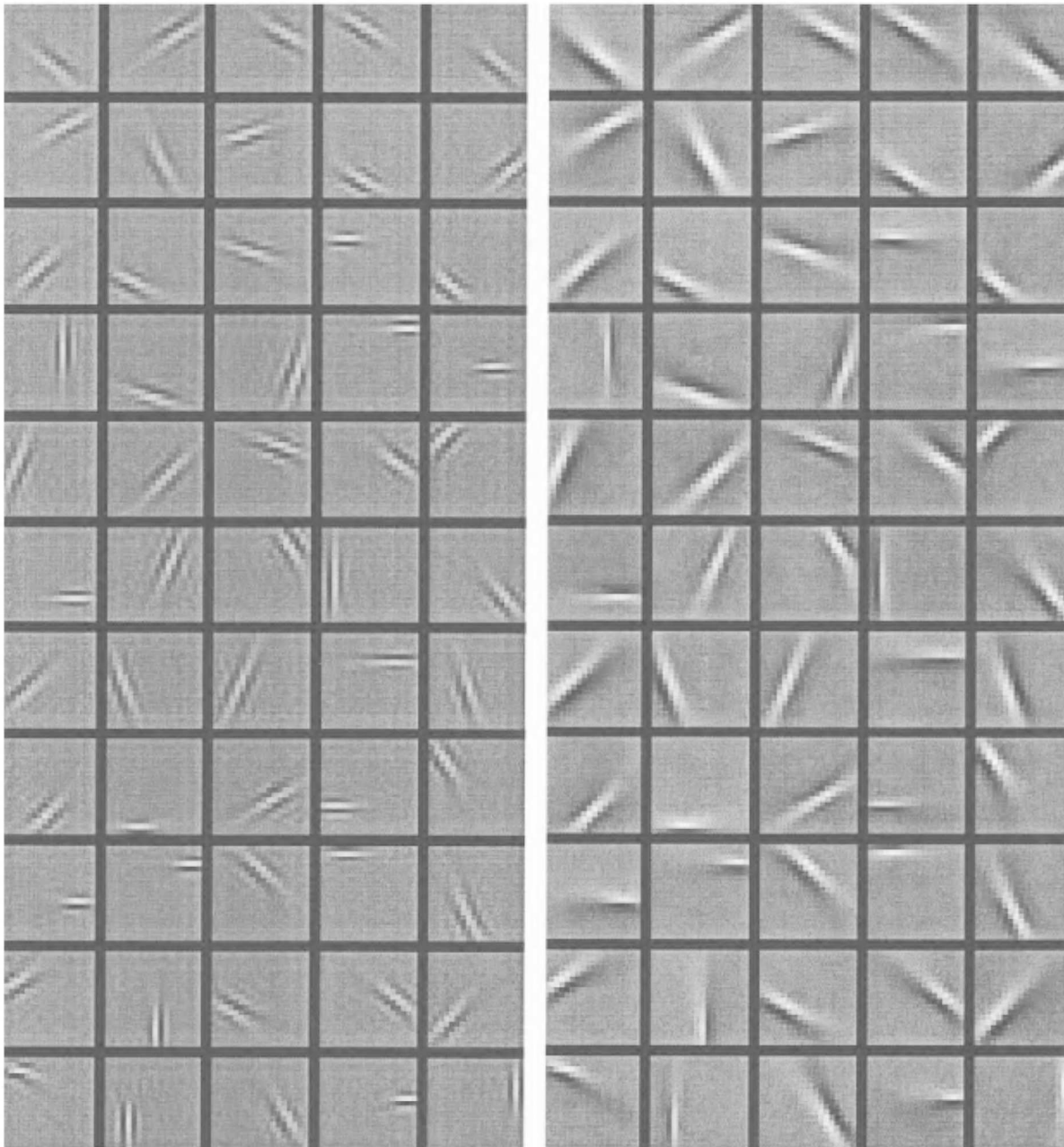


Figure 3. A selection of the 324 independent basis functions (left) and filters (right) obtained by training on 18×18 patches drawn from natural images. The results were obtained by van Hateren and van der Schaaf using Hyvärinen's (1999) FastICA method and are similar to those found using

Bell and Sejnowski's (1997) Infomax/ML-ICA. (From van Hateren, J. H., and vander Schaaf, *Proc. R. Soc. Lond. B*, 265:359–366. Reprinted with permission.)

As well as decomposing correlated alpha wave activity across electrodes into prominent rhythms in different components with different time courses, many ICA outputs are easy to identify with artifacts known to contaminate brainwave data. In Figure 4, five of these are displayed, corresponding to eye blinks, localized scalp muscle movements, 60-Hz electrical line noise, heartbeat, and a horizontal eye movement. Both their time courses and their scalp maps help support these interpretations. The EEG data can then be cleansed of these artifacts, by zeroing the columns in \mathbf{W}^{-1} corresponding to the artifacts and reconstructing the data using Equation 3.

Another kind of brain recording, functional magnetic resonance imaging (fMRI), monitors humans during the performance of psychomotor tasks and produces a three-dimensional picture of their brain activity with a spatial resolution of about 5 mm^2 and a temporal resolution of about 2 s. The subject is typically asked to alternate between performing a given task and a control task, so that researchers can identify which regions of the brain are differently activated. One of the independent components of the fMRI data studied by McKeown and colleagues was a pattern of spatial brain activation that was activated with a square-wave time course exactly corresponding to the execution of the task. This was true

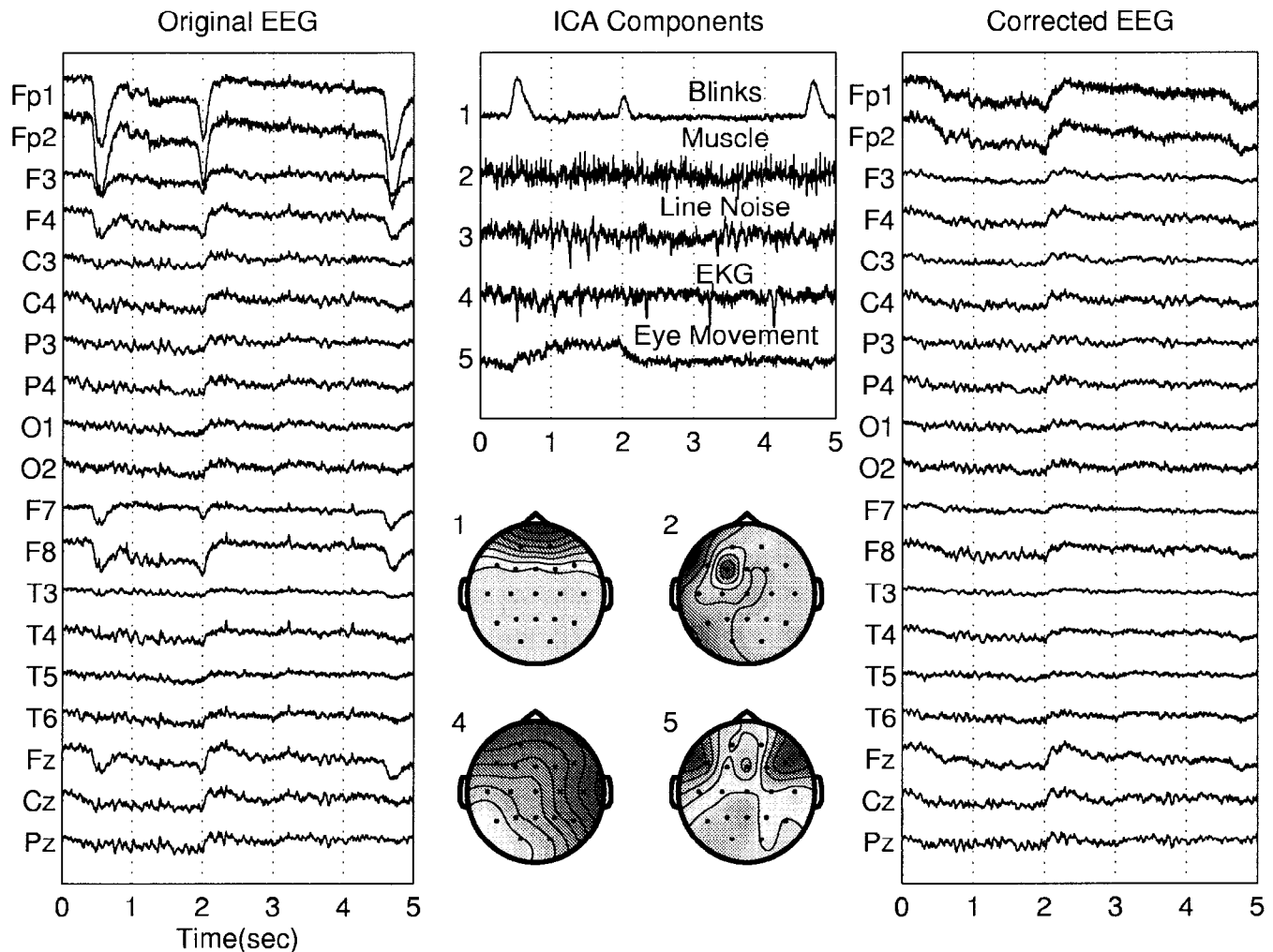


Figure 4. A 5-s portion of the EEG time series (left), ICA components accounting for eye movements, cardiac signals, and line noise sources (center), and the EEG signals corrected for artifacts by removing the five com-

ponents (right). (From Jung, T.-P., et al., 1998, *Adv. Neural Inf. Process. Syst.*, 10:894–900. Reprinted with permission.)

over six trials with two subjects. When the independent components were ranked in terms of their strength in the original signal, this time course was ranked between 14th and 41st out of 146, so it was by no means a strong signal in the original data. The brain map associated with it contained activations in relevant areas of the brain corresponding mainly to vision and visual association, but also some in a motor area, and some in prefrontal cortex.

In each case, in focusing on individual time courses, we are looking at on the order of 1/146th of the brain activation data. Thus, the vast majority of the possibly confusing and irrelevant brain activation is stripped away for us, because its brain maps are statistically independent from the ones that concern us.

Discussion

With these descriptions of the application of ICA to image processing and brain recordings, we hope to move the reader away from toy problems to a realization that algorithms of this kind have many applications. In almost every case, the algorithm has yielded surprises. Applications at the ICA 2000 workshop ranged from clustering World Wide Web documents to extracting the earthquake signal on the volcanic island of Stromboli.

On the theoretical side, ICA is in many ways much more interesting than its cousin, PCA, and this success comes from the simultaneous relaxation of two assumptions. The first is geometrical: that coordinate systems in signal transforms should be orthogonal. The second is statistical: that anything interesting in the probability distribution can be captured by the covariance matrix. Progress in this field will be made by further combined statisticogeometrical innovations, such as the identification of group-theoretic symmetries in probability distributions with the subspaces that they are embedded in, but that's another story.

Road Map: Learning in Artificial Networks

Related Reading: Brain Signal Analysis; Feature Analysis; Principal Component Analysis; Unsupervised Learning with Global Objective Functions

References

- Amari, S.-I., 1997, Natural gradient works efficiently in learning, *Neural Computat.*, 10:251–276.
- Bell, A. J., and Sejnowski, T. J., 1995, An information maximization approach to blind separation and blind deconvolution, *Neural Computat.*, 7:1129–1159.

- Bell, A. J., and Sejnowski, T. J., 1997, The "independent components" of natural scenes are edge filters, *Vision Res.*, 37:3327–3338.
- Cardoso, J.-F., and Laheld, B. H., 1996, Equivariant adaptive source separation, *IEEE Trans. Signal Process.*, 44:3017–3030.
- Cichocki, A., Unbehauen, R., and Rummert, E., 1994, Robust learning algorithm for blind separation of signals, *Electron. Lett.*, 30:1386–1387.
- Comon, P., 1994, Independent component analysis: A new concept? *Signal Process.*, 36:287–314.
- Girolami, M., Ed., 2000, *Advances in Independent Component Analysis*, New York: Springer-Verlag.
- Herauld, J., and Jutten, C., 1986, Space or time adaptive signal processing by neural network models, in *Neural Networks for Computing: AIP Conference Proceedings 151* (J. S. Denker, Ed.), New York: American Institute for Physics.
- Haykin, S., Ed., 2000, *Unsupervised Adaptive Filtering*, vol. 1: *Blind Separation*, New York: Wiley.
- Hyvärinen, A., 1999, Survey on independent component analysis, *Neural Comput. Surv.*, 2:94–128.
- Jung, T.-P., Makeig, S., Lee, T.-W., McKeown, M. J., Brown, G., Bell, A. J., and Sejnowski, T. J., 2000, Independent component analysis of biomedical signals, in *Proceedings of ICA 2000*.
- Lee, T.-W., Girolami, M., Bell, A. J., and Sejnowski, T. J., 1998, A unifying information-theoretic framework for independent component analysis, *Int. J. Math. Comput. Model.*, 31(11):1–21.
- Olshausen, B. A., and Field, D. J., 1997, Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.*, 37:3311–3325.
- Pham, D. T., Garrat, P., and Jutten, C., 1992, Separation of a mixture of independent sources through a maximum likelihood approach, in *Proceedings of the EUSIPCO*, pp. 771–774.
- van Hateren, J. H., and Ruderman, D. L., 1998, Independent component analysis of natural image sequences yields spatiotemporal filters similar to simple cells in primary visual cortex, *Proc. R. Soc. Lond. B*, 265:2315–2320.

Information Theory and Visual Plasticity

Nathan Intrator

Introduction

The relevance of information theory to neural networks has become more apparent in recent years. This theory has become important in analyzing and understanding the nature of the neuronal code that is relayed between cortical layers and the nature of the learning goals that guide neuronal learning and synaptic modification. Rapid advances in single- and multiple-electrode recording and other non-invasive techniques have provided a clearer view on neuronal activity and synaptic modification. However, the puzzle is still unsolved. We do not know what neuronal learning goals are, how they are being incorporated, and, most importantly, the nature of the neuronal code and how it is formed and interpreted by successive layers.

Information theory is an excellent tool that tells us how to code the information we want to relay. When studying the brain, we do not exactly know what the brain is coding at a local neuronal neighborhood and, thus, we cannot rely entirely on information theory to understand what neurons do. However, under various assumptions regarding the role of neuronal activity, one can test whether a certain code is optimal. When we try to understand the nature of synaptic learning rules, we should be concerned with possible goals that may underlie synaptic changes. It is conceivable that knowing what could be a useful goal under different input environments could serve for distinguishing between synaptic plasticity theories. Only after this distinction between goals has been achieved can one continue on and distinguish between learning rules aimed at achieving the same objective, on the basis of their detailed mathematical or computational properties. In this article, we briefly indicate possible neuronal goals resulting from various information-theoretic considerations. We are not making a statement about the nature of information relay in the brain or about the nature of information representation, where possible candidates may be action potentials, single spikes, spikes averaged along a time window of few milliseconds, or spike activity averaged over few cells.

For a book on information theory, see *Elements of Information Theory* (Cover and Thomas, 1991). Useful relevances to neuronal activity can be found in Rieke et al. (1996).

Brief Review of Information Theory

Information theory was developed about 50 years ago for the study of communication channels (Shannon, 1948). Shannon considered

information as a loss of *uncertainty* and defined it as a function of the probability distribution of the code words. If, for example, the probability distribution $P(X)$ is concentrated on a single value, then the information we can transmit, when choosing values from such distribution, is zero, since there is only one value to transmit. Thus, the amount of information is a function of the variability of the distribution and actually the exact shape of the distribution. This information quantity, which we denote by $H(X)$, should satisfy an additivity constraint, which states that when two random variables are independent, the information contained in both of them should be the sum of the information contained in each of them, namely

$$P(X_1, X_2) = P_1(X_1)P_2(X_2) \Rightarrow H(X_1, X_2) = H(X_1) + H(X_2) \quad (1)$$

Shannon has shown that the only function that is consistent with this condition (and with a few other simple constraints) is the Boltzmann entropy of statistical mechanics. The connection between information theory, statistics, and statistical mechanics is demonstrated in Jaynes (1957). The entropy in continuous and in discrete cases respectively is given by

$$\begin{aligned} H(X) &= - \int_K P(x) \log P(x) dx \\ H(X) &= - \sum_{i=1} p(x_i) \log p(x_i) \end{aligned} \quad (2)$$

where $p(x_i)$ is the probability of observing the value x_i out of possible K discrete values of the random variable X . (In information theory, it is customary to neglect the Boltzmann constant, which sets up the units correctly, and to use the logarithm of base 2, so that the information is measured in bits.) An intuitive way to look at this function is by considering the average number of bits that is needed to produce an efficient code. It is desirable to use a small number of bits for sending those words that appear with high probability and to use a larger number of bits for sending words that appear with lower probability. In the special case of n words arriving with the same probability, the number of bits that are required for each word is $\log_2 n$.

Shannon formulated this idea for the problem of information flow through a bottleneck, having to optimize the code so as to send the smallest number of bits on average. This led to questions such as how the receiver, given only the transmitted information,

maximizes his knowledge about the data available at the sender's end. For our purpose, we formulate the mutual information idea in terms of a neural network of a single layer. Let $\mathbf{d}^i \in R^n$ be an input vector to the network occurring with a probability distribution P_d , and let $\mathbf{c}^i \in R^k$ be the corresponding k -dimensional network activity with its probability distribution P_c . The *relative entropy* or the *Kullback-Leibler distance* between the two probability distributions is defined as

$$\begin{aligned} D(P_d \| P_c) &= \sum_i P_d(\mathbf{d}_i) \log \frac{P_d(\mathbf{d}_i)}{P_c(\mathbf{c}_i)} \\ &= E_{P_d}[\log(P_d) - \log(P_c)] \end{aligned} \quad (3)$$

Note that this is not symmetric and does not satisfy the triangle inequality.

Consider now the joint probability distribution of the input and output random variables $P(\mathbf{d}, \mathbf{c})$ such that P_d and P_c are the corresponding marginal distributions. The *mutual information* $I(\mathbf{d}, \mathbf{c})$ is the relative entropy between the joint distribution and the product distribution, namely,

$$\begin{aligned} I(\mathbf{d}, \mathbf{c}) &= D(P(\mathbf{d}, \mathbf{c}) \| P(\mathbf{d})P(\mathbf{c})) \\ &= \sum_{\mathbf{d}^i} \sum_{\mathbf{c}^j} P(\mathbf{d}^i, \mathbf{c}^j) \log \frac{P(\mathbf{d}^i, \mathbf{c}^j)}{P(\mathbf{d}^i)P(\mathbf{c}^j)} \\ &= \sum_{\mathbf{d}^i} \sum_{\mathbf{c}^j} P(\mathbf{d}^i, \mathbf{c}^j) \log \frac{P(\mathbf{d}^i | \mathbf{c}^j)}{P(\mathbf{d}^i)} \\ &= H(\mathbf{d}) - H(\mathbf{d} | \mathbf{c}) \end{aligned} \quad (4)$$

Additional properties of mutual information can be found in Cover and Thomas (1991). For example, $I(\mathbf{d}, \mathbf{c}) = H(\mathbf{c}) - H(\mathbf{c} | \mathbf{d})$, and $I(\mathbf{d}, \mathbf{c}) = H(\mathbf{c}) + H(\mathbf{d}) - H(\mathbf{d}, \mathbf{c})$.

By maximizing the mutual information, we effectively minimize $H(\mathbf{d} | \mathbf{c})$. Namely, we reduce the uncertainty about the input \mathbf{d} by knowing the output \mathbf{c} . Thus, given a constrained situation in which the output \mathbf{c} carries less data than the input \mathbf{d} , information theory tells us what the optimal output should be for a given input so as to have, on average, maximal knowledge about the input. Synaptic modification rules can be derived from solving the mutual information maximization problem under various assumptions about the probability distribution of the inputs. The solution to such a learning rule is based on gradient ascent, or other more sophisticated optimization algorithms.

Distributions That Maximize Entropy under Various Constraints

When we observe a certain distribution, it is natural to ask if this distribution represents a redundant coding or if it maximizes entropy under certain constraints. In some cases, we may be interested to recover the constraints under which the distribution maximizes the entropy. This section mentions some of the most common constraints and the distributions that are naturally connected with these constraints. Entropy maximization (or, as it is sometimes called, the MAX-ENT principle) is a powerful statistical inference tool. Given a certain set of constraints on a random variable, it suggests the *only possible* unbiased underlying distribution for the process. If the observed distribution is different, this implies that there are additional or different constraints governing the process. An excellent review with connection to statistical mechanics can be found in Jaynes (1957). Applications of this inference tool are many (see, e.g., Skilling, 1989).

Bounded distributions. The uniform distribution maximizes the entropy of a random variable with bounded values. Note that when discretizing a random variable, its distribution becomes automatically bounded, but it is the nondiscretized distribution that governs

the process. Thus, we would expect a maximal entropy distribution of 8-bit gray-level pictures to have a Gaussian and not a uniform distribution.

Positive valued random variables. Distributions that take only positive values or, more generally, that are bounded from below are a special and important case. They include, for example, distributions of spike counts over a certain measurement window. It turns out that under mean value constraint, the Poisson distribution maximizes the entropy. Under a variance constraint (of positive-valued distribution), the Gibbs distribution maximizes the entropy. This distribution occurs often when a quadratic functional (also called an energy or a Hamiltonian) can be associated with a configuration state of a physical system. A famous example is the annealing process (Brillouin, 1956) and a numerical algorithm called simulated annealing (Kirkpatrick and Gelatt, 1983).

Fixed variance constraint. Under a fixed mean and variance, an unbounded random variable has a normal distribution for entropy maximization. This makes a strong connection between minimizing entropy and searching for distributions that are far from Gaussian. It also shows that a linear layered network receiving Gaussian distributed inputs should extract the projections that maximize the variance, namely, find the principal components of the data in order to maximize the entropy of the projections.

Minimal Description Length

We have seen how information theory can suggest an optimal coding \mathbf{c} for a given input \mathbf{d} based on the probability distribution of the inputs. In this case, the code is transmitted through the bottleneck transmission line to the receiver, which then tries to reconstruct the original inputs. This formulation does not take into account the complexity of the code that is being sent and the complexity of constructing or decoding this code. A different information-theoretic formulation, one that is more appropriate for supervised learning, does take the above considerations into account. This formulation is based on the *minimal description length* (MDL) principle (Rissanen, 1984), which states that the way to choose a better model for data is by minimizing concurrently the cost of describing the model and the cost of describing the misfit between the model and the data. In terms of the information bottleneck (described earlier), we can view the current situation as a teacher-student network in which the teacher is trying to send the student the network to solve a certain problem. Under a supervised setup, the assumption is that both the student and the teacher can see the input data (zero cost), but only the teacher knows what the output should be. For the student to reconstruct the output, he would need to have a good model (network), namely, a small misfit between network output and desired output, and for quick learning, the model should be simple. Both of these properties can be measured by the entropy of sending the information about data misfit and about the model. In classical information theory, it is often assumed that the cost of learning a model can be neglected, as learning takes place only once, whereas data are sent continuously. However, when modeling learning, it is clear that the cost of learning plays an important role and should not be neglected. Hinton and colleagues have presented a formulation based on MDL principles for learning in neural networks (see Zemel and Hinton, 1995). It remains to be seen whether measuring the model (or learning) cost using the cost of sending the model parameters (entropy of the weight distribution) will turn out to be a useful constraint and a useful neuronal learning goal.

Note that measuring model cost by the entropy of its parameters may be radically different from measuring model cost by the actual

or effective number of parameters, as has been proposed before (Akaike, 1974).

Projection Pursuit and Cortical Plasticity

Thus far we have discussed maximization of entropy under various conditions. At times, however, minimization of entropy is more relevant. This occurs, for example, when classification is sought and small ambiguity of the outputs is desired. It also occurs when one is searching for independent components (Comon, 1994) and, especially, when one is looking for structure in the data by searching for interesting projections. To see the connection between independent components and the search for structure in the projections, we note that when two (or more) independent components are added together, their combined distribution is more Gaussian than is each of the component's distributions. Thus, seeking projections that are far from Gaussian can find the original distributions, assuming they were non-Gaussian to start with. More generally, the central limit theorem implies that given a list of independent random variables, their mean is normally distributed. Thus, a random projection of high-dimensional data would yield a single-dimensional Gaussian distribution unless there was a strong dependency between the projection vector and the data. A theoretical analysis of properties of projections was done by Diaconis and Freedman (1984). They have shown that for most high-dimensional clouds (of points), most low-dimensional projections are approximately Gaussian. Since entropy is maximized for a Gaussian distribution, searching for minimal entropy amounts to searching for maximal deviation from a Gaussian distribution. In practice, calculating the entropy is computationally expensive and requires knowledge (or robust estimation) of the projected distribution. Therefore, approximations to the entropy are used that rely on polynomial moments. The general framework of exploratory projection pursuit (Friedman, 1987) suggests various ways to seek such non-Gaussian projections. Its supervised version is called *projection pursuit regression* (Friedman and Stuetzle, 1981). It turns out that polynomial moments that are not approximating the entropy are also good candidates for measuring deviation from Gaussian distribution. For example, skewness and kurtosis, which are functions of the first four moments of the distribution, are frequently used in the search for independent components or non-Gaussian projections.

Intrator and Cooper (1992) have shown that a BCM neuron can find structure in the input distribution that exhibits deviation from Gaussian distribution in the form of multimodality in the projected distributions. Since clusters cannot be found directly in the data, owing to its sparsity, this type of deviation, which is measured by the first three moments of the distribution, is particularly useful for finding clusters in high-dimensional data, and thus is useful for classification or recognition tasks. This learning rule has been compared with skewness and kurtosis measures for the purpose of extracting simple-cell receptive fields from natural images (Blais et al., 1998).

Summary

We have demonstrated some features of information theory that are relevant to information relay in cortex. We have presented cases in which information theory considerations led people to seek methods for Gaussianizing the input distribution and, in other cases, led people to seek learning goals for non-Gaussian distributions. The MDL principle was presented as a learning goal that takes into account the complexity of the decoding network. In particular, we have made the connection of entropy-based methods, projection pursuit, and cortical plasticity. Further details on the extraction of information in visual cortex can be found in FEATURE ANALYSIS.

Road Maps: Learning in Artificial Networks; Neural Plasticity

Related Reading: Feature Analysis; Learning and Statistical Inference; Minimum Description Length Analysis; Unsupervised Learning with Global Objective Functions

References

- Akaike, H., 1974, A new look at the statistical model identification, *IEEE Trans. Autom. Control*, 19:716–723.
- Blais, B. S., Intrator, N., Shouval, H., and Cooper, L. N., 1998, Receptive field formation in natural scene environments: Comparison of single cell learning rules, *Neural Computat.*, 10:1797–1813.
- Brillouin, L., 1956, *Science and Information Theory*, New York: Academic Press.
- Comon, P., 1994, Independent component analysis: A new concept? *Signal Process.*, 36:287–314.
- Cover, T., and Thomas, J., 1991, *Elements of Information Theory*, New York: Wiley. (See especially Chapters 1 and 13.) ♦
- Diaconis, P., and Freedman, D., 1984, Asymptotics of graphical projection pursuit, *Ann. Statist.*, 12:793–815.
- Friedman, J. H., 1987, Exploratory projection pursuit, *J. Am. Statist. Assoc.*, 82:249–266. ♦
- Friedman, J. H., and Stuetzle, W., 1981, Projection pursuit regression, *J. Am. Statist. Assoc.*, 76:817–823.
- Intrator, N., and Cooper, L. N., 1992, Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions, *Neural Netw.*, 5:3–17. ♦
- Jaynes, E. T., 1957, Information theory and statistical mechanics: 1, *Phys. Rev.*, 106:620–630, 108:171–190. ♦
- Kirkpatrick, S., and Gelatt, C. D., 1983, Optimization by simulated annealing, *Science*, 220:671–680.
- Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W., 1996, *Spikes: Exploring the Neural Code*. Cambridge, MA: MIT Press.
- Rissanen, J., 1984, Universal coding, information, prediction, and estimation, *IEEE Trans. Inf. Theory*, 30:629–636.
- Shannon, C. E., 1948, A mathematical theory of communication, *Bell. Syst. Tech. J.*, 27:379–423, 623–656.
- Skilling, J., 1989, *Maximum Entropy and Bayesian Methods*, Dordrecht: Kluwer Academic. (See especially Chapters 1 and 13.) ♦
- Zemel, R. S., and Hinton, G. E., 1995, Developing population codes by minimizing description length, *Neural Computat.*, 7:549–564.

Integrate-and-Fire Neurons and Networks

Wulfram Gerstner

Introduction

Most biological neurons communicate by short electrical pulses called *action potentials* or *spikes*. In contrast to the standard neuron

model used in artificial neural networks, integrate-and-fire neurons do not rely on a temporal average over the pulses. In integrate-and-fire and similar spiking neuron models, the pulsed nature of the neuronal signal is taken into account and considered as potentially

relevant for coding and information processing. In contrast to more detailed neuron models, integrate-and-fire models do not explicitly describe the form of an action potential. Pulses are treated as formal events. This is no real drawback, since, in a biological spike train, all action potentials of a neuron have roughly the same form. The time course of an action potential, therefore, does not carry any information.

Integrate-and-fire and similar spiking neuron models are phenomenological descriptions on an intermediate level of detail. Compared to other SINGLE-CELL MODELS (q.v.), they offer several advantages. In particular, coding principles can be discussed in a transparent manner. Moreover, dynamics in networks of integrate-and-fire neurons can be analyzed mathematically. Finally, large systems with thousands of neurons can be simulated rather efficiently. Reviews of integrate-and-fire networks can be found in Maass and Bishop (1998) and in Gerstner and Kistler (2002).

Spiking Neuron Models

Integrate-and-Fire Model

In its simplest form, an integrate-and-fire neuron i consists of a resistor R in parallel to a capacitor C driven by an external current I_i . The voltage u_i across the capacitor is interpreted as the membrane potential. The voltage scale is chosen so that $u_i = 0$ is the resting potential. The temporal evolution of u_i is

$$\tau_m \frac{du_i}{dt} = -u_i + RI_i(t) \quad (1)$$

where $\tau_m = RC$ is the membrane time constant of the neuron.

Spikes are formal events. We say that neuron i has fired a spike if u_i reaches at a time $t = t_i^f$ a threshold ϑ . The form of the action potential is not described explicitly. Immediately after spike firing, the potential u_i is simply reset to a value $u_{\text{reset}} < \vartheta$. Integration of Equation 1 is then resumed with u_{reset} as the initial condition (Stein, 1967). Because the spatial structure of the neuron is neglected, such a model is also called a point model (see SINGLE-CELL MODELS).

In a network of neurons, the input I_i to neuron i is due to the spikes of presynaptic neurons j . Detailed models of synaptic input can be found in SYNAPTIC INTERACTIONS. In the simplest model of a synapse, each presynaptic spike arrival evokes a postsynaptic current with a standard time course α . The total input to neuron i is then

$$I_i = \sum_{j,f} w_{ij} \alpha(t - t_j^f) \quad (2)$$

where the sum runs over all firing times t_j^f of all presynaptic neurons. The factor w_{ij} is the synaptic efficacy of a connection from a presynaptic neuron j to a postsynaptic neuron i . Choices for the postsynaptic current include a delayed δ -pulse, $\alpha(s) = \delta(s - \Delta^{\text{ax}})$, or a double exponential, $\alpha(s) = [e^{-(s-\Delta^{\text{ax}})/\tau_1} - e^{-(s-\Delta^{\text{ax}})/\tau_2}] / (\tau_1 - \tau_2)$, where Δ^{ax} is the axonal transmission delay and τ_1, τ_2 are synaptic time constants.

Spike Response Model

The integrate-and-fire equation (Equation 1) with the synaptic current (Equation 2) can be integrated, either numerically or analytically. Since it is a linear equation, the analytical integration can be done for each term in the sum of Equation 2 separately. The total membrane potential is then the sum of all the postsynaptic potentials (PSPs) caused by presynaptic firing, plus the refractory effect of a negative reset potential. Given the last firing time \hat{t}_i of neuron i , the result of the integration is therefore of the form ($t > \hat{t}_i$)

$$u_i(t) = \eta(t - \hat{t}_i) + \sum_{j,f} w_{ij} \varepsilon(t - \hat{t}_i, t - t_j^f) \quad (3)$$

The next firing of i occurs if the membrane potential u_i approaches the threshold ϑ from below. Equation 3 defines the dynamics of the spike response model (SRM). It was introduced above as an integrated version of the integrate-and-fire model, but the SRM is in fact more general (Figure 1). The function η describes the action potential at \hat{t}_i and the spike afterpotential that follows. The function ε describes the voltage response of neuron i to a presynaptic spike at t_j^f . Let us suppose that the last spike of the postsynaptic neuron i was far back in the past ($t - \hat{t}_i \rightarrow \infty$). Then $\varepsilon(\infty, s)$ as a function of s describes the time course of the PSP caused by a presynaptic spike. If the postsynaptic neuron i has been active in the recent past, then a presynaptic spike is less effective in exciting a postsynaptic response. The first argument of $\varepsilon(t - \hat{t}_i, t - t_j^f)$ describes the dependence on the recent firing history of the postsynaptic neuron. With an appropriate choice of the functions ε and η , about 90% of the firing times of the Hodgkin-Huxley model with time-dependent input can be correctly predicted by the SRM, with a precision of ± 2 ms (Kistler, Gerstner, and van Hemmen, 1997). Moreover, the spatial structure of neurons with a linear dendritic tree can be incorporated by an appropriate choice of ε . For synapses that are farther out on the dendritic tree, the PSP, and hence the function ε , rise more slowly.

Noise

Biological neurons that are driven by a time-dependent intracellular current exhibit a reliable, (nearly) deterministic behavior, just as the models in Equations 1 or 3. On the other hand, neurons that are part of a cortical network emit spikes at irregular intervals. Since the exact spike times cannot be controlled by the experiment, the irregularity is interpreted as noise.

Formally, noise can be introduced into the integrate-and-fire model by adding a fluctuating input $\sigma \xi_i(t)$ on the right-hand side of Equation 1, where σ is a parameter controlling the amplitude of the noise and ξ is a normally distributed random variable with zero mean (see DIFFUSION MODELS OF NEURON ACTIVITY). In the presence of noise, we may ask the following question: Given the last firing time \hat{t}_i of neuron i and the input current $I_i(t')$ for $t' > \hat{t}_i$, what is the probability that the next spike occurs around time t ? The answer is given by the conditional interval distribution $P(\hat{t}_i, I(\cdot))$. The calculation of $P(\hat{t}_i, I(\cdot))$ for the diffusion model is equivalent to the solution of a first-passage-time problem. The general solution to this problem is not known.

Noise can also be introduced into spiking neuron models in a different manner. The voltage $u_i(t)$ is calculated according to Equation 1 or 3. Even before u_i reaches the threshold ϑ , neuron i may fire with an “escape rate,” $\rho(t)$, that depends on the momentary distance from threshold and possibly also on the current input I_i ,

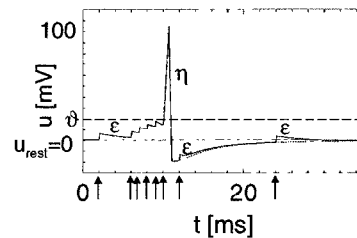


Figure 1. Each input current pulse (arrows) evokes a postsynaptic potential with time course ε . If the sum of the postsynaptic potentials reaches the threshold ϑ , an action potential with time course η is triggered. An input current pulse immediately after the action potential evokes a reduced response because of refractoriness.

viz., $\rho(t) = h(u(t) - \vartheta; I(t))$. In this case, an explicit expression for the conditional interval distribution is known, viz.,

$$P(d\hat{t}_i, I(\cdot)) = \rho(t) \exp \left[- \int_{\hat{t}_i}^t \rho(t') dt' \right] \quad (4)$$

With an appropriate choice of the escape function h , the diffusion model can be approximated by the escape model to a high degree of accuracy. For a review of noise models, see Gerstner and Kistler (2002, chap. 5).

Network Dynamics and Population Equations

In many areas of the brain, neurons are organized into groups of cells with similar properties, e.g., pools of motor neurons or columns in the visual cortex. Instead of looking at the firings of individual neurons, we may simply be interested in the fraction of neurons that are active in the population. In each small time window Δt , let us count the number of spikes $n_{sp}(t; t + \Delta t)$ that are emitted across the population, and divide by the number N of neurons and Δt . This procedure defines the population activity or population rate

$$A(t) = \lim_{\Delta t \rightarrow 0} \frac{n_{sp}(t; t + \Delta t)}{N \Delta t} = \frac{1}{N} \sum_{j,f} \delta(t - t_j^f) \quad (5)$$

where δ is the Dirac δ function and the sum runs over all spikes of all neurons in the population. The population activity has units of 1 over time and can be seen as the rate at which the total spike count increases. Note that the definition of the population rate (Equation 5) does not involve a temporal average, only a spatial average. What is the temporal evolution of $A(t)$ in a (homogeneous) network of spiking neurons?

The state of each neuron depends on its input *and* on the time \hat{t} of its last spike (see Equation 3). We define a *homogeneous* population by the conditions that (1) lateral coupling has a fixed value $w_{ij} = w_0/N$, and (2) external inputs $I^{\text{stim}}(t)$ are the same for all neurons. The total input to any neuron in the network is therefore

$$I(t) = w_0 \int_0^\infty \alpha(s) A(t - s) ds + I^{\text{stim}}(t) \quad (6)$$

Even though they all receive the same input, different neurons will, in general, have different firing times \hat{t} . A neuron that has fired its last spike at \hat{t} and has received an input $I(t')$ for $t' > \hat{t}$ will contribute with weight $P(t\hat{t}, I(\cdot))$ to the population activity at time t . Hence the expected value of the population activity at time t is

$$A(t) = \int_{-\infty}^t P(t\hat{t}, I(\cdot)) A(\hat{t}) d\hat{t} \quad (7)$$

For spiking neurons with escape noise $\rho(t)$, $P(t\hat{t}, I(\cdot))$ is given by Equation 4 and therefore is highly nonlinear. Equation 7 is implicitly contained in Wilson and Cowan (1972) and Knight (1972), and is formally derived in Gerstner (2000) for a homogeneous, fully connected network of spiking neurons in the limit of $N \rightarrow \infty$.

In their 1972 paper, Wilson and Cowan proposed transforming the integral Equation 7 into a differential equation of the form

$$\tau \frac{d}{dt} A(t) = -A(t) + g \left[w_0 \int_0^\infty \alpha(s) A(t - s) ds + I^{\text{stim}}(t) \right] \quad (8)$$

where τ is a time constant, w_0 is the neuronal coupling strength, $I^{\text{ext}}(t)$ is a stimulus, and g is a nonlinear transfer function. One of the problems of Equation 8 is that the time constant τ is the result of a process of “time coarse graining,” which is necessary for the transition from Equation 7 to Equation 8. Since the time window of coarse graining has to be defined somewhat arbitrarily, the time constant τ is basically ad hoc. Because of the problems inherent in Equation 8, it is preferable to work directly with Equation 7.

For the diffusion noise model, Equation 7 is valid but not very useful, because the conditional interval distribution $P(t\hat{t}, I(\cdot))$ is not known. As an alternative to Equation 7, the state of the population can be described by the distribution of membrane potentials $P(u, t)$ (Abbott and van Vreeswijk, 1993; Brunel, 2000; Nykamp and Tranchina, 2000). At each moment of time $P(u, t) \Delta u N$ gives the number of neurons in the population with a membrane potential between u and $u + \Delta u$. The equation of the integrate-and-fire model (Equation 1) with additive diffusion noise $\sigma \xi(t)$ can be transformed into a Fokker-Planck equation for the distribution of membrane potentials:

$$\tau \frac{\partial P(u, t)}{\partial t} = \frac{\sigma^2}{2\tau} \frac{\partial^2 P(u, t)}{\partial u^2} + \frac{\partial}{\partial u} \{ [u - RI(t)] P(u, t) \} \quad (9)$$

The threshold is treated as an absorbing boundary, so that the probability density vanishes for $u \geq \vartheta$. The probability current across threshold equals the population activity

$$A(t) = \frac{\sigma^2}{2\tau^2} \frac{\partial P(u, t)}{\partial u} \Big|_{u=\vartheta} \quad (10)$$

Since the membrane potential of active neurons is immediately reset to u_{reset} , the population activity $A(t)$ acts a source of probability current at $u = u_{\text{reset}}$. For a review, see Gerstner and Kistler (2002, chap. 6).

Application to Coding

Integrate-and-fire models can be used to discuss potential principles of coding and dynamics in a transparent manner (Maass and Bishop, 1998, chaps. 1, 2, 10–14). Before we turn to networks, let us start with two examples of coding on the single-neuron level.

Signal Encoding by Single Neurons

Coherent input is more efficient than incoherent spikes in driving a postsynaptic neuron. To see why, let us consider the SRM (Equation 3). For the sake of simplicity, we assume that the postsynaptic neuron i was inactive in the recent past ($t < 0$) and receives, for $t > 0$, input from two presynaptic neurons $j = 1, 2$, both firing at 100 Hz. We set $w_{i1} = w_{i2} = w_0$. According to Equation 3, each input spike evokes a postsynaptic potential $\varepsilon(-\infty, t - t_j^f)$, where t_j^f is one of the firing times of neuron j . If the two spike trains are out of phase, the summed postsynaptic potential is lower than in the synchronous case (Figure 2). By an appropriate choice of the threshold ϑ , an output spike of the postsynaptic neuron i occurs therefore only in the coherent (or “coincident”) case. Quite generally, coincidence detection is possible if the threshold of the postsynaptic neuron is slightly above the mean value, the membrane potential would take for asynchronous input (König, Engel, and Singer, 1996; Kempter et al., 1998). In the auditory system, it is

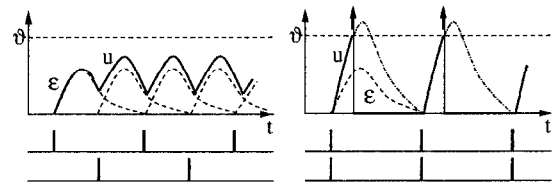


Figure 2. *Left*, Spike trains from two different presynaptic neurons are phase shifted with respect to each other. The summed potential u does not reach the threshold ϑ . *Right*, Spikes from the same presynaptic neurons arrive synchronously, so that u reaches the threshold ϑ and evokes the generation of output spikes (arrows). Afterwards, u is reset (schematic figure).

commonly accepted that coincidence detection is used for the localization of sound sources. On the other hand, it is an open question whether *cortical* neurons operate in the regime of coincidence detection (König et al., 1996; see also SINGLE-CELL MODELS).

Coding by Homogeneous Populations

Spiking neurons connected to each other by excitatory or inhibitory synapses exhibit nontrivial dynamical properties. The population may respond rapidly to external signals. The network activity may explode or die away. Neurons may spontaneously develop a tendency to fire synchronously or in groups. All of these phenomena, which can potentially be the basis of various coding schemes, can be understood from an analysis of Equations 6 through 10. Some of the fundamental questions are highlighted in the following.

First, is it possible, in the absence of an external stimulus, to stabilize a population of spiking neurons at a reasonable level of spontaneous activity? For $N \rightarrow \infty$, spontaneous activity corresponds to a stationary solution $A(t) \equiv A_0$ of the population dynamics described by Equation 7 or 10. Spontaneous asynchronous firing seems to be a generic feature of cortical tissue, but its role is still unclear. A stability analysis shows that without noise, asynchronous firing is never stable. Thus, the apparent noisiness of cortical neurons is a necessary feature of the system.

Even in the presence of noise, neurons often tend to synchronize their firings and develop collective oscillations. This observation leads to the second question: How is the frequency of collective oscillations related to neuronal parameters? It turns out that there are different oscillatory regimes, depending on the form of the post-synaptic potential, the axonal delay, and the value of the threshold (Abbott and van Vreeswijk, 1993; Brunel, 2000; Gerstner, 2000). The frequency of the collective oscillation may be low (about that of individual neurons) or several times faster. Collective oscillations and synchronization (Maass and Bishop, 1998, chaps. 10 and 11; Gerstner and Kistler, 2002, chap. 12) have been suggested as potential coding schemes in cortex and hippocampus (see SYNCHRONIZATION, BINDING AND EXPECTANCY).

Third, how rapidly does the population activity $A(t)$ respond to changes in the input? An analysis of Equation 7 shows that the response time is not limited by the membrane time constant of the neurons, but can be much faster (Gerstner, 2000). The fast response is due to the fact that, during spontaneous activity, there are always some neurons with a membrane potential just below threshold. A slight increase in the input will make those neurons fire immediately. The fast response of populations of spiking neurons to a new input could be important for an explanation of reaction time experiments (Thorpe, Fize, and Marlot, 1996; cf. FAST VISUAL PROCESSING). The same type of argument also shows that populations of spiking neurons can reliably transmit signals that vary on a time scale that is short compared to the interspike intervals of a neuronal spike train, as is the case in the auditory pathway, for example.

All of the above results hold true for homogeneous networks with either excitatory or inhibitory coupling. Formally, the theory is valid for full connectivity in the limit of $N \rightarrow \infty$. It also yields an excellent approximation for networks with random connectivity if the density of connections is either very high or very low. An extension to mixed excitatory/inhibitory populations as found in the cortex is possible (Brunel, 2000).

Coding in Structured Networks

Structure in neuronal networks may arise from a spatial arrangement of neurons or from specific patterns stored in a distributed manner in the network.

In networks with local (or distance-dependent) excitatory connections, traveling waves may occur. In two-dimensional sheets of

neurons, wave fronts may have planar or spiral shapes, similar to the ones found in reaction-diffusion systems. Collective oscillations and asynchronous firing are other possible network states. These effects can be described by a direct generalization of the theory of homogeneous systems to a spatially continuous population. Replace $A(t)$ in Equation 7 by $A(x, t)$ where x is the spatial location. Instead of Equation 6, we use $I(x, t) = \int dx' w(|x - x'|) \int ds \alpha(s) A(x', t - s)$, where $w(\cdot)$ is the distance-dependent coupling strength. Activity waves have been reported in slice cultures. It has also been suggested that similar activity waves could account for some of the trial-to-trial variability in cortical spike train recordings.

In the previous example, neurons that are strongly connected are located next to each other. Activity spreads from one group of neurons to its neighbors, which is easily recognizable by an external observer as a traveling wave of activity. Let us now keep the connections between the same neurons as before, but move all neurons to a new random location on the two-dimensional sheet. Apart from the fact that connection lines are longer, nothing has changed. What used to be a propagating wave in the original spatial arrangement now looks like asynchronous firing of neurons all over the sheet. Nevertheless, it is a specific, nearly deterministic spatiotemporal spike pattern. These “hidden” waves of activity have been termed SYNFIRES CHAINS (q.v.) (Abeles, 1991). Although the existence and stability of synfire chains can be shown by simulation or analysis of model networks, this does not necessarily imply that real brains make use of synfire chains for coding.

Discussion

What is the code used by cortical neurons? What is signal, what is noise in neuronal spike trains? Although the final answers to these questions have to come from additional experiments, modeling on the level of integrate-and-fire networks can contribute to answering, because models allow researchers to explore potential coding schemes and to identify relevant operating regimes.

In populations of integrate-and-fire neurons, a rate code can be a very fast code, if rate is defined by a population average (“population activity”) rather than by a temporal average (Knight, 1972; Gerstner, 2000). In contrast to widespread belief, the speed of signal transmission is not limited by the membrane time constant of the neuron. Moreover, with appropriate spike-based learning rules (Maass and Bishop, 1998, chap. 14), spiking neurons can work, in principle, at a very high temporal precision (Abeles, 1991). Large-scale simulations of integrate-and-fire networks provide a link between theory and experiments.

One of the points that has been stressed in recent models of integrate-and-fire neurons in the relevance of the subthreshold regime. If neuronal and network parameters are chosen so that the mean membrane potential stays just below threshold, then several interesting properties emerge. First, neurons act as coincidence detectors. They are sensitive to fluctuations in the input and can therefore “read out” the coherent aspects of the input signal (König et al., 1996; Kempter et al., 1998). Second, neurons in this regime respond rapidly to changes in the input (Gerstner, 2000). This might be relevant for explaining fast reaction times (Thorpe et al., 1996). Third, to stabilize a highly recurrent network of spiking neurons in the subthreshold regime, a certain amount of noise is necessary (Abbott and van Vreeswijk, 1993; Gerstner, 2000). From that point of view, it comes as no surprise that cortical neurons appear to be noisy. Whether this apparent noisiness is due to intrinsic noise sources in the neuronal dynamics, to noise in the synaptic transmission, or to deterministic chaos in a network is not clear. Model studies have shown that noise itself can arise as a network effect if neurons are in the subthreshold regime. Although individual neurons behave more or less deterministically, the same

neurons show large firing variability when part of a random network of excitatory and inhibitory neurons with sparse connectivity (Brunel, 2000). Such networks can represent past input in their spatiotemporal firing pattern (see TEMPORAL INTEGRATION IN RECURRENT MICROCIRCUITS). Thus, the study of integrate-and-fire networks may shed new light on the burning questions of brain theory.

Road Maps: Biological Networks; Neural Coding

Background: Single-Cell Models

Related Reading: Pattern Formation, NeuralRate Coding and Signal ProcessingSpiking Neurons, Computation with

References

- Abbott, L. F., and van Vreeswijk, C., 1993, Asynchronous states in a network of pulse-coupled oscillators, *Phys. Rev. E*, 48:1483–1490.
- Abeles, M., 1991, *Corticonics*, Cambridge, Engl.: Cambridge University Press.
- Brunel, N., 2000, Dynamics of sparsely connected networks of excitatory and inhibitory neurons, *Computat. Neurosci.*, 8:183–208. ♦
- Gerstner, W., 2000, Population dynamics of spiking neurons: Fast transients, asynchronous states and locking, *Neural Computat.*, 12:43–89.
- Gerstner, W., and Kistler, W. M., 2002, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge, Engl.: Cambridge University Press. ♦
- Kempter, R., Gerstner, W., van Hemmen, J. L., and Wagner, H., 1998, Extracting oscillations: Neuronal coincidence detection with noisy periodic spike input, *Neural Computat.*, 10:1987–2017.
- Kistler, W. M., Gerstner, W., and van Hemmen, J. L., 1997, Reduction of Hodgkin-Huxley equations to a single-variable threshold model, *Neural Computat.*, 9:1015–1045.
- Knight, B. W., 1972, Dynamics of encoding in a population of neurons, *J. Gen. Physiol.*, 59:734–766.
- König, P., Engel, A. K., and Singer, W., 1996, Integrator or coincidence detector? the role of the cortical neuron revisited, *TINS*, 19:130–137. ♦
- Maass, W., and Bishop, C., Eds., 1998, *Pulsed Neural Networks*, Cambridge, MA: MIT Press. ♦
- Nykamp, D., and Tranchina, D., 2000, A population density approach that facilitates large-scale modeling of neural networks: Analysis and application to orientation tuning, *J. Computat. Neurosci.*, 8:19–50.
- Stein, R. B., 1967, Some models of neuronal variability, *Biophys. J.*, 7:37–68.
- Thorpe, S., Fize, D., and Marlot, C., 1996, Speed of processing in the human visual system, *Nature*, 381:520–522.
- Wilson, H. R., and Cowan, J. D., 1972, Excitatory and inhibitory interactions in localized populations of model neurons, *Biophys. J.*, 12:1–24.

Invertebrate Models of Learning: *Aplysia* and *Hermissenda*

John H. Byrne and Terry Crow

Introduction

Certain invertebrates lend themselves to the study of learning and memory because of their relatively simple central nervous systems. In many cases, a fairly complete “wiring diagram” can be specified and modeled. Many neurons are relatively large and can be uniquely identified, which permits the examination of the functional properties of an individual cell and the ability to correlate those properties with a specific behavior mediated by the cell. Biophysical and molecular events underlying the changes in cellular properties can then be elucidated and mathematically modeled. This chapter summarizes the progress that has been made toward a mechanistic analysis of learning in the gastropod mollusks *Aplysia* and *Hermissenda*.

Nonassociative Modifications of Defensive Siphon and Tail Withdrawal Reflexes in *Aplysia*

Behaviors and Neural Circuits

The siphon-gill and tail-siphon withdrawal reflexes of *Aplysia* have been used to analyze the neuronal mechanisms contributing to non-associative and associative learning (see Carew, 2000; Kandel, 2001; Byrne, 2002). The siphon-gill withdrawal reflex is elicited when a stimulus is delivered to the siphon and results in withdrawal of the siphon and gill (Figure 1A). The tail-siphon withdrawal reflex is elicited by stimulation of the tail, which results in a coordinated set of defensive responses composed of a reflex withdrawal of the tail and the siphon (Figure 1B).

Defensive reflexes in *Aplysia* exhibit three forms of nonassociative learning: habituation, dishabituation, and sensitization. Habituation is defined as a decrement in response caused by repeated delivery of a stimulus (see HABITUATION). Dishabituation is defined as a restoration of a habituated (decremented) response by

delivery of another stimulus. Finally, sensitization is defined as an enhancement of a nondecremented response by delivery of another stimulus to the animal. With repeated stimulation, the reflexes undergo both short-term (minutes) and long-term (days) habituation. Applying a noxious stimulus to the head or tail can produce restoration of a habituated response (dishabituation) or sensitization of a nonhabituated response. Short-term sensitization lasts minutes, whereas long-term sensitization lasts days to weeks depending on the type of sensitization training. Although not described here, *Aplysia* also exhibits forms of associative learning such as classical conditioning and operant conditioning, which have been analyzed at the mechanistic level. Some of these mechanisms have been mathematically modeled and simulated by using a series of coupled ordinary differential equations (Byrne et al., 1990).

The afferent limb of the siphon-gill withdrawal reflex consists of sensory neurons with somata in the abdominal ganglion. The sensory neurons monosynaptically excite gill and siphon motor neurons, which are also located in the abdominal ganglion. Excitatory, inhibitory, and modulatory interneurons in the withdrawal circuit have also been identified.

The afferent limb of the tail-siphon withdrawal reflex consists of a cluster of 200 sensory neurons located in the pleural ganglion. These sensory neurons make monosynaptic excitatory connections with motor neurons in the adjacent pedal ganglion (Figure 2). The motor neurons produce withdrawal of the tail. In addition to their connections with tail motor neurons, sensory neurons form synapses with various identified interneurons. Some of these interneurons provide a parallel pathway to activate the tail motor neurons. These same interneurons activate motor neurons in the abdominal ganglion that control reflex withdrawal of the siphon (Figure 2). Several additional neurons modulate the reflex (not shown in Figure 2). Aspects of the neural circuit controlling tail withdrawal and its plasticity have been mathematically modeled and simulated us-

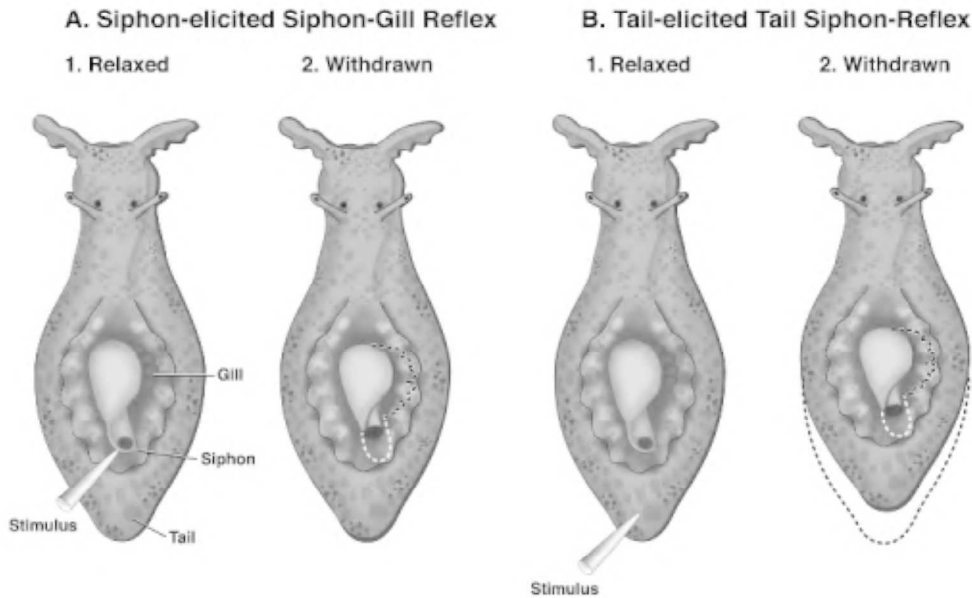


Figure 1. Siphon-gill and tail-siphon withdrawal reflexes of *Aplysia*. *A*, Siphon-gill withdrawal. Dorsal view of *Aplysia*. (1) Relaxed position. (2) A stimulus (e.g., a water jet, brief touch, or weak electric shock) applied to the siphon causes the siphon and the gill to withdraw into the mantle cavity. *B*, Tail-siphon withdrawal reflex. (1) Relaxed position. (2) A stimulus (e.g., touch or weak electric shock) applied to the tail elicits a reflex withdrawal of the tail and siphon.

ing the conductance-based simulator package SNNAP (White et al., 1993; see also NEUROSIMULATION: TOOLS AND RESOURCES).

The sensory neurons for both the siphon-gill and tail-siphon withdrawal reflexes are similar and appear to be key plastic elements in the neural circuits. Changes in their membrane properties and synaptic efficacy are associated with sensitization and the procedures that mimic short- and long-term sensitization training (see below).

Cellular Mechanisms in Sensory Neurons Associated with Short- and Long-Term Sensitization in Aplysia

Short-term sensitization. Short-term sensitization is induced when a single brief train of shocks to the body wall results in the release

of modulatory transmitters such as serotonin (5-HT) from facilitatory neurons that innervate the sensory neurons. The binding of 5-HT to receptors activates adenyl cyclase, raising the level of the second messenger cAMP in sensory neurons. The increase in cAMP activates cAMP-dependent protein kinase (protein kinase A, PKA), which adds phosphate groups to specific substrate proteins and consequently alters their functional properties. One result of this protein phosphorylation is an alteration of the properties of membrane channels. Specifically, the increased levels of cAMP lead to a modulation of the S-K⁺ current ($I_{K,S}$), the delayed K⁺ channel ($I_{K,V}$), and the calcium-activated K⁺ current ($I_{K,Ca}$). These changes in membrane currents lead to depolarization of the membrane potential, enhanced excitability, and an increase in the duration of the action potential. Cyclic AMP also appears to activate a membrane-potential- and spike-duration-independent process of facilitation, which may be due to the translocation of transmitter vesicles from a storage pool to a releasable pool. This process would result in more vesicles available for release with subsequent action potentials in the sensory neuron. These combined effects contribute to the short-term cAMP-dependent enhancement of transmitter release. Serotonin also appears to act through another receptor to increase the level of second messenger diacylglycerol (DAG), which in turn activates protein kinase C (PKC). Like PKA, PKC modulates the delayed K⁺ channel ($I_{K,V}$) and activates the spike-duration-independent process of facilitation. This modulation of $I_{K,V}$ also contributes to the increase in duration of the action potential. Serotonin can also activate mitogen-activated protein kinase (MAPK). One substrate for MAPK is the synaptic-vesicle-associated protein synapsin, which tethers synaptic vesicles to cytoskeletal elements and thus helps to control the reserve pool of vesicles in synaptic terminals. Of general significance is the observation that a single modulatory transmitter (i.e., 5-HT) activates at least three kinase systems. The consequences of the activation of these multiple messenger systems and multiple modulations of cellular processes occur when test stimuli elicit action potentials in the sensory neuron at various times after the presentation of the sensitizing stimuli. The enhanced release of transmitter from the sensory neuron leads to an enhanced activation of follower interneurons and motor neurons and an enhanced behavioral response (i.e., sensitization).

Aspects of the modulation of membrane channels and the dynamics of second messenger systems, calcium regulation, and

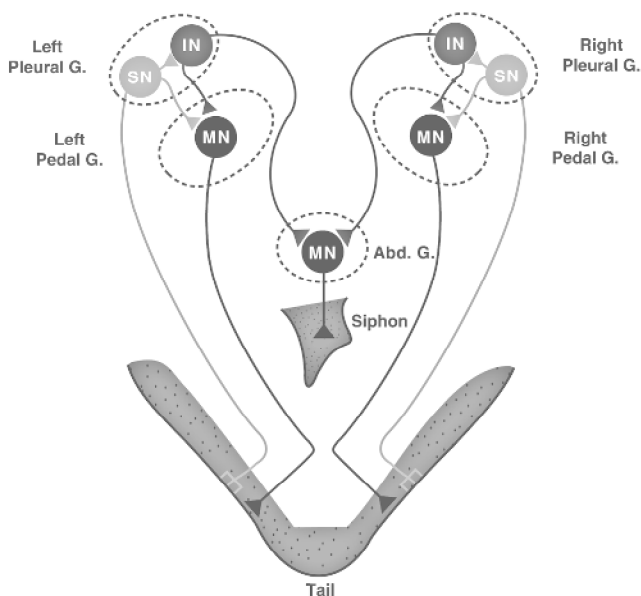


Figure 2. Simplified circuit diagram of the tail-siphon withdrawal reflex (see text for details).

transmitter storage and release have been mathematically modeled and simulated (Gingrich and Byrne, 1987; Baxter et al., 1999). The details of these biophysical and biochemical processes were necessary to simulate the features of the empirical data, which could not be captured by less detailed models (see Baxter and Byrne, 1993).

Long-term sensitization. Repetition of the sensitizing stimuli leads to the induction of long-term facilitation. Repeated training leads to a translocation of PKA to the nucleus, where it phosphorylates the transcriptional activator CREB1 (cAMP-responsive element-binding protein). CREB1 binds to a regulatory region of genes known as the cAMP-responsive element (CRE). Next, this bound and phosphorylated form of CREB1 leads to increased transcription. cAMP also leads to the activation of MAPK, which phosphorylates the transcriptional repressor CREB2. Phosphorylation of CREB2 by MAPK leads to a derepression of CREB2 and therefore promotes CREB1-mediated transcriptional activation. The combined effects of activation of CREB1 and derepression of CREB2 lead to changes in the synthesis of specific proteins.

One protein whose synthesis is regulated in this manner is *Aplysia* tolloid/BMP-like protein (ApTBL-1). Tolloid and the related molecule BMP-1 appear to function as secreted Zn^{2+} proteases. In some preparations, they activate members of the transforming growth factor β (TGF- β) family. Indeed, in sensory neurons, TGF- β mimics the effects of 5-HT in that it produces long-term increases in synaptic strength of the sensory neurons. Interestingly, TGF- β activates MAPK in the sensory neurons and induces its translocation to the nucleus. Thus, ApTBL-1 and TGF- β could be part of an *extracellular* positive feedback loop possibly leading to another round of protein synthesis to further consolidate the memory.

Prolonged stimulation and increased cAMP also activate a process that decreases the level of PKA regulatory subunits, further prolonging PKA activation (Greenberg et al., 1987). With fewer regulatory subunits to bind to catalytic subunits, the catalytic units would be persistently active and could contribute to long-term facilitation of transmitter release through the same cAMP-dependent processes that are seen in the short term. Some of these cAMP-PKA-induced changes include a decrease in $I_{K,S}$ and enhanced excitability, perhaps as well as a change in the synthesis of an $I_{K,S}$ channel protein or protein associated with the channel.

The downregulation of a homolog of a neuronal cell adhesion molecule (NCAM) ApCAM also plays a key role in long-term facilitation. This downregulation has two components. First, the synthesis of ApCAM is reduced. Second, preexisting ApCAM is internalized via increased endocytosis. The internalization and degradation of ApCAM allow for the restructuring of the axon arbor. The sensory neuron can now form additional connections with the same postsynaptic target or make new connections with other cells. As with short-term sensitization, the enhanced release of transmitter from existing contacts of sensory neurons onto motor neurons and interneurons contributes to the enhanced long-term responses of the animal to test stimuli (i.e., sensitization). However, unique to long-term sensitization, increases in axonal arborization and synaptic contacts may contribute to the enhanced activation of follower interneurons and motor neurons (e.g., Figure 2).

In addition to the cellwide changes in protein synthesis described above, recent work by Martin, Kandel, and their colleagues indicates that protein synthesis also occurs at the sites of synaptic contacts between sensory neurons and motor neurons. This local protein synthesis appears to be important in synapse-specific changes in synaptic efficacy.

Other temporal domains for the memory of sensitization. Historically, memory has been divided into two temporal domains: short term and long term. It has become increasingly clear from stud-

ies of a number of memory systems that this distinction is overly simplistic. For example, in *Aplysia*, Carew and his colleagues and Kandel and his colleagues have recently discovered an intermediate phase of memory that has distinctive temporal characteristics and a unique molecular signature. The intermediate-phase memory for sensitization is expressed at times approximately 30 minutes to three hours after the beginning of training. It declines completely prior to the onset of long-term memory. Like long-term sensitization, its induction requires protein synthesis, but unlike long-term memory, it does not require mRNA synthesis. The expression of the intermediate-phase memory requires the persistent activation of PKA.

Associative Learning in *Hermisenda*

Pavlovian Conditioning

Pavlovian (or classical) conditioning of *Hermisenda* involves changes in light-elicited locomotion and foot length (conditioned responses, CRs) produced by stimulation of the visual and vestibular systems with their adequate stimuli (see Sahley and Crow, 1998). The Pavlovian conditioning procedure consists of pairing light, the conditioned stimulus (CS), with high-speed rotation, the unconditioned stimulus (US). After conditioning, the CS suppresses normal light-elicited locomotion and elicits foot shortening (see Figure 3). Retention of conditioned behavior persists for several days to weeks depending on the number of conditioning trials used in initial acquisition (Alkon, 1989; Sahley and Crow, 1998). Pavlovian conditioning in *Hermisenda* exhibits CS specificity and is dependent on the association of the two sensory stimuli involving both contiguity and contingency. Nonassociative contributions to behavior are expressed in the initial trials of the conditioning session and decrement rapidly following the termination of multiple-trial conditioning. In addition to multiple-trial conditioning of suppression of light-elicited locomotion and foot contraction, one-trial conditioning also modifies light-elicited locomotion (Crow and Forrester, 1986). Pairing the CS with direct application of 5-HT (nominal US) to the exposed nervous system of otherwise intact *Hermisenda* produces suppression of light-elicited locomotion when the animals are tested 24 hours after the one conditioning trial. One-trial conditioning also produces enhanced excitability of type B photoreceptors (see Figure 4), a component of the CS pathway that expresses cellular plasticity produced by multiple-trial Pavlovian conditioning (see below).

Cellular and Synaptic Plasticity Associated with Pavlovian Conditioning

Certain sites of intrinsic modifications of cellular and synaptic plasticity in classically conditioned animals are associated with both enhanced excitability and synaptic facilitation, which have been localized to the primary sensory neurons (photoreceptors) of the pathway mediating the CS (Alkon, 1989; Fryszak and Crow, 1994). Enhanced excitability in identified photoreceptors of conditioned *Hermisenda* is expressed by a significant increase in spike activity elicited by the CS or extrinsic current, an increase in the input resistance, an alteration in the amplitude of light-elicited generator potentials, decreased spike frequency accommodation, and a reduction in the peak amplitude of voltage-dependent (I_A , I_{Ca}) and Ca^{2+} -dependent ($I_{K,Ca}$) currents (for reviews, see Alkon, 1989; Sahley and Crow, 1998). The enhanced excitability, expressed by an increase in both the amplitude of CS-elicited generator potentials and the number of action potentials elicited by the CS, may be a major contributor to changes in the duration and amplitude of CS-elicited complex postsynaptic potentials (PSPs) and enhanced CS-elicited spike activity observed in postsynaptic targets. How-

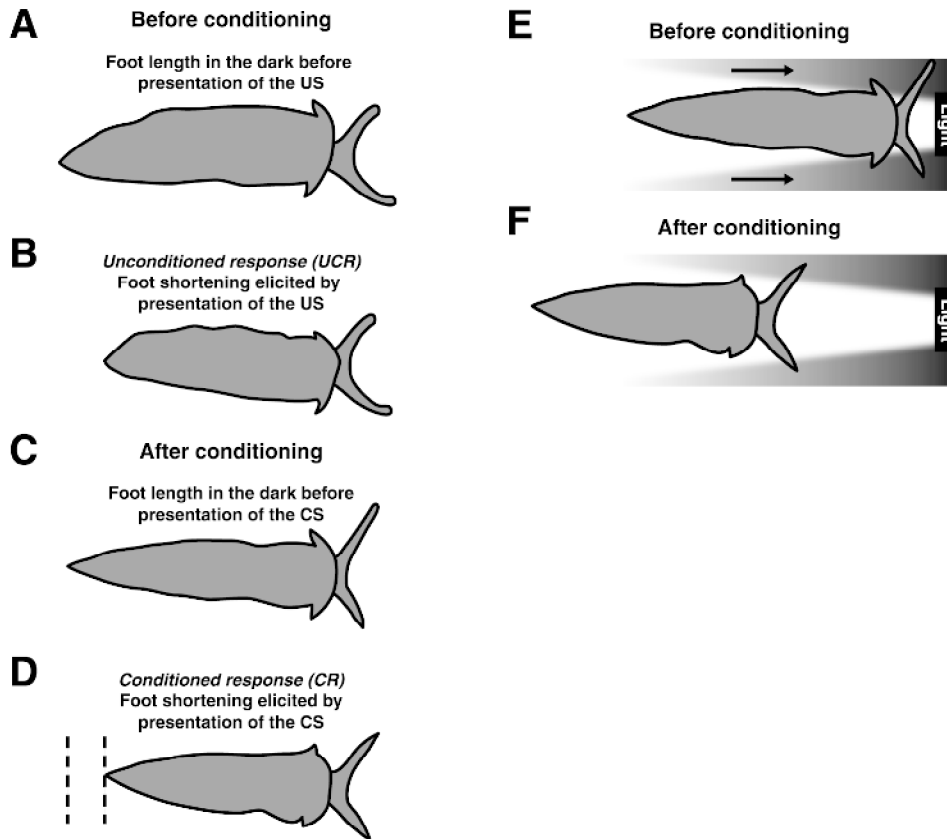


Figure 3. Pavlovian conditioned foot shortening and conditioned suppression of light-elicited locomotion of *Hermisenda*. *A*, Foot length in the dark before presentation of the unconditioned stimulus (US). *B*, The unconditioned response (UCR) elicited by rotation (US) of the animal in the dark. *C*, Foot length in the dark after Pavlovian conditioning and before presentation of the light (CS). *D*, Conditioned response (CR), foot shortening elicited by presentation of the CS. The area indicated between the dashed lines represents the magnitude of foot shortening elicited by the CS after conditioning. *E*, Light-elicited locomotion toward a light source assessed before conditioning. *F*, Suppression of light-elicited locomotion detected after Pavlovian conditioning. Pseudorandom or random presentations of the CS and US do not result in the development of suppression of either light-elicited locomotion or CS-elicited foot shortening.

ever, changes in the strength of synaptic connections between type B photoreceptors and other components of the CS pathway have also been detected following conditioning. Facilitation of the amplitude of monosynaptic inhibitory postsynaptic potentials (IPSPs)

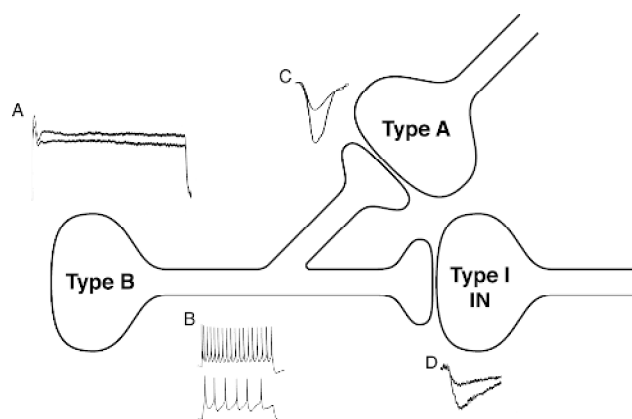


Figure 4. Components of the CS pathway that express plasticity in conditioned *Hermisenda*. *A*, The CS elicits a larger amplitude generator potential (upper trace) recorded from type B photoreceptors of conditioned animals as compared to pseudorandom controls (lower trace). *B*, An extrinsic current pulse elicits more action potentials in type B photoreceptors from conditioned preparations as compared to pseudorandom controls. *C*, Conditioning results in facilitation of the synaptic connections between type B photoreceptors and type A photoreceptors and type B photoreceptors and type I interneurons (*D*) as compared to control animals that received pseudorandom presentations of the CS and US.

elicited by single spikes in type B photoreceptors is detected in type A photoreceptors and type I interneurons of conditioned animals (Figures 4C and 4D). A second site of cellular plasticity in conditioned animals is the type A photoreceptor. Lateral type A photoreceptors of conditioned animals exhibit an increase in CS-elicited spike frequency, a decrease in generator potential amplitude, and enhanced excitability and decreased spike frequency accommodation to extrinsic current (for a review, see Sahley and Crow, 1998). The evidence for localization of cellular changes in the CS pathway indicates that multiple sites of plasticity involving changes in excitability and synaptic strength exist in the type B photoreceptors of conditioned animals (see Figure 4). Anatomical studies of type B photoreceptors indicate the existence of spatially segregated compartments (Alkon, 1989). Phototransduction occurs in the soma-rhabdomeric compartment, spike generation in the distal axon, and synaptic interactions in the axon terminal regions within the cerebropleural neuropil. Therefore, a decrease in K^+ conductances of type B photoreceptors could contribute both directly and indirectly to enhanced excitability by increasing the amplitude of CS-elicited generator potentials and increasing CS-elicited spike activity in the spike-generating zone by modification of conductances that influence the interspike interval.

Mechanisms of Pavlovian Conditioning

Recent modeling studies utilizing a Hodgkin-Huxley type analysis of membrane conductances and the SNNAP simulator (see NEUROSIMULATION: TOOLS AND RESOURCES) have shown that modulation of several K^+ currents (I_A , I_h , $I_{K,Ca}$) can account for both the enhanced excitability of type B photoreceptors and enhancement of monosynaptic IPSPs detected in conditioned animals (Cai, Baxter, and Crow, 2001; see also Fost and Clark, 1996). In addition,

modeling studies incorporating an analysis of membrane conductances in the phototransduction compartment, spike-generating zone, and synaptic terminals are providing insights into determining the relative contribution of changes in excitability and synaptic strength to modifications of complex PSP amplitude in postsynaptic neurons of the CS pathway.

Studies of the signal transduction pathways responsible for the modification of diverse K^+ currents of type B photoreceptors of conditioned animals have identified several second messenger systems. Both protein kinase C (PKC) and extracellular signal-regulated protein kinase (ERK) contribute to the conditioned modification of excitability and synaptic efficacy of *Hermisenda* (Alkon, 1989; Sahley and Crow, 1998; Muzzio et al., 2001).

Studies of one-trial conditioning have provided insights into the mechanisms of memory consolidation. One-trial conditioning (see above) produces short-, intermediate-, and long-term memory for enhanced excitability in identified type B photoreceptors. Associated with intermediate memory is the phosphorylation of a 24-kDa protein (CSP24) that exhibits a sequence identity to the β -thymosin family of actin-binding proteins (Crow and Xue-Bian, 2000). The regulation of CSP24 by one-trial conditioning occurs in neurons of the CS pathway and not in either the pedal or cerebropleural ganglia. Cytoskeletal-related proteins such as CSP24 may contribute to long-term structural remodeling in the CS pathway by regulating the turnover of actin filaments during the intermediate-term transition period between short- and long-term memory.

Discussion

The possibility of relating cellular changes to complex behavior in invertebrates is encouraged by the progress that has already been made in examining the neural mechanisms of simple forms of non-associative and associative learning. The results of these analyses have shown that (1) learning involves changes in existing neural circuitry (at least for the short-term, the growth of new synapses and the formation of new circuits for learning and memory are not necessary); (2) learning involves the activation of second messenger systems; (3) the second messenger affects multiple subcellular processes to alter the responsiveness of the neuron (at least one locus for the storage of memory is the alteration of specific membrane currents); (4) long-term memory requires new protein synthesis, whereas short-term memory does not; and (5) long-term memory may be associated with structural changes in the nervous system.

Road Map: Neural Plasticity

Related Reading: Conditioning; Habituation; Neuromodulation in Invertebrate Nervous Systems

References

- Alkon, D. L., 1989, Memory storage and neural systems, *Sci. American*, 261(1):42–50. ♦
- Baxter, D. A., and Byrne, J. H., 1993, Learning rules for neurobiology, in *The Neurobiology of Neural Networks* (D. Gardner, Ed.), Cambridge, MA: MIT Press, pp. 71–105.
- Baxter, D. A., Canavier, C. C., Clark, J. W., and Byrne, J. H., 1999, Computational model of the serotonergic modulation of sensory neurons in *Aplysia*, *J. Neurophysiol.*, 82:2914–2935.
- Byrne, J. H., 2002, Learning and memory: Basic mechanisms, in *Fundamental Neuroscience*, 2nd ed. (L. R. Squire, F. E. Bloom, J. L. Roberts, M. J. Zigmond, S. K. McConnell, and N. C. Spitzer, Eds.), San Diego: Academic Press. ♦
- Byrne, J. H., Baxter, D. A., Buonomano, D. V., and Raymond, J. L., 1990, Neuronal and network determinants of simple and higher-order features of associative learning: Experimental and modeling approaches, *Cold Spring Harbor Symposium on Quantitative Biol.*, 40:175–186.
- Cai, Y., Baxter, D. A., and Crow, T., 2001, A computational study of enhanced excitability in *Hermisenda* type B photoreceptor underlying one-trial conditioning: Role of conductances modulated by serotonin, *Soc. Neurosci. Abstr.*, 27:2532.
- Carew, T. J., 2000, *Behavioral Neurobiology*, Sunderland, MA: Sinauer Associates, chap. 10. ♦
- Crow, T., and Forrester, J., 1986, Light paired with serotonin mimics the effects of conditioning on phototactic behavior in *Hermisenda*, *Proc. Natl. Acad. Sci. USA*, 83:7975–7978.
- Crow, T., and Xue-Bian, J. J., 2000, Identification of a 24 kDa phosphoprotein associated with an intermediate stage of memory in *Hermisenda*, *J. Neurosci.*, 20:1–5.
- Fost, J. W., and Clark, G. A., 1996, Modeling *Hermisenda*: I. Differential contributions of I_A and I_C to type B cell plasticity, *J. Computat. Neurosci.*, 3:137–153.
- Fryszak, R. J., and Crow, T., 1994, Enhancement of type B- and type A-photoreceptor inhibitory connections in conditioned *Hermisenda*, *J. Neurosci.*, 14:1245–1250.
- Gingrich, K. J., and Byrne, J. H., 1987, Single-cell neuronal model for associative learning, *J. Neurophysiol.*, 57:1705–1715.
- Greenberg, S. M., Castellucci, V. F., Bayley, H., and Schwartz, J. H., 1987, A molecular mechanism for long-term sensitization in *Aplysia*, *Nature*, 329(6134):62–65.
- Kandel, E. R., 2001, Cellular mechanisms of learning and the biological basis of individuality, in *Principles of Neuroscience*, 4th ed. (E. R. Kandel, J. H. Schwartz, and T. M. Jessell, Eds.), New York: McGraw-Hill, pp. 1247–1279.
- Muzzio, I. A., Gandhi, C. C., Manyam, U., Pesnell, A., and Matzel, L. D., 2001, Receptor-stimulated phospholipase A(2) liberates arachidonic acid and regulates neuronal excitability through protein kinase C, *J. Neurophysiol.*, 85:1639–1647.
- Sahley, C., and Crow, T., 1998, Invertebrate learning: Current perspectives, in *Neurobiology of Learning and Memory* (J. Martinez and R. Kesner, Eds.), New York: Academic Press, pp. 171–209. ♦
- White, J. A., Ziv, I., Cleary, L. J., Baxter, D. A., and Byrne, J. H., 1993, The role of interneurons in controlling the tail-withdrawal reflex in *Aplysia*: A network model, *J. Neurophysiol.*, 70:1777–1786.

Ion Channels: Keys to Neuronal Specialization

José Bargas, Lucía Cervantes, Elvira Galarraga, and Andrés Fraguera

Introduction

Neurons code information and communicate by firing voltage spikes called action potentials (APs). Firing of APs is due to the presence of voltage-gated ion channels. Charge movement through them produces transient electrical currents that generate spikes. Ligand-gated ion channels activated during synaptic functioning produce patterns of voltage changes (see TEMPORAL DYNAMICS OF

BIOLOGICAL SYNAPSES) that bring membrane voltage to the activation range of different sets of voltage-gated ion channels. The latter promote or restrain the firing of APs following certain patterns (coding). Patterning is nonlinear and is not the simple summation of excitatory and inhibitory influences. This article summarizes why: the operation of voltage-gated channels.

Hodgkin and Huxley (1952) established a model to explain how ion channels generate APs (see AXONAL MODELING). APs link cel-

lular and systems neurophysiology: they are the feature extracted to design formal neurons (Arbib, 1964) (see CANONICAL NEURAL MODELS); they encode the physical properties of stimuli, motor commands, and working memory, and they correlate with behavior and perception (see section on MAMMALIAN BRAIN REGIONS and COGNITIVE DEVELOPMENT). However, diverse combinations of different voltage-gated ion channels are possible. Each neuron is endowed with a different combination (Llinás, 1988; Huguenard and McCormick, 1994). Therefore, each cell responds with a distinct set of firing patterns on synaptic activation: plateaus with repetitive firing, bistability, frequency adaptation, tonic firing, bursting, spontaneous pacemaking, etc. There is a “neuronal specialization” that reflects different functions of neural nets (locationism) supported by variations in firing (Llinás, 1988), morphology (Cajal, 1899), and the distribution of afferents with transmitters and modulators (Nicoll, 1988) (see BIOPHYSICAL MOSAIC OF THE NEURON) (Fig. 1).

More than 50 genes encode the pore-forming domains (α -subunits) of K^+ channels (KCN). More than 10 genes encode Na^+ channels (SCN), and at least 10 genes encode pores for Ca^{2+} channels (CACN). Five genes encode cation or pacemaking channels (HCN). And α -subunits are not alone. Auxiliary subunits, β , γ , $\alpha_2\delta$, and δ , encoded by other genes, change the kinetics and voltage dependence of channels. Thousands of channel types can theoretically arise from subunit combination and alternative splicing (proteomics from genomics). How can we make functional sense of this complexity? One way is to take *firing* as the crucial property. Since neurons are specialized, we should extract, as simply as possible, their different firing properties (e.g., Suri, Vargas, and Arbib, 2001). We need to know which are the main ion channels that contribute to different firing patterns.

The roles for ion channels are (1) to generate APs and (2) to set a particular pattern, rhythm, oscillation, threshold, adaptation, pa-

cemaking, bursting, etc. for the firing of APs. This is the coding process that produces a distinct input-output function (I/O function) for a neuron in a certain condition (see SINGLE-CELL MODELS).

Operation of Ion Channels

Figure 2 simplifies the operations of an ion channel, and Figure 3 illustrates basic firing patterns. Ion currents through channels follow Ohm's law,

$$I_i = (G_{MAX} \cdot m^n h) \cdot (V - E_i) \quad (1)$$

where I_i = current, $G_{MAX}m^n h$ = conductance ($g(V)$), and $V - E_i$ = voltage (ΔV). Channels can be closed (C), open (O), or inactive (I) (Figure 2). Only the O -state produces an I_i . To be in any state, C , O , or I , depends on voltage and time: $I(V, t)$. A voltage change (ΔV) modifies the probability of being in any state. But velocity of change, or *time dependence*, is a main difference between ion channels, which is assessed by extracting time constants (τ_s) from current (I_i) records represented by (for example):

$$I_i(t) = r_1(1 - \exp(-t/\tau_m))^n \exp(-t/\tau_h) + r_2 \quad (2)$$

where τ_m is an activation time constant, τ_h is an inactivation time constant, r_1 and r_2 are constants, and n is the exponent that fits nonlinear kinetics (Figure 2). τ_s (h or m above) depend on voltage. Fitting I_i records for different voltages (I - V plots) with Equation 2 produces a family of τ_s (Figure 2) where $\tau \approx 1/(\alpha + \beta)$ and α is the forward rate constant, whereas β is the backward rate constant for changing state (Figure 2). Differences in *time dependence* dictated by τ_s make currents *transient*, *fast*, *slow*, *persistent*, or *slowly inactivating*.

Conductance, $g(V)$, measurements are also obtained from I - V plots, where Equation 1 is applied to each record to obtain $g(V)$,

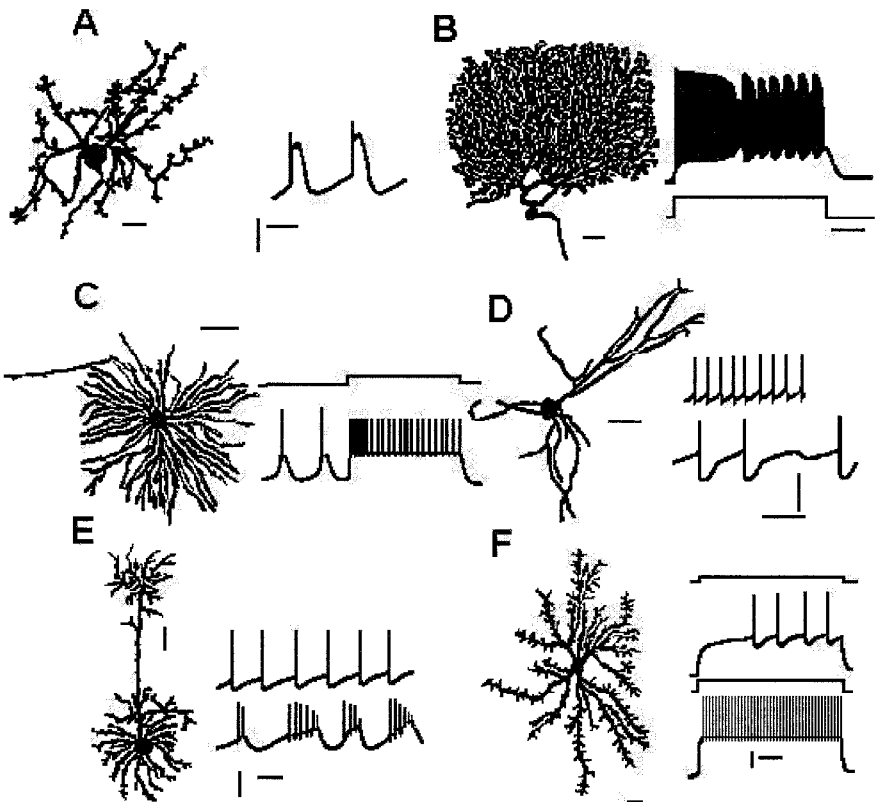


Figure 1. Neurons differ in morphology and firing pattern (see NEOCORTEX-BASIC NEURON TYPES). A, Inferior olive neuron. B, Cerebellar Purkinje cell. C, Thalamic relay neuron. D, Substantia nigra compacta neuron. E, Cortical pyramidal neuron. F, Spiny neuron of the neostriatum.

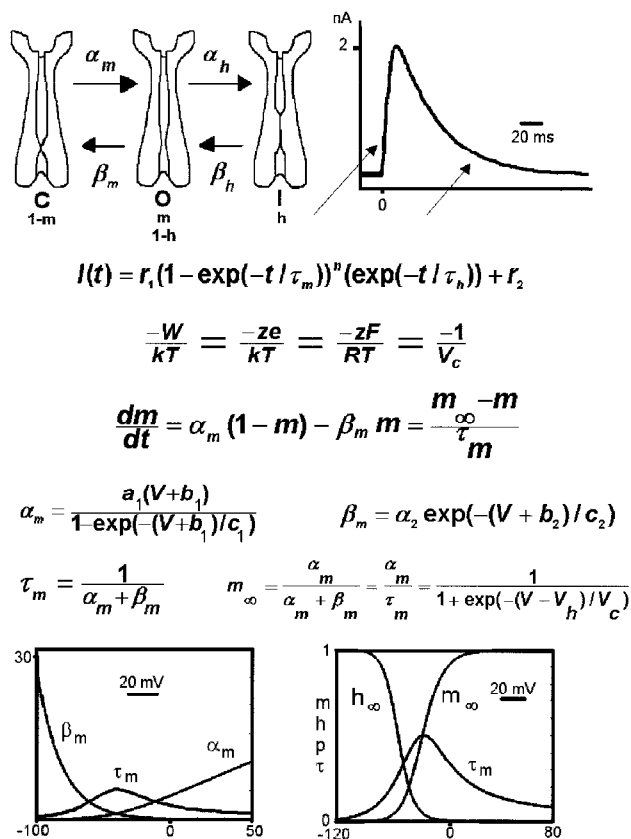


Figure 2. Channel function. At the top, a closed channel (C) may open (O) following rate constant α_m (below). The channel may close again following rate constant β_m or may inactivate (I) following rate constant α_h . De-inactivation follows rate constant β_h . $\tau_{m,h} \approx 1/\alpha + \beta$. Activation (m_∞) and inactivation (h_∞) functions can be obtained by plotting the conductance obtained from I-V plots. The bottom graph depicts relations between m_∞ , τ , α , and β . Once α and β are obtained, a differential equation (dm/dt) describes the kinetics of a channel. Channels differ in voltage and time dependence.

which, plotted against V , forms a sigmoid function (Figure 2) called the *activation* (or *inactivation*) function (m_∞ and h_∞). $g(V)$ dictates the *voltage dependence* of the current and is the other main difference between ion channels, which are then *threshold*, *subthreshold*, or *suprathreshold* in respect to firing. $g(V)$ can be fitted to:

$$g(V) = G_{\text{MAX}} / (1 + \exp(\pm(V - V_h)/V_c)) \quad (3)$$

(activation: $-$; inactivation $+$ sign). Where $G_{\text{MAX}} = g$ when all channels are open, V denotes voltage, V_h is the voltage at which half the channels are open, and V_c is the slope factor of the sigmoid equivalent to a Boltzmann exponent (RT/F or kT/e -in mV; ≈ 26 mV at 25°C), z denotes valence, $V - V_h$ is ΔV , and F , R , T , k , T , and e have their usual meaning (Figure 2). $g(V)$ is a cumulative Boltzmann distribution (Figure 2) but, normalizing $G_{\text{MAX}} = 1.0$, becomes a probability (P) function where m is the P to open and $1 - m$ is the P to close (Figure 2). *Inactivation* is denoted by h (Figure 2 uses p for both).

α and β define first-order kinetics: simple exponential solutions multiplied by constants (Figure 2) (Jack, Noble, and Tsien, 1975) or Boltzmann distributions. $g(V)$ is proportional to permeability

($P(I, V)$), which can be obtained transforming I - V plots with the Goldman-Hodgkin-Katz equation for current,

$$P(I, V) = \frac{I}{V} \cdot \frac{RT}{(zF)^2} \cdot \left(\frac{\exp(-zFV/RT) - 1}{[C]_o \cdot \exp(-zFV/RT) - [C]_i} \right) \quad (4)$$

where $[C]_i$ denotes internal ion concentrations, $[C]_o$ denotes external ion concentrations, I is current, and V is voltage. $g(V)$ can also be obtained by differentiating the I - V plot or by using instantaneous I - V plots (tail currents).

In summary, V_h , V_c , and τ_s characterize ion conductances, i.e., $g(V)/\tau_m = \alpha_m$, etc., giving their time and voltage dependence. Neurons use an array of ion conductances differing in time, voltage dependence, and the ion carried. The concerted work of them makes up the firing properties or coding.

The activation function m (Equations 1 and 3) is raised to an n power, where n is the order of the kinetic reaction. Thus, nonlinearity arises by the parallel action of n first-order processes. Each represents the movement of a protein (channel) subunit or domain. All of them move for the whole channel molecule to change state. The sigmoidal delay to change may be viewed as the many "closed states" (C_1, C_2, \dots, C_n) that have to be crossed before entering the "open state." Note that, since each channel is composed of several domains, nonlinearity arises at the molecular level. Imagine then what happens when many of such molecules combine to produce a firing pattern, when neurons possessing different arrays of these channels combine to form a neural net, and when nuclei possessing different nets arrange to produce a nervous system.

Inward or outward ion currents depend on the value of E_i or equilibrium potential of the ion (Equation 1). If E_i is less than firing threshold, I is a hyperpolarizing outward current that will tend to arrest firing (Figure 3). If E_i is greater than firing threshold, I is a depolarizing inward current that will promote firing. When a neuron increases firing, there is no a priori way to know whether an inward current was facilitated or an outward current was restrained.

Neuronal Specialization

Each neuronal class has a different set of ion conductances. The original HH model (see AXONAL MODELING), consisting of a transient inward current (Figures 3B1, 3C1), a persistent outward current (Figures 3B3, 3C2), and a leak current (Figure 3A2) (I_{Na} , I_K , and I_{leak}), gives a *basic firing mechanism* (BFM). Leak currents are made up of a family of K^+ channels with a double pore-forming domain in tandem (KCNK1-7, 9, 10, 12, 13, 16, 17) with 13 members. These channels make up the electrotonic structure or cable properties.

A good exercise using a simulator (e.g., Huguenard and McCormick, 1994) is to begin with the BFM (Figure 3A3) and then add one class of conductance at a time to see how the BFM is modified (column A in Figure 3). From top to bottom, the records in column A of Figure 3 show the BFM (Figure 3A3) with one auxiliary current at a time (Figures 3A4-9). Any firing pattern is due to a sum of conductances,

$$C \frac{Vd}{dt} = - \sum_i^n (I_i) = -(I_{Na} + I_K + I_{leak} + I_{to} + I_{so} + I_{ti} + I_{si} + \dots + I_n) \quad (5)$$

where to denotes transient outward, so denotes slowly inactivating outward, ti denotes transient inward, si denotes slowly inactivating inward, and so on; C denotes capacitance. Column D shows a modeling experiment adding several auxiliary conductances in sequence.

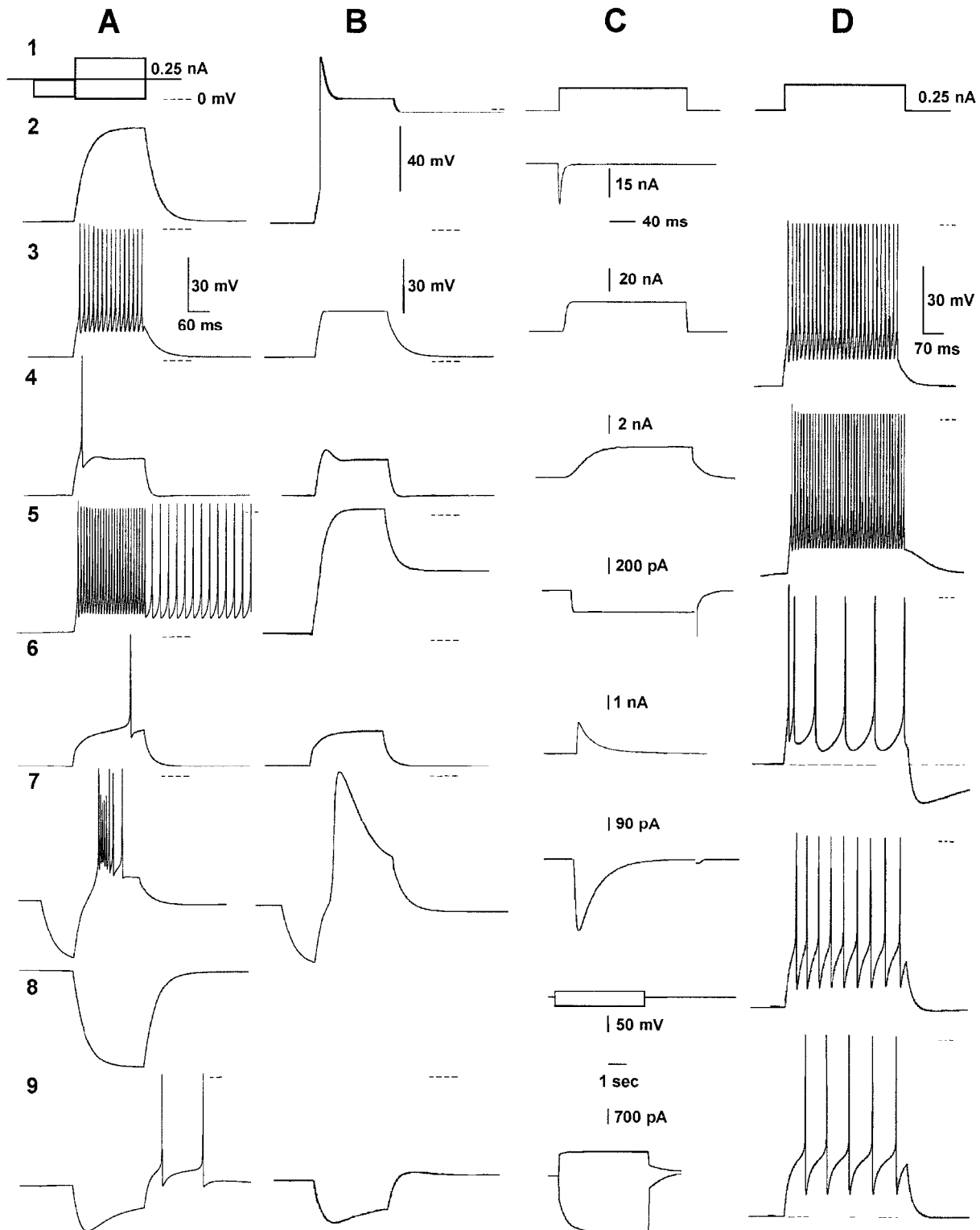


Figure 3. Firing patterns. Columns A, B, and D are voltage recordings; column C shows current recordings. A1, Stimuli. The main stimulus is a depolarizing current step. A2, I_{leak} produces RC responses. A3, I_{Na} and I_{K} are added to produce a basic firing mechanism (BFM). A4, BFM plus I_{SO} produces adaption. A5, BFM plus I_{SI} produces a plateau potential and increases firing. A6, BFM plus I_{TO} delays firing. A7, BFM plus I_{TI} produces bursting after a hyperpolarization. A8, Hyperpolarizing RC response (cf. A2). A9, BFM plus I_{h} produces rebound excitation. B, RC response plus-

I_{Na} (B2), I_{K} (B3), I_{SO} (B4), I_{SI} (B5), I_{TO} (B6), I_{TI} (B7), and I_{h} (B9). C, A depolarizing command (C1) evokes I_{Na} (C2), I_{K} (C3), I_{SO} (C4), I_{SI} (C5), I_{TO} (C6), I_{TI} (C7), and hyperpolarizing and depolarizing commands (C8) elicit I_{h} (C9). D, BFM after a depolarizing step (D3), plus subthreshold I_{Ca} (firing increases: D4–5), I_{SK} (AHP and adaptation increase while frequency decreases: D6), I_{BK} (frequency increases and there is less adaptation: D7–8), and I_{P} (frequency decreases and becomes regular; AHPs look smaller than D9) added in sequence.

Functional Classification

- Activated by depolarization
 - Inward
 - Transient (I_T)
 - Slowly or incompletely inactivating (I_{si})
 - Outward
 - Transient (I_{to})
 - Slowly or incompletely inactivating (I_{so})
- Activated by hyperpolarization
 - Inward
 - Cationic currents (pacemaking) (I_h)
 - Outward
 - Inward rectifiers (K_{ir})

Transient inward (I_T) currents are sodium currents (I_{Na} in Figure 3C2; Goldin, 2001) that inactivate quickly and produce the depolarizing phase of the spike (Figures 3A3, 3B2). One class, Nav1.1 to 1.9 or SCN1–11, is blocked by tetrodotoxin and saxitoxin, although some of them need high concentrations (SCN10, 11). A second class, Nav2.1 (SCN6, 7: SNS and NaN), is tetrodotoxin resistant. Certain types allow a small percentage of the current (<10%) without inactivation, generating a persistent current (e.g., Nav 1.6).

Another I_h are calcium “T-currents” ($\alpha_{1G,H,h}$, I_T , or CACNA, blocked by kurtuxin) (Randal and Benham, 1999), which produce “low-threshold spikes” (sinoatrial node) (Figure 3B7) that trigger sodium spikes (in neurons), producing bursts (Figure 3A7). A previous hyperpolarization—an afterhyperpolarization (AHP) or an inhibitory synaptic potential (IPSP)—de-inactivates T-channels (Figure 3A7). Thalamic, nigral, pallidal, pontine, cortical, and many other neurons use I_T to fire in bursts after an IPSP, producing rebound excitation, postinhibitory afterdischarges, augmentation responses, and slow rhythmic bursting, like that seen in spindle waves (see OSCILLATORY AND BURSTING PROPERTIES OF NEURONS). At depolarized potentials, T-channels are inactive.

Thus, neurons are able to respond differently depending on previous membrane potential. This allows a net to behave differently at different moments, while using the same neuronal elements (multitask networks). Since APs can be evoked from two different membrane potentials, the neuron has two functionally different firing thresholds. A neuron may have more than one threshold, and different firing patterns may be evoked from each threshold.

Transient outward (I_{to}) currents are potassium currents or I_A (e.g., Kv1 and Kv4 families, or KCN1,4, blocked by dendrotoxin and aminopyridines) (Figure 3C6). They oppose I_T . Inactivating between spikes, they set a stereotyped behavior between APs pacing tonic firing (Figure 3D9) at low frequencies. When subthreshold, they oppose any depolarization (Figure 3B6), so that membrane potential reaches threshold with a delay: I_A “retards” membrane trajectory toward threshold (Figure 3A6). A “conditioning” stimulus inactivates I_A , increasing responses with time, so that a previous subthreshold stimulus may reach threshold after a delay (*facilitation*).

A number of genes, differing in τ_h , code for I_A (Shieh et al., 2000). Hence, firing latency and rhythmic firing depend on voltage and time, expanding the I/O function (see column D of Figure 3).

Slowly inactivating inward (I_{si}) currents inactivate after several hundreds of milliseconds or seconds. On a short time scale they may be viewed as persistent. They are calcium currents, or persistent components of some sodium currents, or cationic currents. Their activation produces a depolarization that adds to the stimulus, enhancing it (Figure 3B5; cf. Figure 3A2) and increasing firing (Figure 3A5). Calcium channels come in two families, L ($\alpha_{1S,C,D,F}$ or CACNA-S,C,D,F) and non-L ($\alpha_{1A,B,E}$ or CACNA-A,B,E). L channels are blocked by calciseptine and dihydropyridines. Their inactivation depends on intracellular Ca^{2+} and voltage: Ca^{2+} en-

ters and shuts down the channel (feedback). Inactivation of the non-L family depends mainly on voltage (except for P).

I_{si} produce “plateau potentials” that sustain repetitive firing (Figure 3D5; cf. Figure 3D3) and bistable properties, i.e., two stable membrane potentials that alternate. There is a gain in the I/O function during plateaus. In the dendrites, slow synaptic depolarizations (NMDA) boost plateaus produced by I_{si} .

Activated during the spike, I_{si} contribute to it and to activating the potassium conductances that generate the AHP (Figure 3D6). Plateau potentials, bistability, dendritic spikes, and activation of outward currents are only some roles of I_{si} . Calcium entry has many other roles: in transmitter release, muscle contraction, cytoskeleton function, enzyme and gene activation, and so on.

Slowly inactivating outward (I_{so}) potassium currents (Figure 3C4) consist of (1) Kv channels as Kv2,3 or KCNA-D, 1–4; (2) KvLQT or KCNQ1–5, HERG or KCNH1–4, which are blocked by TEA, noxiustoxin, etc. (Shieh et al., 2000); (3) SK and IK channels or KCNN1–3 and KCNN4, respectively, which are blocked by apamin and charybdotoxin and activated by intracellular calcium; and (4) slow channels, BK or KCNMA1, which are blocked by TEA, iberiotoxin, and charybdotoxin (Shieh et al., 2000) and activated by voltage and intracellular calcium. All tend to maintain a quiet membrane-opposing depolarizing stimulus (Figure 3B4), spikes, inward currents, and plateau potentials. They decrease excitability, decrease or arrest firing frequency, and augment firing threshold. Depolarization activates I_{so} , it hyperpolarizes the membrane and then shuts down again (feedback). The BFM (Figure 3A3) becomes adapting firing (Figure 3A4) if a slow outward current (Figure 3C4) is superimposed. Conversely, when I_{so} is blocked, a frequency gain occurs. Note that several I_{so} depend on calcium accumulation. Ca^{2+} enters with each spike, allowing a short-time memory. Calcium-dependent gating of AHP opens a fraction of channels, depending on the number of spikes fired (digital to analog conversion) (Figure 3D6). The AHP fixes the time interval between spikes or trains of spikes and ends episodes of increased excitability, sets the pace for rhythmic firing and bursting, and allows frequency control and adaptation (see THALAMUS). Firing depends on calcium dynamics (see column D in Figure 3) because all neurons possess some variety of a calcium-dependent I_{so} . Thus, firing mechanisms have to simulate calcium dynamics to be realistic.

Inward currents activated by hyperpolarization (I_h) or HCN1–4 blocked by cesium (Kaupp and Seifert, 2001) activate when the membrane potential hyperpolarizes below firing threshold, producing voltage “sags” ($E_i \approx -35$ mV) that oppose the same hyperpolarization (Figure 3B9). They contribute to rebound “humps” that may attain firing (Figure 3A9) when the hyperpolarization is over. I_h , I_T , and I_A , acting in concert, produce rhythmic bursting (see SLEEP OSCILLATIONS). I_h are “pacemaking currents” that allow spontaneous firing: an AP produces an AHP; the AHP activates I_h , which then depolarizes the membrane back to fire another AP, which is followed by another AHP that repeats the cycle, keeping the cell firing. When I_h reaches the threshold for T-channels during rebound, a low-threshold Ca spike and bursting may ensue. Since the cycle (e.g., the orbit in the phase plane) is initiated by a hyperpolarization, the cell has a “threshold” going in the hyperpolarizing direction (an unstable singular point that initiates the entire orbit).

Outward currents activated by hyperpolarization (K_{ir} s) are inward rectifier potassium channels composed of four domains, each with only two transmembrane segments. Strong activation by a hyperpolarization produces inward potassium currents (K_{ir} or KCNJ1–15, blocked by cesium and barium) (Shieh et al., 2000), but no physiological stimulus hyperpolarizes the cell beyond E_K . Thus, the role of these channels is to *close* when the cell is depolarized. At rest, some channels are open and participate in the resting membrane potential and cable properties. When synaptic entries depolarize the cell, K_{ir} current shuts down and all inputs increase their value abruptly, since electrotonic length has shrink.

The threshold is dynamic. It depends on K_{irs} , which act as a gate, requiring a convergence of inputs to pass the signal. A sum of synaptic inputs is not responsible for firing, but a complex interaction of intrinsic and synaptic conductances change the I/O function. Activation of K_{irs} depends on the potential and on extracellular potassium, which depend on the excitability level (synchronicity).

Discussion

Brain neuronal circuits are complex and dynamic. Ion channels endow neurons with properties such as multiple thresholds, different firing patterns associated with each threshold, the possibility of switching between different firing patterns, timing, multiple time constants for spike frequency adaptation, a changing electrotonic length, pacemaking, intrinsic facilitation, gating, bistable properties, short memory traces, etc. Each ion current contributes to a set of properties. Many ion conductances modify the BFM. Different firing patterns are shaped by previous activity and modulation. Accordingly, neuronal nets dynamically switch between different firing states and different configurations of synaptic weights: n thresholds and n firing levels due to n ion conductances activated at each level. A different pattern at each level will reach synaptic terminals differently. Each pattern encompasses a different I/O function. An I/O function may favor learning, while others may favor consolidation. Some would be preferred by sensory neurons, others by intermediate or motor neurons. Some net-states induce resonance between a set of nuclei. Other net-states decouple these nuclei but may couple other nuclei.

The simple picture used here to hint into this complexity is an abstraction. Nonlinearities produce many counterintuitive outcomes. One example is shown in column *D* of Figure 3, in which several conductances were added in sequence (from top to bottom) to the BFM (in Figures 3D3–4): a subthreshold Ca current (Figures 3D4–5) enhanced evoked discharge, a Ca-activated SK-type current (Figure 3D6) decreased firing frequency and increased AHPs and adaptation, but addition of a BK type of current did not decrease firing (Figure 3D8) but, surprisingly, increased firing and produced less adaptation. The addition of I_A (with a rather slow τ_h) produced

tonic firing with very low firing frequencies never reached by BFM alone (e.g., as in a neostriatal spiny neuron). Comparing the AHPs of records in Figure 3D6 and Figure 3D9, one would not imagine that the firing in Figure 3D9 involved more outward current.

Road Map: Biological Neurons and Synapses

Related Reading: Activity-Dependent Regulation of Neuronal Conductances; Biophysical Mechanisms in Neuronal Modeling; Biophysical Mosaic of the Neuron; Neocortex: Chemical and Electrical Synapses

References

- Arbib, M. A., 1964, *Brains, Machines, and Mathematics*, New York: McGraw-Hill.
- Cajal, S. R., 1899, *Textura del Sistema Nervioso del Hombre y de Los Vertebrados*, Madrid: Universidad de Alicante.
- Goldin, A. L., 2001, Resurgence of sodium channel research, *Annu. Rev. Physiol.*, 63:871–894.
- Hodgkin, A. L., and Huxley, A. F., 1952, A quantitative description of membrane current and its application to conduction and excitation in nerve, *J. Physiol. (Lond.)*, 117:500–544.
- Huguenard, J., and McCormick, D. A., 1994, *Electrophysiology of the Neuron*, New York: Oxford University Press. ♦
- Isomoto, S., Kondo, C., and Kurachi, Y., 1997, Inwardly rectifying potassium channels: The molecular heterogeneity and function, *Jpn. J. Physiol.*, 47:11–39.
- Jack, J. J. B., Noble, D., and Tsien, R. W., 1975, *Electric Current flow in Excitable Cells*, Oxford: Clarendon Press. ♦
- Kaupp, U. B., and Seifert, R., 2001, Molecular diversity of pacemaker ion channels, *Annu. Rev. Physiol.*, 63:235–257.
- Llinás, R., 1988, The intrinsic electrophysiological properties of mammalian neurons: Insights into central nervous system function, *Science*, 242:1654–1664.
- Nicoll, R. A., 1988, The coupling of neurotransmitter receptors to ion channels in the brain, *Science*, 241:545–551.
- Randall, A., and Benham, C. D., 1999, Recent advances in the molecular understanding of voltage-gated Ca^{2+} channels, *Mol. Cell. Neurosci.*, 14:255–272.
- Shieh, C.-C., Coghlan, M., Sullivan, J. P., and Gopalakrishnan, M., 2000, Potassium channels: Molecular defects, diseases, and therapeutic opportunities, *Pharmacol. Rev.*, 52:557–593.
- Suri, R. E., Bargas, J., and Arbib, M. A., 2001, Modeling functions of striatal dopamine modulation in learning and planning, *Neuroscience*, 103:65–85.

Kalman Filtering: Neural Implications

Simon Haykin

Introduction

The time-domain description of a system by a *state-space model*, depicted in Figure 1, is of profound importance. The notion of state plays a key role in the formulation of this model. The *state*, denoted by the vector $\mathbf{x}(n)$, is defined as any set of quantities that would be sufficient to uniquely describe the unforced dynamic behavior of the system at discrete time n . The model of Figure 1 is not only mathematically convenient, it also offers a close relationship to physical/neurobiological reality and a basis for accounting for the statistical behavior of the system.

The state-space model of Figure 1 embodies two basic equations:

1. Process equation

$$\mathbf{x}(n+1) = \mathbf{F}(n+1, n)\mathbf{x}(n) + \mathbf{v}_1(n) \quad (1)$$

where $\mathbf{F}(n+1, n)$ is a transition matrix and the vector $\mathbf{v}_1(n)$ is an additive dynamic noise.

2. Measurement equation

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n) \quad (2)$$

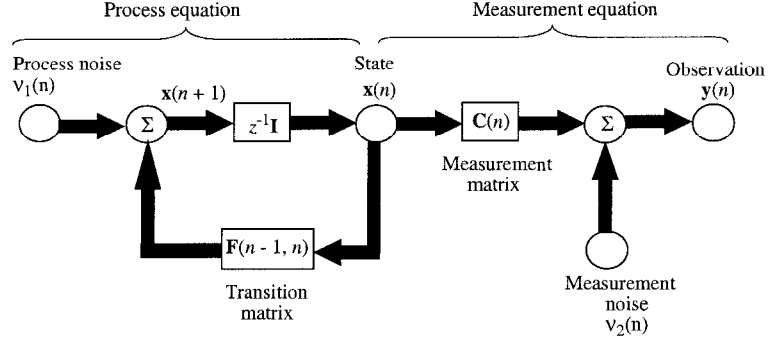
where the vector $\mathbf{y}(n)$ is the *observation*, $\mathbf{C}(n)$ is a *measurement matrix*, and the vector $\mathbf{v}_2(n)$ is an additive *measurement noise*.

Typically, the state $\mathbf{x}(i)$ is hidden and therefore unknown, and the requirement is to estimate it using a sequence of observations $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n)$. The sequential estimation problem is called *filtering* if $i = n$, *prediction* if $i > n$, and *smoothing* if $1 \leq i < n$. Unlike smoothing, both filtering and prediction are real-time operations. In this article, we only consider prediction and filtering, which are closely related.

Kalman Filters

In a classic paper, Kalman (1960) derived a general solution for the *linear* filtering problem, and with it the celebrated *Kalman filter*

Figure 1. Signal-flow graph representation of a linear, discrete-time dynamical system.



was born. Assuming that the dynamic noise $v_1(n)$ and measurement noise $v_2(n)$ are independent, white, and Gaussian processes, the Kalman filter is a recursive estimator that is optimum in the minimum mean-square error or, equivalently, maximum likelihood sense (Jazwinski, 1970).

Let \mathbf{Y}_{n-1} denote the subspace spanned by the observations $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n-1)$. Given the new observation $\mathbf{y}(n)$, the current estimate of the state denoted by $\hat{\mathbf{x}}(n|\mathbf{Y}_{n-1})$ is recursively updated as follows:

$$\hat{\mathbf{x}}(n+1|\mathbf{Y}_n) = \mathbf{F}(n+1, n)\hat{\mathbf{x}}(n|\mathbf{Y}_{n-1}) + \mathbf{G}(n)\alpha(n) \quad (3)$$

where $\mathbf{G}(n)$ is the *gain matrix*, and the vector

$$\alpha(n) = \mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathbf{Y}_{n-1}) \quad (4)$$

is the *innovation*, representing the part of $\mathbf{y}(n)$ that is new. Equations 3 and 4 show that the underlying structure of the Kalman filter is in the closed-loop form of a *predictor-corrector*, consisting of two steps:

1. *Measurement update*, which uses the current observation $\mathbf{y}(n)$ to compute the innovation $\alpha(n)$.
2. *Time update*, which uses $\alpha(n)$ to update the past estimate $\hat{\mathbf{x}}(n|\mathbf{Y}_{n-1})$.

In addition to Equations 3 and 4, the Kalman filter involves three other basic steps (Haykin, 2002):

1. *Computation* of the gain matrix $\mathbf{G}(n)$ in terms of an error covariance matrix $\mathbf{K}(n, n-1)$, where the error refers to the difference between the true state $\mathbf{x}(n)$ and the current estimate $\hat{\mathbf{x}}(n|\mathbf{Y}_{n-1})$.
2. *Time updating* of the error covariance matrix $\mathbf{K}(n, n-1)$ via the so-called *Riccati equation*.
3. *Initialization* of the filter by setting $\hat{\mathbf{x}}(1|\mathbf{Y}_0) = \mathbf{0}$ and $\mathbf{K}(1, 0) = \mathbf{\Pi}_0$, where $\mathbf{0}$ is the null vector and $\mathbf{\Pi}_0$ is a prescribed diagonal matrix.

The computational complexity of the Kalman filter is of order M^2 , where M is the dimensionality of the state space.

A serious limitation of the standard Kalman filter is that it is prone to unstable behavior that may arise due to model mismatch and use of finite-precision arithmetic. The origin of the problem is traced to the fact that in situations of this kind, the Riccati equation may *not* result in a non-negative definite solution for the error covariance matrix $\mathbf{K}(n, n-1)$, which is unacceptable. (An important property of a covariance matrix is that it must be non-negative definite.) The unstable behavior of the Kalman filter is referred to as the *divergence phenomenon*.

To overcome this phenomenon, we may use *square-root filtering*, whereby the square root of the error covariance matrix rather

than the error covariance matrix itself is propagated through the filter. According to the *Cholesky factorization*, we may write

$$\mathbf{K}(n, n-1) = \mathbf{K}^{1/2}(n, n-1)\mathbf{K}^{T/2}(n, n-1) \quad (5)$$

where $\mathbf{K}^{1/2}(n, n-1)$ is a lower triangular matrix (i.e., all the elements above the main diagonal are zero), and $\mathbf{K}^{T/2}(n, n-1)$ is its transpose. The important thing to note here is that the product of a lower triangular matrix and its transpose is unlikely to become indefinite.

The discussion up to this point rests on the premise that the state-space model of Equations 1 and 2 is linear. What if the model is nonlinear, as shown by the equations

$$\mathbf{x}(n+1) = \mathbf{f}(n, \mathbf{x}(n)) + \mathbf{v}_1(n) \quad (6)$$

$$\mathbf{y}(n) = \mathbf{c}(n, \mathbf{x}(n)) + \mathbf{v}_2(n) \quad (7)$$

where \mathbf{f} and \mathbf{c} are time-varying, vector-valued functions? The explicit dependence on time n accounts for a possibility that the equations are time varying. To deal with this new situation, we may extend the use of the standard Kalman filter. Specifically, the nonlinear process equation (Equation 6) and the nonlinear measurement equation (Equation 7) are linearized at each iteration of the filter around most recent estimates of the state, which is achieved by retaining the first-order terms in the Taylor series expansions of the nonlinear functions \mathbf{f} and \mathbf{c} . For obvious reasons, the resulting filter is referred to as the *extended Kalman filter*, or EKF (Jazwinski, 1970; Haykin, 2002).

Supervised Training of Neural Networks

The Kalman filter offers two important properties: (1) estimation of the state using the entire past sequence of observations, and (2) use of second-order information in the form of an error covariance matrix. These two properties make the Kalman filter into a powerful tool for the supervised training of neural networks. The issue of concern here is how to proceed with this approach in a computationally feasible manner without compromising the application of Kalman filter theory. The answer lies in using a *decoupled* form of the extended Kalman filter, in which the computational complexity is made to suit the requirements of a particular application and available computational resources (Puskorius and Feldkamp, 2001).

In this article, we consider the supervised training of a recurrent multilayer perceptron (RMLP), for which the decoupled extended Kalman filter (DEKF) has established itself as an enabling technology by solving some difficult signal processing and control problems.

Let the vector $\mathbf{w}(n)$ denote the synaptic weights of the entire RMLP at iteration n . With adaptive filtering in mind and $\mathbf{w}(n)$ viewed as a state of the RMLP, we may formulate the state-space model of the network as (Haykin, 1999)

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{v}_1(n) \quad (8)$$

$$\mathbf{d}_0(n) = \mathbf{c}(\mathbf{w}(n), \mathbf{u}(n), \mathbf{v}(n)) + \mathbf{v}_2(n) \quad (9)$$

Equation 8 describes a diffusion process. The vector-valued function \mathbf{c} accounts for the overall nonlinearity from the input layer to the output layer of the RMLP. The arguments $\mathbf{u}(n)$ and $\mathbf{v}(n)$ of the function \mathbf{c} denote the signal vector applied to the input layer and the vector of recurrent activation potentials at internal nodes of the RMLP, respectively. The vector $\mathbf{v}_1(n)$ denotes noise *artificially* introduced into the process equation, and the vector $\mathbf{v}_2(n)$ denotes additive noise in the measured data. The vector $\mathbf{d}_0(n)$ is the desired response of the RMLP.

There are two contexts in which the term “state” is used here:

- The network’s weights, $\mathbf{w}(n)$, which are adjusted during training.
- The recurrent activation functions, $\mathbf{v}(n)$, which continue to evolve nonlinearly with time once the training ends.

By comparing the model of Equations 8 and 9 for the RMLP with the linear dynamical model of Equations 1 and 2, we see that the difference between these two models is in the nonlinear form of the measurement equation (Equation 9). This matter is taken care of through linearization, thereby facilitating application of the EKF. This linearization requires the partial derivatives of the output(s) of the RMLP with respect to its weights. (Backpropagation through time provides an efficient algorithm for computing these partial derivatives.) Decoupling is introduced into the extended Kalman filtering algorithm by assuming that the interactions between the estimates of certain weights in the RMLP can be ignored, the effect of which is to introduce zeros into the error covariance matrix. If the weights are decoupled so that certain subgroups of weights are mutually exclusive of one another, then the error covariance matrix can be arranged into a block-diagonal form, thereby reducing the computational burden of the algorithm (Puskorius and Feldkamp, 2001).

Two noteworthy points on the use of Kalman filtering for the supervised training of recurrent networks are:

1. Introduction of the artificial noise $\mathbf{v}_1(n)$ in the process equation (Equation 8) has the desirable effects of accelerating the convergence process and enhancing the likelihood of reaching a global minimum of the error performance surface. Through the use of an annealing procedure, the effect of $\mathbf{v}_1(n)$ can be gradually reduced as the network approaches an equilibrium condition.
2. The presence of second-order information in the form of error covariance matrix has the desirable effect of overcoming the vanishing gradients problem that arises when a recurrent network is trained with a gradient-based algorithm such as the real-time recurrent learning algorithm (Haykin, 1999).

Dynamic Model of Visual Recognition

The visual cortex is endowed with two key anatomical properties:

- *Abundant use of feedback.* The connections between any two connected areas of the visual cortex are bilateral, thereby accommodating the transmission of forward as well as feedback signals between the interconnected cortical areas.
- *Hierarchical multiscale structure.* The receptive fields of lower-area cells in the visual cortex span only a small fraction of the visual field, whereas the receptive fields of higher-area cells increase in size until they span almost the entire visual field. It is this constrained network structure that makes it possible for the fully connected visual cortex to perform prediction in a high-

dimensional data space with a reduced number of free parameters and therefore in a computationally efficient manner.

Rao and Ballard (1997) exploit these two properties of the visual cortex to build a dynamic model of visual recognition, recognizing that vision is fundamentally a nonlinear dynamic process. The motivation for building the model was to explain the way in which the responses of cells in the visual cortex are significantly modulated by stimuli from beyond the classical receptive field. This modulation can be exerted from multiple sources, including the higher-level systems that are activated when an animal views a natural scene.

The Rao-Ballard model of visual recognition is a hierarchically organized neural network, with each intermediate level of the hierarchy receiving two kinds of information: bottom-up information from the preceding level, and top-down information from the higher level. For its implementation, the model uses a multiscale estimation algorithm that may be viewed as a hierarchical form of the extended Kalman filter. In particular, the EKF is used to simultaneously learn the feedforward, feedback, and prediction parameters of the model using visual experiences in a dynamic environment. The resulting adaptive processes operate on two different time scales:

- A *fast* dynamic state-estimation process, which allows the dynamic model to anticipate incoming stimuli.
- A *slow* Hebbian learning process, which provides for synaptic weight adjustments in the model.

In a subsequent study, Patel, Becker, and Racine (2001) studied the use of an RMLP trained with the DEKF algorithm to deal with high-dimensional signals, namely, moving visual images. The particular problem dealt with is the tracking of objects that vary in both shape and location, which is a challenging problem. By making use of short-term continuity, Patel et al. show that their model is capable of tracking a mixture of different geometric shapes (circles, squares, and triangles). As with the Rao-Ballard dynamic model of visual recognition, the Patel-Becker-Racine model is designed with a hierarchical structure; specifically, the first hidden layer of neurons in the RMLP was connected to relatively small, local regions of the visual field applied to the input layer, and a subsequent hidden layer spanned the entire visual field. The Patel-Becker-Racine model may be viewed as a first step toward modeling the dynamic mechanism by which the human brain might be simultaneously recognizing and tracking moving stimuli.

Hypothesis That the Cerebellum Is a Neural Analog of a Dynamic State Estimator

The cerebellum has an important role to play in the control and coordination of movements, which are ordinarily carried out in a very smooth and almost effortless manner. The fundamental issue to be resolved here is whether the cerebellum plays the role of a controller or a dynamic state estimator, in light of what we know about modern control theory. Unfortunately, this issue cannot be resolved solely on the basis of the evidence that cerebellar damage or disease causes inaccuracy or instability of movements.

The key point in support of the dynamic state-estimation hypothesis is embodied in the following statement, the validity of which has been confirmed by decades of work on the design of automatic tracking and guidance systems: *Any system, be it a biological or artificial system, required to predict and/or control the trajectory of a stochastic multivariate dynamic system, can only do so by using or invoking the essence of Kalman filtering in one way or another.*

Building on this key point, Paulin (1997) presented several lines of evidence that favor the hypothesis that the cerebellum is a neural analog of a dynamic state estimator. A particular line of evidence presented therein relates to the vestibulo-ocular reflex (VOR), which is part of the oculomotor system. The function of the VOR is to maintain visual (i.e., retinal) image stability by making eye rotations that are opposite to head rotations. This function is mediated by a neural network that includes the cerebellar cortex and vestibular nuclei. Now, from modern control theory we know that a Kalman filter is an optimum linear system with minimum mean-square error for predicting the state trajectory of a dynamic system using noisy measurements; it does so by estimating the particular state trajectory that is most likely, given an assumed model for the underlying dynamics of the system. A consequence of this strategy is that when the dynamic system deviates from the assumed model, the Kalman filter makes estimation errors of a predictable kind, which may be attributed to the filter believing in the assumed model rather than the actual sense data. According to Paulin (1997), estimation errors of this kind are observed in the behavior of the VOR.

The important point to note, in this brief discussion of the hypothesis that the cerebellum is a neural analog of a Kalman filter, is that the hypothesis is *not* to be taken to imply that the cerebellum physically resembles a Kalman filter. Rather, the cerebellum may provide information in the nervous system, which is analogous to state estimation in a Kalman filter.

From Kalman Filters to Particle Filters

Many dynamical phenomena, whether biological or physical, that are encountered in practice are inherently very complex, involving one or more of the following elements: nonlinearity, non-Gaussianity, and high dimensionality. The EKF deals with the first two elements by doing two things:

- Localized linearization of the nonlinear functions in the process and measurement equations by retaining first-order terms in their Taylor series expansions.
- Approximations of the process and measurement processes by Gaussian processes that are propagated analytically through the first-order approximations of the nonlinear functions.

The practical limitation of this approach is that it may produce large errors in the state estimates.

An analytic approach to circumvent this problem is to use the unscented Kalman filter, which builds on the *unscented transformation* due to Julier, Uhlmann, and Durrant-Whyte (1995). In this new transformation, the input stochastic process is again approximated by a Gaussian process, but the nonlinear transformation is treated in a special way involving a carefully chosen set of sample points. To be specific, let the vector $\bar{\mathbf{x}}$ and the matrix \mathbf{K}_x respectively denote the mean and covariance of a stochastic process $\mathbf{x}(n)$ whose dimensionality is M . Let $\mathbf{x}(n)$ be propagated through a nonlinear function: $\mathbf{y} = \mathbf{f}(\mathbf{x})$. To calculate the mean and covariance of the output vector $\mathbf{y}(n)$, we first form a set of $2M + 1$ *sigma vectors*, defined by

$$\left. \begin{aligned} \chi_0 &= \bar{\mathbf{x}} \\ \chi_i &= \bar{\mathbf{x}} + \sqrt{M + \lambda} (\mathbf{K}_x^{1/2})_i \quad i = 1, 2, \dots, M \\ \chi_i &= \bar{\mathbf{x}} - \sqrt{M + \lambda} (\mathbf{K}_x^{1/2})_i \quad i = M + 1, M + 2, \dots, 2M \end{aligned} \right\} \quad (10)$$

where λ is a scaling factor under the designer's control, and $(\mathbf{K}_x^{1/2})_i$ is the i th column of the square root of matrix \mathbf{K}_x (i.e., lower triangular matrix in the Cholesky factorization of \mathbf{K}_x). The mean and covariance of the nonlinearly transformed vector $\mathbf{y}(n)$ are ob-

tained by using weighted sums of a new vector \mathbf{y}_i related to the sigma vectors as

$$\mathbf{Y}_i = \mathbf{f}(\chi_i), \quad i = 1, 2, \dots, M$$

For an arbitrary nonlinearity, the deceptively simple unscented transformation produces approximations that are accurate to third order for Gaussian inputs, and at least second order for non-Gaussian inputs. The *unscented Kalman filter* (UKF) is a straightforward extension of the unscented transformation (Wan and van der Merwe, 2001). It is a derivative-free state estimator in that, by using multiple forward propagations, the need for explicit computation of Jacobians is avoided; hence, differentiable nonlinear functions are no longer a necessary requirement. The computational complexity of the UKF is, in general, $O(M^3)$; but under special conditions the computation can be restructured to be $O(M^2)$, which is essentially the same as that for the EKF.

The EKF and UKF rely on the use of approximations to ensure mathematical tractability. To avoid approximations, we have to sacrifice mathematical tractability. This is precisely what is done in particle filters, which are rooted in Bayesian theory (Bernardo and Smith, 1998) and Monte Carlo simulation (Robert and Casella, 1999).

Particle filters provide a tool for recursive computation of a stochastic point-mass approximation to the posterior distribution of the hidden states of a nonlinear dynamic system, given a set of observations related to the states. These filters are based on the following stochastic model (Andrieu, Doucet, and Punskeya, 2001):

1. The hidden states are described by the initial distribution $p(\mathbf{x}(0))$ and transition distribution $p(\mathbf{x}(n)|\mathbf{X}_{n-1})$.
2. The observations $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n)$ are conditionally independent, given the likelihood $p(\mathbf{y}(n)|\mathbf{Y}_{n-1}, \mathbf{x}(n))$.

The goal of particle filtering is to recursively estimate the *posterior distribution* $p(\mathbf{X}_n|\mathbf{Y}_n)$, the *filtering distribution* $p(\mathbf{x}(n)|\mathbf{Y}_n)$, and certain expectations such as the conditional mean and conditional covariance of $\mathbf{x}(n)$, where \mathbf{X}_n denotes the sequence $\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(n)$, and \mathbf{Y}_n denotes the sequence $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n)$. Note that the model described under points 1 and 2 defines a generic dynamic model, embodying hidden Markov models as a special case.

In Monte Carlo simulation, a set of weighted, independent, and identically distributed particles (i.e., samples) is drawn from the posterior distribution, thereby mapping integrals to discrete sums. Typically, it is not possible to sample directly from the posterior distribution; hence the recourse to *importance sampling* from an arbitrary *proposal distribution*. The choice of the proposal is a major design issue. In this context, we may use approximations based on the EKF or UKF to generate Gaussian proposal distributions. Basically, the use of UKF here leads to the formulation of the unscented particle filter (Wan and van der Merwe, 2001). Some nice results on image tracking using the unscented particle filter are reported in Rui and Chen (2001).

A limitation of importance sampling is that it does not lend itself to recursive estimation. To get around this problem, we may use a constrained version of importance sampling known as *sequential importance sampling*. The use of this approach yields a set of parameters known as *normalized importance weights*; they are involved in recursively computing the conditional expectations of interest.

Unfortunately, sequential importance sampling has a serious limitation of its own: The variance of the normalized importance weights increases stochastically over time. Typically, after a few iterations the normalized weights of a large number of particles (samples) become practically insignificant, with the result that they

are removed from the sample set, in which case the algorithm no longer adequately represents the posterior distribution of interest. This degeneracy problem is avoided by using a *bootstrap filter*, which involves the elimination of samples with small weights and the multiplication of samples with large weights. The bootstrap filter is essentially modular in extent and therefore simple to implement.

A particle filter may be made highly efficient (i.e., the variance of the estimation error is reduced) by using the so-called Rao-Blackwellization procedure (Robert and Casella, 1999). In so doing, each particle is replaced by a Gaussian distribution propagated through a Kalman filter. The Rao-Blackwellized particle filter represents a stochastic bank (i.e., mixture) of standard Kalman filters (de Freitas, 2002).

To sum up, particle filters offer a powerful tool for sequential state estimation under full nonlinear and non-Gaussian conditions. Although it is computationally intensive, it lends itself to straightforward implementation on a parallel computer.

Discussion

Kalman filtering is a powerful idea rooted in modern control theory and adaptive signal processing; it has withstood the test of time since 1960. Under the ideal conditions of linearity and Gaussianity, the Kalman filter produces an estimate of the hidden state of a dynamic system, with the estimate being optimum in the mean-square-error sense or, equivalently, the maximum likelihood sense. The state-estimation procedure is recursive, which makes it well suited for implementation on a digital computer.

In practical terms, the Kalman filter provides an indispensable tool for the design of automatic tracking and guidance systems, and an enabling technology for the design of recurrent multilayer perceptrons that can simulate any finite-state machine. In the context of neurobiology, Kalman filtering provides invaluable insight into visual recognition and motor control.

The classic Kalman filter and its extensions, namely, the extended Kalman filter and the unscented Kalman filter, offer mathematical tractability by invoking certain approximations and Gaussian assumptions. When the issue of approximations and Gaussian assumptions is of serious concern, we may resort to another powerful tool, particle filters, which are rooted in Bayesian theory and Monte Carlo simulation. Again, particle filters are computationally

intensive, but with a parallel computer, they can be implemented in a straightforward modular fashion.

Acknowledgments. Input from Michael Arbib, Sue Becker, Nando de Freitas, and Ron Racine is much appreciated.

Road Maps: Applications; Vision

Related Reading: Filtering, Adaptive; Sensorimotor Learning

References

- Andrieu, C., Doucet, A., and Punskeya, E., 2001, Sequential Monte Carlo methods for optimal filtering, in *Sequential Monte Carlo Methods in Practice* (A. Doucet, N. de Freitas, and N. Gordon, Eds.), New York: Springer-Verlag. ♦
- Bernardo, J. M., and Smith, A. F. M., 1998, *Bayesian Theory*, New York: Wiley.
- de Freitas, N., 2002, Rao-Blackwellised particle filtering for fault diagnosis, presented at a meeting of the IEEE AC, paper no. 493.
- Haykin, S., 2002, *Adaptive Filter Theory*, 4th ed., Englewood Cliffs, NJ: Prentice-Hall. ♦
- Haykin, S., 1999, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Englewood Cliffs, NJ: Prentice-Hall.
- Jazwinski, A. H., 1970, *Stochastic Processes and Filtering Theory*, New York: Academic Press.
- Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H., 1995, A new approach for filtering nonlinear systems, in *Proceedings of the American Control Conference*, pp. 1628–1632.
- Kalman, R. E., 1960, A new approach to linear filtering and prediction problems, *Trans. ASME J. Basic Engr.*, 82:35–45.
- Patel, G. S., Becker, S., and Racine, R., 2001, Learning shape and motion from image sequences, in *Kalman Filtering and Neural Networks* (S. Haykin, Ed.), New York: Wiley, pp. 69–81.
- Paulin, M. G., 1997, Neural representations of moving systems, *Int. Rev. Neurobiol.*, 41:515–533. ♦
- Puskorius, G. V., and Feldkamp, L. A., 2001, Parameter-based Kalman filter training: Theory and implementation, in *Kalman Filtering and Neural Networks* (S. Haykin, Ed.), New York: Wiley, pp. 23–67.
- Rao, R. P. N., and Ballard, D. H., 1997, “Dynamical model of visual recognition predicts response properties in the visual cortex,” *Neural Computation*, vol. 9, pp. 721–763. ♦
- Robert, C. P., and Casella, G., 1999, *Monte Carlo Statistical Methods*, New York: Springer-Verlag.
- Rui, Y., and Chen, Y., 2001, Better proposed distributions: Object tracking using unscented particle Filter, *IEEE CPVR*, vol. II, pp. 786–793.
- Wan, E. A., and van der Merwe, R., 2001, The unscented Kalman filter, in *Kalman Filtering and Neural Networks* (S. Haykin, Ed.), New York: Wiley, pp. 221–280.

Laminar Cortical Architecture in Visual Perception

Stephen Grossberg

Introduction

The cerebral cortex is the seat of the highest forms of biological intelligence in all sensory and cognitive modalities. Neocortex has an intricate design that exhibits a characteristic organization into six distinct cortical layers. Differences in the thickness of these layers and in the sizes and shapes of neurons led the German anatomist Korbinian Brodmann to identify more than 50 divisions, or areas, of neocortex. This classification has been invaluable as a basis for classifying distinct functions of different parts of neocortex. The functional utility of the laminar organization itself in the control of behavior has, however, remained a mystery until re-

cently. Several models of visual cortex (e.g., Li, 1998; Stemmler, Usher, and Niebur, 1995; Somers et al., 1998; Yen and Finkel, 1998) have clarified aspects of cortical dynamics, but have not articulated how the laminar architecture of cortex contributes to visual perception. A LAMINART model has recently proposed clear functional roles for these layers for the purposes of visual perception (Grossberg, 1999a; Grossberg and Raizada, 2000). The present article uses this model as an organizing theme with which to integrate the analysis and explanation of a variety of data about visual perception and neuroscience. Additional recent research suggests that the functional roles for cortical layers that are proposed by the model may also generalize, with appropriate specializations, to other forms of sensory and cognitive processing.

Bottom-up, top-down, and horizontal interactions are well known to occur within and between the cortical layers. The model proposes how these interactions help the visual cortex to realize (1) the binding process whereby cortical groups distributed data into coherent object representations, (2) the attentional process whereby cortex selectively processes important events, and (3) the developmental and learning processes whereby cortex shapes its circuits to match environmental constraints. It is suggested that the mechanisms that achieve the third property imply the first and second properties. That is, constraints that control stable cortical self-organization in the infant seem to strongly constrain properties of learning, perception, and attention in the adult.

Perceptual Grouping and Attention

During visual perception, the visual cortex can generate perceptual groupings and can focus attention on objects of interest. *Perceptual grouping* is the process whereby the brain organizes image contrasts into emergent boundary structures that segregate objects and their backgrounds in response to texture, shading, and depth cues in scenes and images. Perceptual grouping is a basic step in solving the “binding problem,” whereby spatially distributed features are bound into representations of objects and events in the world. Vivid perceptual groupings, such as illusory contours, can form over image positions that do not receive contrastive bottom-up inputs from an image or scene. Perceptual groupings can form *preattentively* and automatically, without requiring the conscious attention of a viewing subject.

Attention enables humans and other animals to selectively process information that is of interest to them. In contrast to perceptual grouping, top-down attention typically does not form visible percepts over positions that receive no bottom-up inputs. Attention can modulate, sensitize, or prime an observer to expect an object to occur at a given location or with particular stimulus properties. But were attention by itself able to routinely generate fully formed perceptual representations at positions that did not receive bottom-up inputs, then we could not easily tell the difference between external reality and internal fantasy, and we would experience hallucinations all the time. In fact, it has been proposed that a breakdown in this modulatory property of attention can give rise to hallucinations in patients with mental disorders like schizophrenia.

Despite the fact that perceptual grouping and attention make opposite requirements on bottom-up inputs, many recent experiments have shown that perceptual grouping and attention can occur simultaneously within the same circuits of the visual cortex, notably cortical areas V1 and V2 (see Grossberg, 1999a, and Grossberg and Raizada, 2000, for reviews). How is this possible? How does cortical circuitry form perceptual groupings that can complete a boundary grouping over locations that receive no bottom-up visual inputs, whereas top-down attention cannot do so? Why should attention be deployed throughout the visual cortex, including cortical areas that previously were thought to accomplish purely preattentive processing? An answer can be found by exploring the link between attention and learning.

Attention and Learning

Top-down attention has been proposed to be a key mechanism whereby the brain solves the *stability-plasticity* dilemma (Grossberg, 1999b). The stability-plasticity dilemma concerns that fact that our brains can rapidly learn enormous amounts of information throughout life, without just as rapidly forgetting what they already know. Brains are *plastic* and can rapidly learn new experiences, without losing the *stability* that prevents catastrophic forgetting. How are attentive processes realized within neocortex in order to stabilize the learning process?

An improper solution to this problem could easily lead to an infinite regress. This is true because perceptual groupings can form preattentively and provide the substrate on which higher-level attentional processes can act. How can the preattentive grouping mechanisms develop in a stable way, before higher-order attentional processes can develop with which to stabilize them? How can you use attentional mechanisms to stabilize the formation of preattentive grouping circuits, if these attentional mechanisms cannot develop until the preattentive grouping mechanisms do? This is called the *attention-preattention interface problem*. Below we discuss the possibility that the laminar circuits of visual cortex enable preattentive grouping processes to use some of the same circuitry that attentive mechanisms use, even before attentive mechanisms come into play, in order to stabilize their own cortical development and learning. Preattentive grouping uses top-down *intracortical* feedback between the layers, whereas attention uses top-down *intercortical* feedback between them. Both feedback processes converge on a shared decision circuit that helps to determine which perceptual groupings will be perceived.

A solution to the attention-preattention interface problem can be derived from earlier efforts to understand how attention helps to solve the stability-plasticity dilemma: bottom-up signals activate top-down expectations, whose signals are matched against bottom-up data. Both the bottom-up and the top-down pathways contain adaptive weights, or long-term memory traces, that can be modified by experience. The learned top-down expectations “focus attention” on information that matches them. They select, synchronize, and amplify the activities of cells within the attentional focus while suppressing the activities of irrelevant cells, which could otherwise be incorporated into previously learned memories and thereby destabilize them. The cell activities that survive such top-down attentional focusing rapidly reactivate bottom-up pathways. The amplified, synchronized, and prolonged activation of cells within the bottom-up and top-down signal exchanges form a *resonant* state. Such resonances rapidly bind distributed information at multiple levels of brain processing into context-sensitive representations of objects and events. The greater activity, duration, and synchrony of these resonances can support slower processes of learning; hence the term *adaptive resonance*. ADAPTIVE RESONANCE THEORY (q.v.), or ART, has been developed to quantitatively explain how processes of learning, expectation, attention, synchronization, memory search, and consciousness are linked in both healthy subjects and clinical patients. A rapidly growing body of neurobiological data has begun to confirm the predicted links between learning, top-down matching, attention, synchronization, and consciousness.

Learning can easily lead to catastrophic forgetting in response to a changing world. Many popular neural models experience such catastrophic forgetting, notably feedforward models such as back-propagation. ART shows how top-down attention can stabilize learning if it satisfies four properties that together are called the ART matching rule:

1. *Bottom-up automatic activation*: A cell, or cell population, can become active enough to generate output signals if it receives a large enough bottom-up input, other things being equal. Such an input can drive the cell to supraliminal levels of activation.
2. *Top-down priming*: A cell becomes subliminally active if it receives only a large top-down expectation input. Such a top-down priming signal can sensitize, or modulate, the cell and thereby prepare it to react more quickly and vigorously to subsequent bottom-up inputs that match the top-down prime. The top-down prime by itself, however, cannot generate supraliminal output signals from the cell.
3. *Match*: A cell is activated if it receives large convergent bottom-up and top-down inputs. Such a matching process can generate enhanced activation as resonance takes hold.

4. **Mismatch:** A cell's activity is suppressed, even if it receives a large bottom-up input, if it also receives only a small, or zero, top-down expectation input.

Recent data analyses have suggested that variants of the simplest circuit (Figure 1), a top-down on-center off-surround network, is used by the brain (Grossberg, 1999b). In such a circuit, when only bottom-up signals are active, all cells can fire that receive large enough inputs. When only top-down attention is active, cells in the off-surround that receive inhibition but no excitation can be strongly inhibited, while cells in the on-center that receive a combination of excitation and inhibition can become at most subliminally activated, owing to the balance between excitation and inhibition. When bottom-up and top-down inputs match (pathway 2 in Figure 1C), the two excitatory sources of excitation (bottom-up and top-down) that converge at the cell can overwhelm the one inhibitory source; it is a case of two against one. When bottom-up and top-down inputs mismatch (pathway 1 in Figure 1C), the top-down inhibition can neutralize the bottom-up excitation; it is a case of one against one.

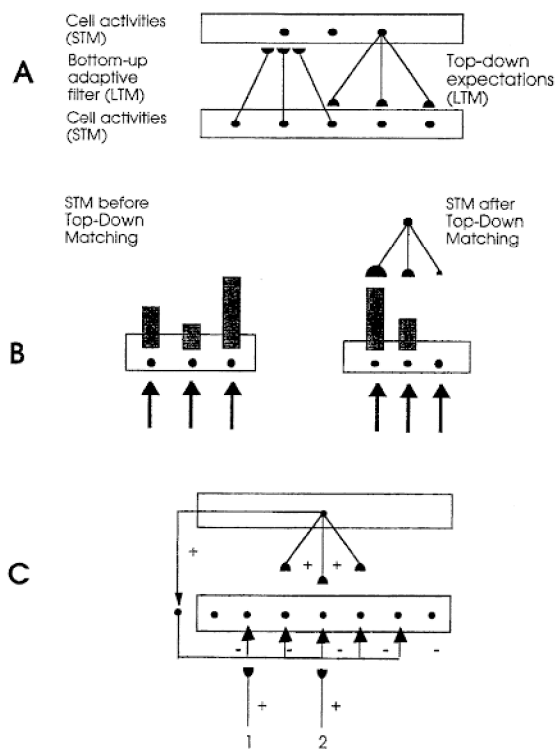


Figure 1. A, Patterns of activation, or short-term memory (STM), on a lower processing level send bottom-up signals to a higher processing level. These signals are multiplied by adaptive weights, or learned long-term memory (LTM) traces, which influence which cells are activated at the higher processing level. These latter cells, in turn, activate top-down expectation signals that are also multiplied by learned LTM traces. These top-down expectations are matched against the STM pattern that is active at the lower level. B, This matching process selects, amplifies, and synchronizes STM activations that are supported by large LTM traces in an active top-down expectation, and suppresses STM activations that do not get top-down support. The size of the hemidisks at the end of the top-down pathways represents the strength of the learned LTM trace that is stored in that pathway. C, The ART matching rule can be realized by a modulatory top-down on-center off-surround network, as discussed in the text. (From Grossberg, S., 1999a, *How does the cerebral cortex work? Learning, attention, and grouping by the laminar circuits of visual cortex*, *Spatial Vision*, 12:163–185. Reprinted with permission.)

Attention Is Modulatory

The ART matching rule predicted that top-down attention is part of a modulatory *priming* and *matching* process. By itself, attention cannot supraliminally activate cells, so they cannot generate output signals. Data compatible with this prediction have gradually been reported over the years. For example, Zeki and Shipp (1988, p. 316) wrote that “backward connections seem not to excite cells in lower areas, but instead influence the way they respond to stimuli.” Likewise, the data of Sillito et al. (1994) on attentional feedback from V1 to the lateral geniculate nucleus (LGN) led them to conclude that “the cortico-thalamic input is only strong enough to exert an effect on those dLGN cells that are additionally polarized by their retinal input . . . the feedback circuit searches for correlations that support the ‘hypothesis’ represented by a particular pattern of cortical activity.” Their experiments demonstrated all of the properties of the ART matching rule, since they found in addition that “cortically induced correlation of relay cell activity produces coherent firing in those groups of relay cells with receptive-field alignments appropriate to signal the particular orientation of the moving contour to the cortex . . . this increases the gain of the input for feature-linked events detected by the cortex.” In other words, top-down priming by itself cannot fully activate LGN cells; it needs matched bottom-up retinal inputs to do so, and those LGN cells whose bottom-up signals support cortical activity get synchronized and amplified by this feedback. In addition, anatomical studies have shown that the top-down V1 to LGN pathway realizes a top-down on-center off-surround network.

How to Stabilize Cortical Development and Learning

The preceding discussion suggests that top-down attentional mechanisms should be present in *every* cortical area in which self-stabilizing learning can occur, since without top-down learned expectations that focus attention, any such learned memories could easily be degraded due to catastrophic forgetting.

These analyses should, in particular, apply to the perceptual grouping process, because the cortical horizontal connections that support perceptual grouping in cortical areas like V1 develop through a learning process that is influenced by visual experience (e.g., Calloway and Katz, 1990; Antonini and Stryker, 1993). It is also known that many developmental and learning processes, including those that control horizontal cortical connections, are stabilized dynamically and can be reactivated by lesions and other sources of cortical imbalance (Das and Gilbert, 1995); and that adult learning uses the same types of mechanisms as the infant developmental processes on which it builds (Kandel and O’Dell, 1992). What cortical mechanisms ensure this type of dynamical stability?

This is a particularly challenging problem for perceptual groupings because they can generate suprathreshold responses over positions that do not receive bottom-up inputs. They therefore seem to violate the ART matching rule. How, then, can the horizontal connections that generate perceptual groupings maintain themselves in a stable way? Why are they not washed away whenever an illusory contour grouping forms over positions that do not receive a bottom-up input? The LAMINART model proposes an answer to this question that clarifies how attention, perceptual grouping, development, and perceptual learning work and interact within the laminar circuits of visual cortex.

Preattentive Mechanisms of Perceptual Grouping

Four circuit properties summarize this proposal of how the visual cortex, notably areas V1 and V2, uses its laminar design to generate coherent perceptual groupings that maintain their analog sensitivity

to environmental inputs, the so-called property of *analog coherence*. Four additional circuit properties will then be summarized whereby attention, development, and learning may be integrated into this laminar design. Each of these design constraints is supported by neurophysiological, anatomical, and psychophysical data.

Analog Sensitivity to Bottom-Up Sensory Inputs

Bottom-up inputs from the retina go through the LGN on their way to cortex. LGN outputs directly excite layer 4. LGN inputs also excite layer 6, which then indirectly influences layer 4 via an on-center off-surround network of cells, as in Figure 2A. The net effect of LGN inputs on layer 4 cells is thus via an on-center off-surround network. Such a feedforward on-center off-surround network of cells can preserve the analog sensitivity of, and normalize, the activities of target cells if these cells obey the membrane equations of neurophysiology. Such a network can preserve the analog sensitivity of layer 4 cells in response to LGN inputs that may vary greatly in intensity.

Bipole Boundary Grouping

The active layer 4 cells input to pyramidal cells in layer 2/3. These cells initiate the formation of perceptual groupings. They generate excitatory signals among themselves using monosynaptic long-range horizontal connections, and inhibition using short-range disynaptic inhibitory connections, as in Figure 2B. These interactions support inward perceptual groupings between two or more boundary inducers, as in the case of illusory contours, but not outward groupings from a single inducer, which would fill the visual field with spurious groupings.

These grouping properties may be ensured as follows. When a single active pyramidal cell sends horizontal monosynaptic excitation to other pyramidal cells, this excitation is inhibited by the disynaptic inhibition that it also generates; this balance between excitation and inhibition is a case of one against one. Model simulations have shown that such an approximate balance between excitation and inhibition is needed to stabilize the development of horizontal connections. A different result obtains when two or more pyramidal cells are activated at positions that are located at opposite sides of a target pyramidal cell, and all the cells share approximately the same orientation preference and are approximately colinear across space. Then the excitation from the active pyramidal cells summates at the target cell, thereby generating a larger total excitatory input than a single pyramidal cell could. In addition, the active cells excite a single population of disynaptic inhibitory interneurons, which generates a saturating, or normalized, inhibitory output to the target cell. Thus, excitation is bigger than inhibition in this case, so that grouping can occur; it is a case of two against one. This combination of constraints is called the *bipole* property. Layer 2/3 pyramidal cells may thereby become active either because of direct inputs from layer 4, or because of bipole boundary groupings that form in response to other active layer 2/3 cells.

Folded Feedback and Analog Coherence

The active cells in layer 2/3 can form groupings on their own in response to unambiguous visual inputs. In response to scenes wherein multiple groupings are possible but only a few of them are correct, intracortical feedback helps to select the correct cells, and also binds them together in a coherent way. This selection happens when active cells in layer 2/3 send excitatory feedback signals to layer 6 via layer 5, as in Figure 2C. Layer 6 then activates the on-center off-surround network from layer 6 to 4. This feedback process is called *folded feedback*, because feedback signals from layer

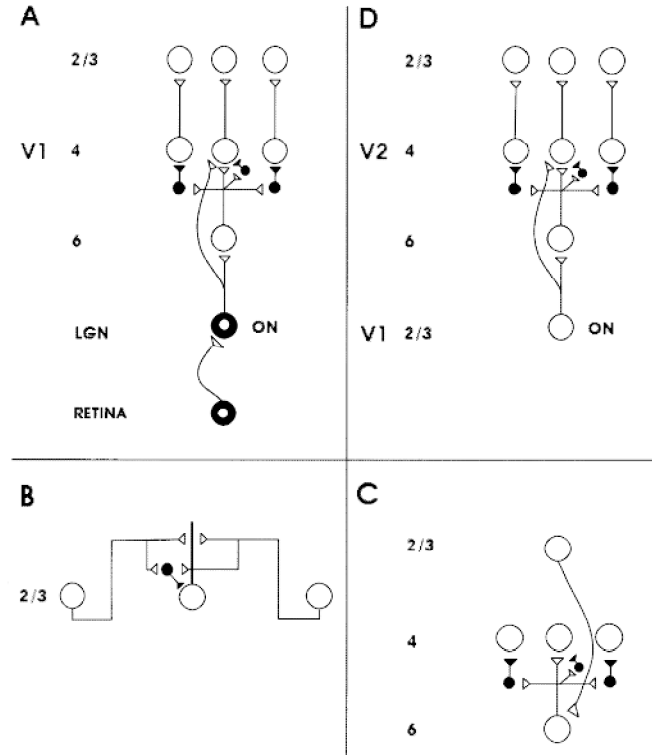


Figure 2. A model circuit of retinal, lateral geniculate nucleus (LGN), and cortical V1 interactions. Open symbols indicate excitatory interactions and closed symbols indicate inhibitory interactions. **A**, Feedforward circuit from retina to LGN to cortical layers 4 and 6. Retina: Retinal ON cells have an on-center off-surround organization. Retinal OFF cells have an off-center on-surround organization. LGN: The LGN ON and OFF cells receive feedforward ON and OFF cell inputs from the retina. Layer 4: Layer 4 cells receive feedforward inputs from LGN and layer 6. LGN ON and OFF cell excitatory inputs to layer 4 directly establish oriented simple-cell receptive fields. Layer 6 cells excite layer 4 cells with a narrow on-center and inhibit them using inhibitory interneurons that span a broader off-surround, which includes cells in the on-center (not shown). Like-oriented layer 4 simple cells with opposite-contrast polarities compete (not shown) before generating half-wave rectified outputs that converge on layer 2/3 pyramidal (complex) cells. Layer 2/3: The converging simple cell outputs enable complex cells to respond to both polarities. They thereby full-wave rectify the image. **B**, Horizontal grouping interactions in layer 2/3: After being activated by inputs from layer 4, layer 2/3 pyramidal (complex) cells excite each other monosynaptically via horizontal connections, primarily on their apical dendrites. They also inhibit one another via disynaptic inhibition that is mediated by model smooth stellate cells. Multiple horizontal connections share a common pool of stellate cells near each target pyramidal cell. This ensures that boundaries form inwardly between pairs or greater numbers of boundary inducers, but not outwardly from a single inducer. **C**, Cortical feedback loop from layer 2/3 to layer 6: Layer 6 cells receive excitatory inputs from layer 2/3. The long-range cooperation thereby engages the feedforward layer 6-to-4 on-center off-surround network, which then reactivates layer 2/3 cells. This “folded feedback” loop can select winning groupings without a loss of analog coherence. **D**, Outputs from layer 2/3 to area V2 directly excite layer 4 cells and layer 6 cells, which indirectly influence layer 4 cells via an on-center off-surround network, as in area V1. (From Grossberg, 1999a).

2/3 to layer 6 get transmitted in a feedforward fashion back to layer 4; that is, feedback is “folded” back into the feedforward flow of bottom-up information within the laminar cortical circuits.

Folded feedback turns the cortex into a feedback network that binds the cells throughout layers 2/3, 4, and 6 into functional col-

umns. The on-center off-surround network also helps to select the strongest groupings that are formed in layer 2/3 and to inhibit weaker groupings, while preserving the analog values of the selected groupings. In particular, the on-center signals from layer 6-to-4 support the activities of those pyramidal cells in layer 2/3 that are part of the strongest horizontal groupings. The off-surround signals can inhibit inputs to layer 4 that were supporting less active groupings in layer 2/3. In this way, signals from layer 4 to the less active groupings in layer 2/3 are removed, and thus these groupings collapse.

Self-Similar Hierarchical Boundary Processing

Converging evidence suggests that area V2 replicates aspects of the structure of area V1, but at a larger spatial scale. Thus, layer 2/3 in area V1 sends bottom-up inputs to layers 4 and 6 of area V2, much as LGN sends bottom-up inputs to layers 4 and 6 of area V1, as in Figure 2D. This input pattern from V1 to V2 can preserve the analog sensitivity of layer 4 cells in V2 for the same reason that the LGN inputs to V1 can preserve the analog sensitivity of layer 4 cells in V1. The shorter perceptual groupings in layer 2/3 of area V1 are proposed to group together, and enhance the signal-to-noise ratio of, nearby V1 cells with similar orientation and disparity selectivity. The longer perceptual groupings in area V2 are proposed to build long-range boundary segmentations that separate figure from background; generate 3D groupings of the edges, textures, shading, and stereo information that go into object representations; and complete boundaries across gaps in bottom-up signals due to the retinal blind spot and veins (Grossberg, 1994).

Attention, Development, and Learning

The following four circuit properties are proposed to integrate top-down attention into the preattentive grouping process in a way that solves the attention-preattention interface problem and enables grouping circuits to develop and learn in a stable way.

Top-Down Feedback from V1 to LGN

As noted above, layer 6 of area V1 sends a top-down on-center off-surround network to the LGN, as in Figure 3A. This top-down pathway automatically gain-controls and focuses attention on those LGN cells whose activities succeed in activating V1 cells. Data of Sillito et al. (1994) are compatible with the hypothesis that this feedback obeys the ART matching rule, and thus can only subliminally activate, or modulate, LGN cells. Matched bottom-up inputs are needed to supraliminally activate LGN cells while top-down signals are active. This process is predicted to help stabilize the development of receptive fields in V1, including disparity-tuned complex cells, during the visual critical period.

Folded Feedback from Layer 6 of V2 to Layer 4 of V1

A similar top-down process seems to occur at all stages of visual cortex, and probably beyond. Layer 6 in a given cortical area, such as V2, generates top-down cortical signals to layer 6 of lower cortical areas, such as V1, where they activate the layer 6-to-4 folded feedback network in the lower area (Figure 3B). One such known top-down pathway exits layer 6 in V2 and activates V1 via layer 1, then layer 5, then layer 6, as in Figure 3C. Top-down feedback can thereby activate a top-down on-center off-surround circuit, as required by the ART matching rule. Intercortical attention is here-with suggested to use outputs from layer 6 of a given cortical area to activate layer 4 of a lower cortical area via layer 6-to-4 folded feedback.

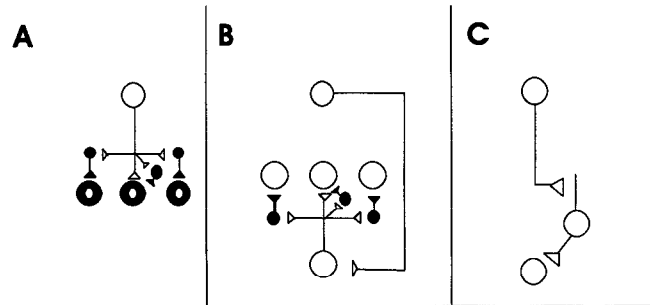


Figure 3. A, Top-down corticogeniculate feedback from layer 6: LGN ON and OFF cells receive topographic excitatory feedback from layer 6 in V1, and more broadly distributed inhibitory feedback via LGN inhibitory interneurons that are excited by layer 6 signals. The feedback signals pool outputs over all cortical orientations and are delivered equally to ON and OFF cells. Corticogeniculate feedback selects, gain-controls, and synchronizes LGN cells that are consistent with the cortical activation that they cause, thereby acting like a type of automatic attentional focus. B, Attentional feedback from V2 to V1: Layer 6 in V2 activates layer 6 in V1, which then activates the layer 6-to-4 on-center off-surround network that attentionally primes layer 4 cells. C, One feedback pathway arises from layer 6 cells in V2 and activates apical dendrites in layer 1 of V1. Cells in layer 5 are activated through these apical dendrites and thereupon activate layer 6 cells. Layer 6 in V2 can also modulate layer 2/3 of V1 by activating layer 1 dendrites of both excitatory and inhibitory cells in layer 2/3. (From Grossberg, 1999a).

Layer 6-to-4 Signals Are Subliminal

The ART matching rule predicts that this top-down pathway subliminally activates, or modulates, cells in layer 4. This modulatory property is predicted to be due to the fact that the excitatory and inhibitory signals within the on-center from layer 6-to-4 are approximately balanced, so that at most, a weak excitatory effect occurs after activating the circuit via top-down feedback. Consistent data show that “feedback connections from area V2 modulate but do not create center-surround interactions in V1 neurons” (Hupé et al., 1997, p. 1031) and that top-down connections have an on-center off-surround organization (Bullier et al., 1996). Model simulations have shown that that this approximate balance is needed to achieve stable development of interlaminar 6-to-4 connections.

Although it is modulatory, this top-down circuit can have a major effect on cortical cell activations when the cortex is activated bottom-up by visual inputs: it can strongly inhibit activities of layer 4 cells whose layer 2/3 cell projections are not bound into strong groupings, and amplify the strongest groupings until they can resonate. A competitive effect of top-down attention has been reported in the neurophysiological experiments of Reynolds, Chelazzi, and Desimone (1999). Its laminar substrates have not yet been tested, however. By using such an attentional mechanism, higher-level influences such as figure-ground separation or even learned object prototypes can bias the cortex to select consistent groupings at lower cortical levels. In this way, automatic early vision filtering, 3D boundary and surface processing, and higher-order knowledge constraints can mutually influence one another.

Two Bottom-Up Input Sources to Layer 4

A simple functional explanation can now be given of a ubiquitous cortical design; namely, why there are direct bottom-up inputs to layer 4, as well as indirect bottom-up inputs to layer 4 via layer 6 (e.g., Figures 2A and 2D). Why are these two separate input pathways not just a gigantic waste of wire? In particular, why is the

indirect layer 6-to-4 pathway not sufficient to fully activate layer 4 cells *and* to maintain their analog sensitivity using its on-center off-surround network? The proposed explanation is that the indirect layer 6-to-4 inputs need to be modulatory to preserve the stability of cortical development and learning. Direct inputs to layer 4 are therefore also needed to fully activate layer 4 cells.

Taken together, these eight cortical design principles lead to the circuit diagram in Figure 4 for perceptual grouping, attention, development, and learning within and between areas LGN, V1, and V2. The generality of the constraints that lead to this design poses the intriguing possibility that the same cortical circuits may explain data at multiple levels and modalities of neocortical sensory and cognitive processing.

The Preattentive Perceptual Grouping Is Its Own Attentional Prime

These circuit constraints suggest how the horizontal connections within cortical area V1 and V2 can develop and learn stably in response to visual inputs, and thereby solve the preattention-attention interface problem: both preattentive perceptual groupings within V1 and attentive feedback from V2 to V1 generate feedback signals to layer 6 of V1, one via intracortical pathways from layer 2/3 of the same cortical area, and the other via intercortical pathways from layer 6 of a higher cortical area. Both types of feedback activate the folded feedback circuit from layer 6-to-4. Top-down attention uses this circuit to focus attention within V1 by inhibiting layer 4 cells that are not supported by excitatory 6-to-4 feedback. Perceptual grouping uses it to select the correct grouping by inhib-

iting layer 4 cells that would otherwise form incorrect groupings. In both cases, folded feedback prevents the wrong combinations of cells in layers 4 and 2/3 from being active simultaneously. In the adult, this selection process defines perceptual grouping properties. In the infant, and also during adult perceptual learning, it prevents incorrect horizontal connections from being learned, since "cells that fire together wire together." This sharing of the layer 6-to-4 selection circuit by both grouping and attention clarifies how attention can propagate along a boundary grouping and can thereby selectively prime an object representation (Grossberg and Raizada, 2000; Roelfsema, Lamme, and Spekreijse, 1998).

The folded feedback circuit from layer 6-to-4 gets activated by perceptual grouping signals from layer 2/3 at *all* positions of the grouping, even positions that do not receive bottom-up inputs. The ART matching rule is thus satisfied at all positions, and the source of the "top-down expectation" is intracortical top-down signals from the perceptual grouping itself. In summary, the *preattentive perceptual grouping is its own attentional prime* because it can use the modulatory 6-to-4 circuit to stabilize its own development using *intracortical* feedback, even before attentional *intercortical* feedback can develop.

Discussion

All sensory and cognitive neocortical areas share key laminar properties. For example, long-range horizontal connections are known to occur in many areas of neocortex, such as the auditory and language areas of the human temporal cortex. Ongoing research suggests that the above principles of how to achieve stable cortical development and learning, to bind together distributed cortical data through a combination of bottom-up adaptive filtering and horizontal associations, and to modulate it with top-down attention generalize to other neocortical areas.

Acknowledgments. Work was supported in part by grants from the Defense Advanced Research Projects Agency and the Office of Naval Research (ONR N00014-95-1-0409), the National Science Foundation (NSF IRI-97-20333), and the Office of Naval Research (ONR N00014-95-1-0657 and ONR N00014-01-1-0624).

Road Maps: Mammalian Brain Regions; Vision

Background: Adaptive Resonance Theory

Related Reading: Contour and Surface Perception; Visual Attention; Visual Cortex: Anatomical Structure and Models of Function

References

- Antonini, A., and Stryker, M. P., 1993, Functional mapping of horizontal connections in developing ferret visual cortex: Experiments and modeling, *J. Neurosci.*, 14:7291–7305.
- Bullier, J., Hupé, J. M., James, A., and Girard, P., 1996, Functional interactions between areas V1 and V2 in the monkey, *J. Physiol. (Paris)*, 90:217–220.
- Calloway, E. M., and Katz, L. C., 1990, Emergence and refinement of clustered horizontal connections in cat striate cortex, *J. Neurosci.*, 10:1134–1153.
- Das, A., and Gilbert, C. D., 1995, Long-range horizontal connections and their role in cortical reorganization revealed by optical recording of cat primary visual cortex, *Nature*, 375:780–784.
- Grossberg, S., 1994, 3-D vision and figure-ground separation by visual cortex, *Percept. Psychophys.*, 55:48–120.
- Grossberg, S., 1999a, How does the cerebral cortex work? Learning, attention, and grouping by the laminar circuits of visual cortex, *Spatial Vision*, 12:163–185. ♦
- Grossberg, S., 1999b, The link between brain learning, attention, and consciousness, *Consciousness Cognit.*, 8:1–44. ♦
- Grossberg, S., and Raizada, R. D. S., 2000, Contrast-sensitive perceptual grouping and object-based attention in the laminar circuits of primary visual cortex, *Vision Res.*, 40:1413–1432.

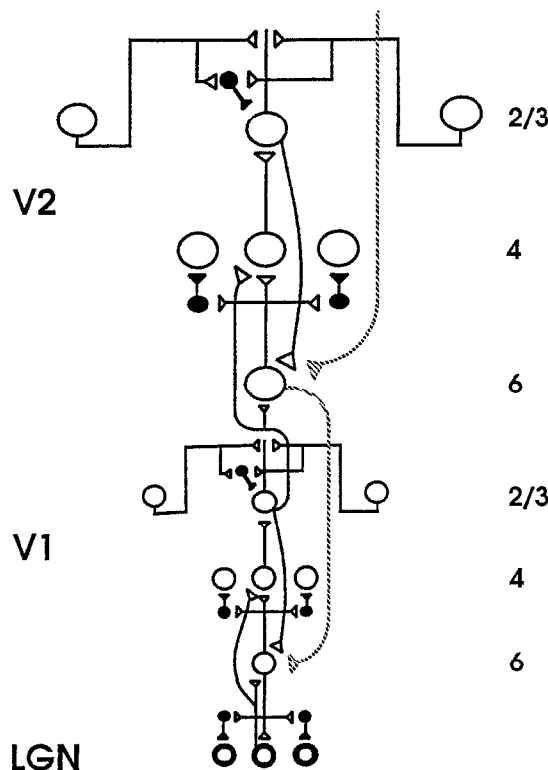


Figure 4. A model synthesis of bottom-up, top-down, and horizontal interactions in LGN, V1, and V2. Cells and connections with open symbols denote preattentive excitatory mechanisms that are involved in perceptual grouping. Closed symbols denote inhibitory mechanisms. Gray denotes top-down attentional mechanisms. (From Grossberg, 1999a).

- Hupé, J. M., James, A. C., Girard, P., and Bullier, J., 1997, Feedback connections from V2 modulate intrinsic connectivity within, *Soc. Neurosci. Abstr.*, 23:1031, abstr. 406.15
- Kandel, E. R., and O'Dell, T. J., 1992, Are adult learning mechanisms also used for development? *Science*, 258:243–245.
- Li, Z., 1998, A neural model of contour integration in the primary visual cortex, *Neural Computat.*, 10:903–940.
- Reynolds, J., Chelazzi, L., and Desimone, R., 1999, Competitive mechanisms subserve attention in macaque areas V2 and V4, *J. Neurosci.*, 19:1736–1753.
- Roelfsema, P. R., Lamme, V. A. F., and Spekreijse, H., 1998, Object-based attention in the primary visual cortex of the macaque monkey, *Nature*, 395:376–381. ♦
- Sillito, A. M., Jones, H. E., Gerstein, G. L., and West, D. C., 1994, Feature-linked synchronization of thalamic relay cell firing induced by feedback from the visual cortex, *Nature*, 369:479–482. ♦
- Somers, D. C., Todorov, E. V., Siapas, A. G., Toth, L. J., Kim, D., and Sur, M., 1998, A local circuit approach to understanding integration of long-range inputs in primary visual cortex, *Cereb. Cortex*, 8:204–217.
- Stemmler, M., Usher, M., and Niebur, E., 1995, Lateral interactions in primary visual cortex: A model bridging physiology and psychophysics, *Science*, 269:1877–1880.
- Yen, S. C., and Finkel, L. H., 1998, Extraction of perceptually salient contours by striate cortical networks, *Vision Res.*, 38:719–741.
- Zeki, S., and Shipp, S., 1988, The functional logic of cortical connections, *Nature*, 335:311–317. ♦

Language Acquisition

Brian MacWhinney

Introduction

Language is a uniquely human achievement. All of the major social achievements of human culture—architecture, literature, law, science, art, and even warfare—rely on the use of language. Although there have been attempts to teach language to primates, the ability to learn language is a distinctive mark of the human species. This view of language led Chomsky (1965) to voice this assessment:

It is, for the present, impossible to formulate an assumption about initial, innate structure rich enough to account for the fact that grammatical knowledge is attained on the basis of the evidence available to the learner. Consequently, the empiricist effort to show how the assumptions about a language acquisition device can be reduced to a conceptual minimum is quite misplaced. The real problem is that of developing a hypothesis about initial structure that is sufficiently rich to account for acquisition of language, yet not so rich as to be inconsistent with the known diversity of language.

To address this challenge, neural network researchers have explored a wide variety of network architectures and linguistic problems. This work has shown how children can learn language without relying on specifically linguistic, innate initial structure. However, several problems must be addressed before we can say that neural networks have answered Chomsky's challenge.

An Example

Let us consider, as an example of this type of research, the neural network developed by MacWhinney et al. (1989). This model was designed to explain how German children learn to select one of the six different forms of the German definite article. In English, we have a single word “the” that serves as the definite article. In German, the article can take the form *der*, *die*, *das*, *des*, *dem*, and *den*, as indicated in Table 1. The choice of a particular form of the article depends on three additional features of the noun: its gender (masculine, feminine, or neuter), its number (singular or plural), and its role within the sentence (subject, possessor, direct object, prepositional object, or indirect object). There are 16 cells in the paradigm for the four cases and four genders (masculine, feminine, neuter, plural), but there are only six forms of the article. This means that a given form of the article, such as *der* can be used for either masculine-nominative-singular or feminine-genitive-singular, and so on.

To make matters worse, assignment of nouns to gender categories in German is quite nonintuitive. For example, the word for

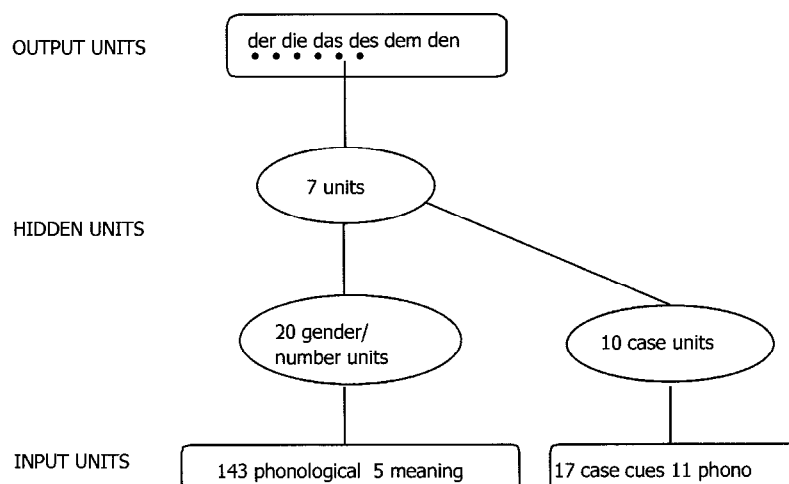
“fork” is feminine, the word for “spoon” is masculine, and the word for “knife” is neuter. Acquiring this system of arbitrary gender assignments is particularly difficult for adult second language learners. Mark Twain expressed his consternation at this aspect of German in a treatise entitled “The awful German language” in which he accuses the language of unfairness and capriciousness in its treatment of young girls as neuter, the sun as feminine, and the moon as masculine. Along a similar vein, Maratsos and Chalkley (1980) argued that, because neither semantic nor phonological cues can predict which article accompanies a given noun in German, children could not learn the language by relying on simple surface cues.

These relations are so complex that a careful linguistic description of the system occupies well over 200 pages. However, MacWhinney et al. (1989) show that it is possible to construct a connectionist network that learns this system from the available cues (see Figure 1). The model uses a simple feedforward architecture. The input is structured into two pools of units. The first pool has 143 phonological units and 5 token meaning units. The second pool has 17 case cues from syntactic structure and 11 phonological cues from endings on the noun. These two input pools feed into two separate pools of collector units that then feed together into a second level of hidden units. The output is a set of six nodes for the six possible forms of the German article. It is important to remember that each of these six articles must serve several functions to fill up the 16 cells of the declensional paradigm.

The network was trained using the backpropagation algorithm. After 40 epochs of training on a set of 102 real German nouns, the network was able to choose the correct article 98% of the time. This meant that it not only succeeded in getting the gender of the noun right, but also figured out how to use the case cues to correctly select one of the 16 cells of the paradigm. To test the network's generalization abilities, we presented it with old nouns in new case roles. In these tests, the network chose the correct article in 92% of trials. This type of cross-paradigm generalization provides clear evidence that the network went far beyond rote memorization during the training phase. In fact, the network quickly succeeded in learning the whole of the basic formal paradigm for the marking of German case, number, and gender on the noun.

In addition, the simulation was able to generalize its internalized knowledge to solve the problem that had so perplexed Mark Twain—guessing at the gender of entirely novel nouns. We presented the network with 48 new high-frequency German nouns in a variety of sentence contexts. On this completely novel set, the simulation chose the correct article from the six possibilities in 61%

Figure 1. A network model of the acquisition of German declensional marking



of trials, versus 17% expected by chance. Thus, the system's learning mechanism, together with its representation of the noun's phonological and semantic properties and the context, produced a good guess about what article would accompany a given noun, even when the noun was entirely unfamiliar. In a subsidiary simulation, we showed that, when the model only has to guess the gender of the noun, and not its position in the paradigm, it achieves over 70% accuracy on new nouns. This is a level that comes close to that achieved by native speakers.

The network's learning paralleled children's learning in a number of ways. Like real German-speaking children, the network tended to overuse the articles that accompany feminine nouns. The reason for this is that the feminine forms of the article have a high frequency, because they are used both for feminines and for plurals of all genders. The simulation also showed the same type of overgeneralization patterns that are often interpreted as reflecting rule use when they occur in children's language. For example, although the noun "Kleid" (clothing) is neuter, the simulation used the initial "kl" sound of the noun to conclude that it was masculine. Because of this, it invariably chose the article that would accompany the noun if it were masculine. Interestingly, the same article-noun combinations that are the most difficult for children were also the most difficult for the network.

Demonstrations of this type illustrate how children can acquire linguistic knowledge without relying on stipulated, hard-wired constraints of the type envisioned by Chomsky. Similar demonstrations have been produced in a wide variety of areas including: the English past tense, Dutch word stress, universal metrical features, German participle acquisition, German plurals, Italian articles, Spanish articles, English derivation for reversives, lexical learning from perceptual input, deictic reference, personal pronouns, polysemic patterns in word meaning, vowel harmony, historical change, early auditory processing, the phonological loop, early phonological output processes, ambiguity resolution, relative clause processing, word class learning, speech errors, bilingualism, and the vocabulary spurt.

Challenges

Researchers have contested the logic underlying these demonstrations. Some of the problems that have been raised relate only to minor implementational features of the earliest models (MacWhinney and Leinbach, 1991), but others are more fundamental. The five most fundamental challenges are:

Dual route. Neural networks provide a good account of associative processes, but fail to account for the learning of regular rules.

Lexical learning. Neural networks have problems learning large numbers of words.

Syntax. Neural networks have problems dealing with the compositional aspects of complex syntax.

Neuronal realism. Some neural network architectures make inappropriate assumptions regarding neural processing.

Embodiment. Neural networks have not been able to model the ways in which the mind is linked to the body.

Let us look at responses to each of these five challenges in greater detail.

Dual Route

Pinker (1999) has been a key proponent of the application of a dual-route model to language acquisition. He contends that irregular forms, such as *fell*, *went*, *feet*, or *broken*, are processed through an associative memory grounded on neural networks, but that regular forms, such as *jumped*, *wanted*, *cats*, and *dropped*, are produced by rule. Pinker views his defense of the psychological reality of linguistic rules as a part of a general defense of the linguistic theory of generative grammar.

The fact that irregulars are processed differently from regulars does not prove the existence of symbolic rules. Neural networks have no problem representing both regular and irregular patterns in a single network. For example, the network developed by Kawamoto (1994) encodes regular forms as nodes in competition with less regular nodes. In that homogeneous recurrent network architecture, regular and irregular forms display quite different temporal activation patterns, in accord with empirical observations. Kawamoto's model also accords with the fact that even the most regular patterns display phonological conditioning and patterns of gradience (Bybee, 1995) of the type modeled by neural networks.

Lexical Learning

Neural networks have problems learning large numbers of words. Typically, neural network architectures have been used primarily as methods for extracting and classifying patterns. Word learning differs from classification in two ways. First, the association between a word's meaning and its sound is almost entirely arbitrary. There is nothing in the specific sounds of *table* that depicts

the shape or purpose of a table. This means that the learning of words cannot rely on the methods for pattern detection that are so important in neural network research. Second, the number of words that a speaker must learn is extremely large. If we look just at word stems, adult English speakers control between 10,000 and 50,000 words. Many of these words have multiple meanings and many can be further combined into compounds and rote phrases. Thus, the effective lexicon of an adult English speaker is from 20,000 to 80,000 words.

Self-organizing maps (SOMs) (Farkas and Li, 2001) offer a promising framework for dealing with the encoding of lexical items. Precision of encoding can be obtained by increasing the dimensionality of the coding space and then recompressing the additional dimensions. Conflicts between related words that are close on the lexical map can trigger a process of focused learning that concentrates specifically on words that are being confused. Catastrophic interference can be avoided by adding new nodes without forgetting older patterns (Hamker, 2001).

Because SOMs provide a relatively local encoding of words, they then allow us to address four additional problems that stem from problems of representing words in neural networks.

U-shaped learning. Children often produce a form like *went* correctly for several weeks or months and then shift to occasionally saying *go-ed*. Later, they move back to saying *went* consistently. This pattern, known as *U-shaped learning*, requires an ability to learn some forms first by rote. Backpropagation networks are good at producing overgeneralizations like *go-ed* but weak at producing and holding on to a rote form like *went* (Plunkett and Marchman, 1993). By default, SOMs place an emphasis on early rote learning and are slower to generalize out the regular patterns.

Homophony. Because most neural network models do not have discrete representations for lexical items, they have problems distinguishing homophonous forms. Consider what happens to the three homophones of the word *ring* in English. We can say *the maid wrung out the clothes*, *the soldiers ringed the city*, or *the choirboy rang the bell*. These three different words all have the same sound /rɪŋ/ in the present, but each takes a different form in the past. In SOMs, these three words have clearly different representations in semantic space.

Compounds. Without discrete representations for lexical items, neural networks have problems with compound words. The fact that the past tense of *undergo* is *underwent* depends on the fact that *undergo* is a variant of the stem *go*. When the compound itself is high enough in frequency, the network can learn to treat it as an irregular. Networks have problems learning the past tense of low-frequency irregular compounds. However, if the network can detect the present of “go” inside “undergo,” it can solve this problem.

Derivational status. Neural networks have problems utilizing information regarding the derivational status of lexical items. In English, the past tense forms of denominal verbs always receive the regular past tense suffix. For example, the word *ring* can be used as a verb in *the groom ringed her finger*, but we would never say *the groom rung her finger*. Without an ability to know that a word derives from a noun, neural networks cannot encode this pattern. German provides even clearer examples of the importance of derivational status. All German nouns that derive from verbs are masculine. For example, the noun *der Schlag* (“blow”; “cream”) derives from the verb *schlagen* (“to hit”). However, there is no motivated way of indicating this in the model. In general, the model includes no independent way of representing morphological relationships between words. Thus, no distinction is made between true

phonological cues such as final /e/ or initial /kn/ and derivational markers for the diminutive, such as *-chen* or *-ett*. This leads to some very obvious confusions. For example, masculines such as *der Nacken* (“neck”) and *der Hafen* (“harbor”) end in phonological /en/, whereas neuters such as *das Wissen* (“knowledge”) and *das Lernen* (“learning”) end in the derivational suffix *-en*. Confusion of these two suffixes leads to inability to correctly predict gender for new nouns ending in /en/. Without having a way of representing the fact that derivational morphemes have an independent lexical status, neural networks cannot process these patterns.

These four difficulties reflect a single core problem. By working with neural networks that flexibly encode lexical items, we can begin to address these additional features of word structure.

Syntax

Elman (1990) has provided demonstrations of the ability of neural networks to process complex syntactic structures. His model uses recurrent connections to update the network’s memory after it listens to each word. The network’s task is to predict the next word. This framework views language comprehension as a highly constructive process in which the major goal is trying to predict what will come next. Psycholinguists recognize the importance of prediction, but they view the major task of language processing as the construction of mental models. It is not clear how understanding prediction will help us understand the construction of mental models, although the two processes are certainly related.

An alternative to the predictive framework relies on the older neural network mechanisms of spreading activation and competition. For example MacDonald, Perlmuter, and Seidenberg (1994) have presented a model of ambiguity resolution in sentence processing that is grounded on competition between lexical items. Models of this type do an excellent job of modeling the temporal properties of sentence processing. Such models assume that the problem of lexical learning in neural networks has been solved. They then proceed to use localist representations to control interactive activation during sentence processing. Until we have indeed solved the problem of lexical learning, this is a very effective way of advancing the research agenda.

Another approach that makes similar assumptions uses a linguistic framework known as Construction Grammar. This framework emphasizes the role of individual lexical items in early grammatical learning (Tomasello, 2000). Early on, children learn to use simple frames such as *my + X* or *his + X* to indicate possession. As development progresses, these frames are merged into general constructions, such as the possessive construction. In effect, each construction emerges from a lexical gang. Sentence processing then relies on the child’s ability to combine constructions online. When two alternative constructions compete, errors appear. An example would be **say me that story*, instead of *tell me that story*. In this error, the child has treated *say* as a member of the group of verbs that forms the dative construction. In the classic theory of generative grammar, recovery from this error is supposed to trigger a learnability problem, since such errors are seldom overtly corrected and, when they are, children tend to ignore the feedback. Neural network implementations of Construction Grammar address this problem by emphasizing the direct competition between *say* and *tell* during production. The child can rely on positive data to strengthen the verb *tell* and its link to the dative construction, thereby eliminating this error without corrective feedback. In this way, models that implement competition provide solutions to the logical problem of language acquisition.

These various approaches to syntactic learning must eventually find a way of dealing with the compositional nature of syntax (Valiant, 1994). A noun phrase such as “my big dog and his ball” can be further decomposed into two segments conjoined by the “and.”

Each of the segments is further composed of a head noun and its modifiers. Our ability to recursively combine words into larger phrases stands as a major challenge to connectionist modeling. One likely solution would use predicate constructions to activate arguments that are then combined in a short-term memory buffer during sentence planning and interpretation. To build a model of this type, we need to develop a clearer mechanistic link between constructions as lexical items and constructions as controllers of the on-the-fly process of syntactic combination.

Neuronal Realism

Some researchers have criticized neural network models in the area of language acquisition for a failure to properly represent basic facts about the brain. To the degree that the backpropagation algorithm relies on reciprocal connections between units, this criticism is well founded. However, work in this area has begun to rely on models such as self-organizing feature maps, adaptive resonance, and Hebbian learning that have closer mappings to the features of neural organization. In fact, Elman (1999) has shown how the imposition of biologically realistic assumptions, such as the brain's preference for short connections, can lead to more effective language learning. Thus, this particular challenge to neural network theory may end up being more of a searchlight than a barrier.

Neural networks must also achieve a closer match to what we are now learning about functional neural circuitry. We know that auditory cortex, Broca's area, temporal word storage, and frontal attentional areas are all involved in various ways in language processing. However, we have not yet figured out exactly how these separate brain structures map onto separate aspects of lexical and syntactic processing.

Embodiment

The final challenge to neural network modeling comes from researchers who have begun to explore the ways in which the mind is grounded on the body. This relatively new line of research emphasizes the importance of findings that mental imagery makes use of the reactivation of perceptual systems to recreate physically grounded images. A convergence of work in neuroscience, psychology, and cognitive linguistics points to the view of language use not as disembodied symbol processing, but as indirectly grounded on basic mechanisms for perception and action, which themselves operate on the human body. Neural network models have just begun to deal with this new challenge. One approach emphasizes the ways in which distal learning processes can train action patterns such as speech production on the basis of their perceptual products (Plaut and Kello, 1999). Another approach, adopted by the NTL (Neural Theory of Language) group (Bailey et al., 1997) relies on the higher-order formalism of Petri nets to represent the control structure of body motions such as *pull* or *stumble*. The architecture then includes transparent methods of linking the higher-level representation to a neural network implementation.

One trend that will facilitate this work, as well as all modeling of language acquisition is the increasing availability of transcript and multimedia data from children interacting with their caretakers. The Child Language Data Exchange System (CHILDES) at <http://childes.psy.cmu.edu> now provides thousands of hours of transcripts of child language data, much of it linked to audio and some to video. In the context of the broader TalkBank Project at <http://talkbank.org>, this data is being recoded in XML format and linked to a variety of computational tools for analyzing gestural, phonological, morphological, and syntactic structure. This growing database provides increasingly rich targets for neural network modeling.

Conclusions

Neural networks have addressed many aspects of Chomsky's challenge. They have been used to develop useful models of virtually all aspects of language learning and processing. However, further challenges lie ahead. Of these, the most pressing is the need to develop methods for simulating the learning of a realistically sized lexicon of several thousand words. If this problem can be solved, it will have further positive consequences for models of syntactic development that emphasize the importance of item-based learning. It is likely that a good solution to this problem will need to rely on an improved understanding of the ways in which the brain stores and processes lexical items. An even greater challenge will be developing models that express the ways in which language processing is grounded on embodied cognition. Together, these challenges guarantee vitality in this area for years to come.

Road Map: Linguistics and Speech Processing

Related Reading: Cognitive Development; Constituency and Recursion in Language; Developmental Disorders; Language Evolution and Change; Language Evolution: The Mirror System Hypothesis

References

- Bailey, D., Feldman, J., Narayanan, S., and Lakoff, G., 1997, Modeling embodied lexical development, *Proceedings of the 19th Meeting of the Cognitive Science Society*, pp. 18–22. ♦
- Bybee, J., 1995, Regular morphology and the lexicon, *Lang. Cognit. Proc.*, 10:425–455.
- Chomsky, N., 1965, *Aspects of the Theory of Syntax*, Cambridge, MA: MIT Press.
- Elman, J., 1990, Finding structure in time, *Cognit. Sci.*, 14:179–212. ♦
- Elman, J. L., 1999, The emergence of language: A conspiracy theory, in *The Emergence of Language* (B. MacWhinney, Ed.), Mahwah, NJ: Lawrence Erlbaum Associates, pp. 1–28.
- Farkas, I., and Li, P., 2001, Modeling the development of lexicon with a growing self-organizing map, in *Processing of the Sixth Joint Conference on Information Science* (H. J. Caulfield, Ed.), New York: Association for Intelligent Machines, pp. 553–556.
- Hamker, F. H., 2001, Life-long learning cell structures—continuously learning without catastrophic interference, *Neural Networks*, 14:551–573.
- Kawamoto, A., 1994, One system or two to handle regulars and exceptions: How time-course of processing can inform this debate, in *The Reality of Linguistic Rules* (S. D. Lima, R. L. Corrigan, and G. K. Iverson, Eds.), Amsterdam: John Benjamins, pp. 389–416.
- MacDonald, M. C., Perlmutter, N. J., and Seidenberg, M. S., 1994, Lexical nature of syntactic ambiguity resolution, *Psychol. Rev.*, 101(4):676–703. ♦
- MacWhinney, B., and Leinbach, J., 1991, Implementations are not conceptualizations: Revising the verb learning model, *Cognition*, 29:121–157. ♦
- MacWhinney, B. J., Leinbach, J., Taraban, R., and McDonald, J. L., 1989, Language learning: Cues or rules? *J. Mem. Lang.*, 28:255–277. ♦
- Maratsos, M., and Chalkley, M., 1980, The internal language of children's syntax: The ontogenesis and representation of syntactic categories, in *Children's Language: Volume 2* (K. Nelson, Ed.), New York: Gardner, pp. 127–214.
- Pinker, S., 1999, *Words and Rules: The Ingredients of Language*, New York: Basic Books.
- Plaut, D. C., and Kello, C. T., 1999, The emergence of phonology from the interplay of speech comprehension and production: A distributed connectionist approach, in B. MacWhinney (Ed.), *The Emergence of Language* (B. MacWhinney, Ed.), Mahwah, NJ: Lawrence Erlbaum Associates, pp. 381–416.
- Plunkett, K., and Marchman, V., 1993, From rote learning to system building, *Cognition*, 49:21–69.
- Tomasello, M., 2000, The item-based nature of children's early syntactic development, *Trends Cognit. Sci.*, 4:156–163. ♦
- Valiant, L., 1994, *Circuits of the Mind*, Oxford, UK: Oxford University Press.

Language Evolution and Change

Morten H. Christiansen and Rick Dale

Introduction

No direct evidence remains from before the emergence of writing systems to inform theories about the evolution of language. Only as evidence is amassed from many different disciplines can theorizing about the evolution of language be sufficiently constrained to remove it from the realm of pure speculation and allow it to become an area of legitimate scientific inquiry. To go beyond existing data, rigorously controlled thought experiments can be used as crucial tests of competing theories. Computational modeling has become a valuable resource for such tests because it enables researchers to test hypotheses about specific aspects of language evolution under controlled circumstances (Cangelosi and Parisi, 2002; Turner, 2002). With the help of computational simulations, it is possible to study various processes that may have been involved in the evolution of language, as well as the biological and cultural constraints that may have shaped language into its current form (see *EVOLUTION AND LEARNING IN NEURAL NETWORKS*).

Connectionist models have played an important role in the computational modeling of language evolution. In some cases, the networks are used as simulated agents to study how social transmission via learning might give rise to the evolution of structured communication systems. In other cases, the specific properties of neural network learning are enlisted to help illuminate the constraints and processes that may have been involved in the evolution of language. This article surveys this connectionist research, starting from the emergence of early syntax and continuing to the role of social interaction and constraints on network learning in subsequent evolution of language and to linguistic change within existing languages.

Emergence of Simple Syntax

Models of language evolution focus on two primary questions: how language emerged, and how languages continue to change over time. An important feature of the first question is the emergence of syntactic communication. Cangelosi (1999) studied the evolution of simple communication systems, but with an emphasis on the emergence of associations not only between objects (meaning) and symbols (signal), but also between the symbols themselves (syntax). In particular, the aim was to demonstrate that simple syntactic relations (a verb-object rule) could evolve through a combination of communicative interactions and cross-generational learning in populations of neural networks.

In Cangelosi's simulations, populations of networks evolved based on their ability to forage in an environment consisting of a two-dimensional 100×100 array of cells. About 12% of the cells contained randomly placed mushrooms that served as food. Three types of mushrooms were edible, increasing a network's fitness if collected, whereas another three types were poisonous, decreasing the network's fitness if collected. The networks had a standard feedforward architecture with a single hidden unit layer and were trained using backpropagation (see *BACKPROPAGATION: GENERAL PRINCIPLES*). Input was represented in terms of three sets of input units encoding the location of a mushroom, the visual features of the mushroom, and words naming objects or actions. The output contained sets of units representing actions (*approach*, *avoid*, *discriminate*) and words with the latter units organized into two winner-take-all clusters (object and verb). Populations consisted of 80 networks, each with a life span of 1,000 actions. The 20 networks with the highest fitness level were selected for asexual reproduc-

tion, each producing four offspring through random mutation of 10% of its starting weights. During the first 300 generations, the populations evolved an ability to discriminate between edible and poisonous mushrooms without the use of words. In subsequent populations, parents provided teaching input for the learning of words denoting the different mushrooms (objects) and the proper action to take (verbs). The simulations were repeated with different random starting populations. Sixty-one percent of the simulations resulted in optimal vocabulary acquisition, with different "verb" symbols used with edible (*approach*) and poisonous (*avoid*) mushrooms, and different "noun" symbols used for the different types of mushrooms.

The simulations indicate how a simple noun-verb communication system can evolve in a population of networks. Because the features of a mushroom were perceived only 10% of the time, paying attention to the parental language input provided a selective advantage with respect to foraging, thus reinforcing successful linguistic performance.

Another approach to the emergence of elementary syntax has been offered by Batali (1998). He suggested that a process of negotiation between agents in a social group may have given rise to coordinated communication. Whereas Cangelosi's model involved the emergence of rudimentary verb-object syntax in a foraging environment, Batali's networks were assigned the task of mapping meaning onto a sequence of characters for the purpose of communication in a social environment. The networks in this simulation did not start out with a predetermined syntactic system. Instead, a process of negotiation across generations engendered the evolution of a syntactic system to convey common meanings.

Each agent in the simulation was a simple recurrent network (SRN; Elman, 1990), capable of processing input sequences consisting of four characters and producing an output vector representing a meaning involving a subject and a predicate. In a negotiation round, one network was chosen as a learner, and ten randomly selected teachers conveyed a meaning converted into a string of characters. The learner then processed the string produced by the teacher, and was trained using the difference between the teacher's and the learner's meaning vectors. Batali described this interaction between learners and teachers as a kind of negotiation, since each must adjust weights in accordance with its own cognitive state and that of others. At the start of the simulations the networks generated only very long strings that were unique to each meaning. After several thousand rounds of negotiation, the agents developed a more efficient and partially compositional communication system, with short sequences of letters used for particular predicates and referents. To test whether novel meanings could be encoded by the communication system, Batali omitted ten meanings, and reran the simulations. After training, networks performed well at sending and processing the omitted meaning vectors, demonstrating that the rudimentary grammar exhibited systematicity capable of accommodating a structured semantics.

Batali's model offers illuminating observations for the evolution of language. An assumption of this model was that social animals can use their own cognitive responses (in this case, translating meaning vectors into communicable signals) to predict the cognitive state of other members of their community. Batali compared this ability to one that may have arisen early in hominids and contributed to the emergence of systematic communication. Once such an elementary communication system is in place, migration patterns may have promoted dialectical variations. The next section explores how linguistic diversity might arise as a result of geographical separation between groups of communicating agents.

Linguistic Diversity

The diversity of the world's many languages has offered puzzling questions for centuries. Computational simulations allow for the investigation of factors influencing the distribution and diversity of language types. An intuitive approach, considered in the next section, is that languages assume an adaptive shape governed by various constraints in the organism and environment. Livingstone and Fyfe (1999) have proposed an alternative perspective based on simulations in which linguistic diversity arises simply as a consequence of spatial organization and imperfect language transmission in a social group.

The social group in the simulation consisted of networks with two layers of three input and output units, bidirectionally connected and randomly initialized. As in Batali's simulations, agents were given the task of mapping a meaning vector onto an external "linguistic" signal. For each generation, a learner and a teacher were randomly selected. The output of the teacher was presented to the learner, and the error between meaning vectors was used to change the learner's weights. Each successive generation had agents from the previous generation acting as teachers. The agents were spatially organized along a single dimension and communicated only with other agents within a fixed distance. By comparing agents across this spatial organization, performance akin to a dialect continuum was observed: small clusters of agents communicated readily, but as the distance among them increased, error in communication increased. When the simulation was implemented without spatial organization (i.e., each agent was equally likely to communicate with all others), the entire population quickly negotiated a global language, and diversity was lost. This model supports the position that diversity is a consequence of spatial organization and imperfect cultural transmission.

The results of Livingstone and Fyfe's as well as Batali's simulations may not rely directly on the properties of neural network learning, but rather on the processes of learning-based social transmission. However, when it comes to explaining why certain linguistic forms have become more frequent than others, the specific constraints on learning in such networks come to the fore. The next section discusses how limitations on network learning can help explain the existence of certain so-called linguistic universals.

Learning-Based Linguistic Universals

Despite the considerable diversity that can be observed across the languages of the world, it is also clear that languages share a number of relatively invariant features in the way words are put together to form sentences. Spatial organization and error in transmission cannot account for these widespread commonalities. Instead, the specific constraints on neural network learning may offer explanations for these consistent patterns in language types. As an example, we can consider heads of phrases, that is, the particular word in a phrase that determines the properties and meaning of the phrase as a whole (such as the noun *boy* in the noun-phrase *the boy with the bicycle*). Across the world's languages, there is a statistical tendency toward a basic format in which the head of a phrase consistently is placed in the same position—either first or last—with respect to the remaining clause material. English is considered to be a head-first language, meaning that the head is most frequently placed first in a phrase, as when the verb is placed before the object noun-phrase in a transitive verb phrase such as *eat curry*. In contrast, speakers of Hindi would say the equivalent of *curry eat*, because Hindi is a head-last language.

Christiansen and Devlin (1997) trained SRNs with eight input and eight output units encoding basic lexical categories (i.e., nouns, verbs, prepositions, and a possessive genitive marker) on corpora generated by 32 different grammars with differing amount of head-

order consistency. The networks were trained to predict the next lexical category in a sentence. Importantly, these networks did not have built-in linguistic biases; rather, they were biased toward the learning of complex sequential structure. Nevertheless, the SRNs were sensitive to the amount of head-order inconsistency found in the grammars, such that there was a strong correlation between the degree of head-order consistency in a given grammar and the degree to which the network had learned to master the grammatical regularities underlying that grammar. The higher the inconsistency, the more erroneous the final network performance was. The sequential biases of the networks made the corpora generated by consistent grammars considerably easier to acquire than the corpora generated by inconsistent grammars. Christiansen and Devlin further collected frequency data concerning the specific syntactic constructions used in the simulations. They found that languages incorporating fragments that the networks found hard to learn tended to be less frequent than languages the network learned more easily. This suggests that constraints on basic word order may derive from nonlinguistic constraints on the learning and processing of complex sequential structure. Grammatical constructions incorporating a high degree of head-order inconsistency may simply be too hard to learn, and would therefore tend to disappear.

More recently, Van Everbroeck (1999) presented network simulations in a similar vein in support of an explanation for language-type frequencies based on processing constraints. He trained recurrent networks (a variation on the SRN) to produce the correct grammatical role assignments for noun-verb-noun sentences that were presented one word at a time. The networks had 26 input units, providing distributed representations of nouns and verbs as well as encodings of case markers, and 48 output units, encoding the distributed noun-verb representation according to grammatical role. Forty-two different language types were used to represent cross-linguistic variation in three dimensions: word order (e.g., subject-verb-object), and noun and verb inflection. The results of the simulations coincided with many observed trends in the distribution of the world's languages. Subject-first languages, both of which make up the majority of language types (51% and 23%, respectively), were easily processed by the networks. Object-first languages, on the other hand, were not well processed and have very low frequency among the world's languages (object-verb-subject: 0.75%; object-subject-verb: 0.25%). Van Everbroeck argued that these results were a predictable product of network processing constraints. Not all results, however, were directly proportional to actual language-type frequencies. For example, verb-subject-object languages account for only 10% of the world's language types, but the model's performance on these exceeded performance on the more frequent subject-first languages. Van Everbroeck suggested that making the simulations more sophisticated (incorporating semantics or other aspects of language) might allow network performance to better approach observed frequencies. Together, the simulations by Van Everbroeck and by Christiansen and Devlin provide preliminary support for a connection between learnability and frequency in the world's languages based on the learning and processing properties of connectionist networks. The next section discusses additional simulations that show how similar network properties may also help explain linguistic change within a particular language.

Linguistic Change

The English system of verb inflection has changed considerably over the past 1,100 years. Simulations by Hare and Elman (1995) demonstrate how neural network learning and processing constraints may help explain the observed pattern of change. The morphological system of Old English (ca. 870) was quite complex, involving at least ten different classes of verb inflection (with a

minimum of six of these being “strong”). The simulations involved several “generations” of neural networks, each of which received as input the output generated by a trained net from the previous generation. The first net was trained on data representative of the verb classes from Old English. However, training was stopped before learning could reach optimal performance. This reflected the causal role of imperfect transmission in language change. The imperfect output of the first net was used as input for a second generation net, for which training was also halted before learning reached asymptote. Output from the second net was then given as input to a third net, and so on, until seven generations were trained. This training regime led to a gradual change in the morphological system. These changes can be explained by verb frequency in the training corpus, and internal phonological consistency (i.e., distance in phonological space between prototypes). The results revealed that membership in small classes, inconsistent phonological characteristics, and low frequency all contributed to rapid morphological change. As the morphological system changed through generations in these simulations, the pattern of results closely resembled the historical change in English verb inflection from a complex past tense system to a dominant “regular” class and small classes of “irregular” verbs.

Discussion

This article has surveyed the use of neural networks for the modeling of language evolution and change. The results discussed here are encouraging, even though neural network modeling of language evolution is very much in its infancy. However, it is also clear that the current models suffer from obvious shortcomings. Most of them are highly simple and do not fully capture the vast complexity of the issues at hand. For example, the models of the emergence of verb-object syntax and linguistic diversity incorporated very simple relationships between meaning and form. Moreover, although the simulations of the influence of processing constraints on the shape of language involved relatively complex grammars, they did not include any relationship between the language system and the world. Nevertheless, these models demonstrate the potential for exploring the evolution of language from a computational perspective.

Both connectionist and nonconnectionist models (e.g., Nowak and Komarova, 2001) have been used to provide important thought experiments in support of theories of language evolution. Connectionist models have become prominent in such modeling, both for their ability to simulate social interaction in populations and for their demonstrations of how learning constraints imposed on communication systems can engender many of the linguistic properties we observe today. Together, the models point to an important role

for cultural transmission in the origin and evolution of language. This perspective receives further support from neuroscientific considerations, suggesting a picture of language and brain that argues for their co-evolution (e.g., Deacon, 1997). The studies discussed here highlight the promise of neural network approaches to these issues. Future studies will likely seek to overcome current shortcomings and move toward more sophisticated simulations of the origin and evolution of language.

Road Maps: Linguistics and Speech Processing; Neuroethology and Evolution

Background: Language Processing

Related Reading: Constituency and Recursion in Language; Evolution and Learning in Neural Networks; Language Evolution, The Mirror System Hypothesis

References

- Batali, J., 1998, Computational simulations of the emergence of grammar, in *Approaches to the Evolution of Language: Social and Cognitive Bases* (J. R. Hurford, M. Studdert-Kennedy, and C. Knight, Eds.), Cambridge, Engl.: Cambridge University Press, pp. 405–426.
- Cangelosi, A., 1999, Modeling the evolution of communication: From stimulus associations to grounded symbolic associations, in *Advances in Artificial Life: Proceedings of the ECAL99 European Conference on Artificial Life* (D. Floreano, J. Nicoud, and F. Mondada, Eds.), Berlin: Springer-Verlag, pp. 654–663.
- Cangelosi, A., and Parisi, D., 2002, Computer simulation: A new scientific approach to the study of language evolution, in *Simulating Language Evolution* (A. Cangelosi and D. Parisi, Eds.), London: Springer-Verlag, pp. 3–28. ♦
- Christiansen, M. H., and Devlin, J. T., 1997, Recursive inconsistencies are hard to learn: A connectionist perspective on universal word order correlations, in *Proceedings of the 19th Annual Cognitive Science Society Conference*, Mahwah, NJ: Erlbaum, pp. 113–118.
- Deacon, T., 1997, *The Symbolic Species: The Co-evolution of Language and the Brain*, New York: Norton. ♦
- Elman, J. L., 1990, Finding structure in time, *Cogn. Sci.*, 14:179–211.
- Harc, M., and Elman, J. L., 1995, Learning and morphological change, *Cognition*, 56:61–98.
- Nowak, M. A., and Komarova, N. L., 2001, Towards an evolutionary theory of language, *Trends Cognitive Sci.*, 5:288–295.
- Livingstone, D., and Fyfe, C., 1999, Modelling the evolution of linguistic diversity, in *Advances in Artificial Life: Proceedings of the ECAL99 European Conference on Artificial Life* (D. Floreano, J. Nicoud, and F. Mondada, Eds.), Berlin: Springer-Verlag, pp. 704–708.
- Turner, H., 2002, An introduction to methods for simulating the evolution of language, in *Simulating Language Evolution* (A. Cangelosi and D. Parisi, Eds.), London: Springer-Verlag, pp. 29–50. ♦
- Van Everbroeck, E., 1999, Language type frequency and learnability: A connectionist appraisal, in *Proceedings of the 21st Annual Cognitive Science Society Conference*, Mahwah, NJ: Erlbaum, pp. 755–760.

Language Evolution: The Mirror System Hypothesis

Michael A. Arbib

Introduction

What is the evolutionary path leading to language in humans, and what are the relevant data on brain mechanisms? Since the fossil record offers no trace of brain structure beyond clues from ancient skulls on brain size and perhaps some fissures of the brain, the answers to these questions are varied and controversial (Wilkins and Wakefield, 1995). The present article emphasizes the *mirror system hypothesis* (MSH). The “mirror system” for grasping in

monkey, which contains *mirror neurons* that are active both when the monkey executes a specific hand action and when it observes a human or other monkey carrying out a similar action, is the homologue of Broca’s area, a crucial speech area in humans. MSH asserts that the matching of neural code for execution and observation of hand movements is present in the common ancestor of monkey and human and is the precursor of the crucial language property of *parity*, namely, that an utterance usually carries similar meaning for speaker and hearer (using these terms neutrally for

spoken and signed languages). This provides a neurobiological “missing link” for the hypothesis that communication based on manual gesture preceded speech in language evolution (e.g., Stokoe, 2001).

Where the present article focuses on brain mechanisms for vision, action, and language, the article LANGUAGE EVOLUTION AND CHANGE, focuses on the use of connectionist modeling to test hypotheses (usually more psychological than neurological in nature) about specific aspects of language evolution.

What Does the Biology of the Brain Provide?

Chomsky (e.g., 1975) has argued that since children acquire language rapidly despite the “poverty of the stimulus,” the basic structures of language must be innate, forming a universal grammar encoded in the human genome. For example, universal grammar would encode the knowledge that sentences in a human language could be ordered as subject-verb-object, subject-object-verb, or one of a few other options, so that the child simply needs to hear a few sentences of his first language to “set the parameter” for the preferred order of that language. Against this, others have argued that the child has both a rich set of language stimuli linked to action and perception and powerful learning mechanisms, so that a child can indeed learn from its social interactions aspects of syntax which Chomsky would see as genetically prespecified (see LANGUAGE ACQUISITION). There is no argument against the view that human evolution yielded genetic specification of some of the structures which *support* language. Humans have hands, a larynx, and facial mobility suited for generating gestures that can be used in language, and the brain mechanisms needed to produce and perceive rapidly generated sequences of such gestures (Lieberman, 1991). In this sense, the human brain and body is *language ready*. The term *language readiness* was, I believe, my coinage (Arbib, 2002), but of course I was not the first to assert that the biological evolution that took us to ancestors with a language-ready brain (by some other name) had to be followed by a cultural evolution that took us from rudimentary manual-vocal communication to full language capability.

Although it is a matter of ongoing debate to delimit what constitutes language readiness and how it differs from language, we offer here a tentative list of criteria as a basis for future research.

Language Readiness

The first three properties support communication system without necessarily yielding language, while the last four properties are more general:

Symbolization: The ability to associate an arbitrary symbol with a class of events, objects, or actions, etc. At first, these symbols may not have been words in the modern sense, nor need they have been vocalized.

Intentionality: Communication is *intended* by the utterer to have a particular effect on the recipient.

Parity (mirror property): What counts for the speaker (or producer) must count for the listener (or receiver).

Hierarchical structuring: Production and recognition of components with subparts. This relates to basic mechanisms of action-oriented perception with no necessary link to the ability to communicate about these components and their relationships.

Temporal ordering: Temporal activity coding these hierarchical structures.

Beyond the here-and-now: The ability to recall past events or imagine future ones.

Paedomorphy and sociality: The prolonged immaturity of the infant and the prolonged caregiving of adults combine to create conditions for complex social learning.

Deacon (1997) makes symbolization central to his account of the co-evolution of language and the human brain. He stresses the convergent homology that allows us to learn from relations between the brains of monkeys and humans. His more recent writings stress the role of self-organization as the child’s brain adapts to the cultural environment in which the child develops. Where Deacon places most emphasis on the enlargement of frontal cortex, we place more emphasis on the differential development of specific subsystems that support language readiness. Interestingly, Semendeferi et al. (2002) argue that magnetic resonance imaging shows that human frontal cortices are not disproportionately large in comparison to those of the great apes. They thus suggest that the special cognitive abilities of humans may be due to differences in individual cortical areas and to a richer interconnectivity, rather than to an increase in the overall relative size of the frontal lobe during human evolution.

Language

We now turn to criteria for language, which we here hypothesize as what cultural evolution and learning add to the brain’s capabilities for language readiness. Note that nothing in this list rests on the medium of exchange of the language, and that the list applies both to spoken language and to sign language.

Symbolization: The symbols become words in the modern sense, interchangeable and composable in the expression of meaning.

Syntax and semantics: The matching of syntactic to semantic structures co-evolves with the fractionation of utterances. This includes the ability to build utterances recursively (see CONSTITUENCY AND RECURSION IN LANGUAGE).

Beyond the here-and-now: Verb tenses (or alternative syntactic constructs) arise to express recall of past events and imagination of future ones.

Learnability: To qualify as a human language, it must contain a *significant subset* of symbolic structures learnable by most human children (but children do not master a language completely by 5 or 7 years of age).

Bickerton (1995) characterizes *protolanguage* as a form of communication whose users can only string together a small handful of words at a time, may leave out words arbitrarily, may often depart from customary word order, cannot form any complex structures, and use only a tiny fraction of the inflections and “grammatical words” that make up 50% of true language utterances. Bickerton then makes two distinct claims: (1) that the productions of apes who have been taught to use signs, early-stage pidgin languages, and the speech of children under 2 share enough properties in common that they can all be characterized as examples of a common entity, “protolanguage”; and (2) that this same entity, protolanguage, characterizes the “prelanguage” of early hominids. The counterhypothesis advanced here is that the prelanguage of early hominids was not a protolanguage in Bickerton’s sense, but rather was made up of “one-word utterances” in which the “words” were *holophrastic*, i.e., more like today’s complete phrases or sentences, with modern-sense words and syntax later co-evolving culturally.

Neurobiological Foundations

With this, we turn to the neurobiology of the monkey to ground claims as to the brain of the common ancestor of monkeys and

humans of perhaps 20 million years ago, and hypotheses on how such brains changed to become language ready.

Brain Mechanisms for Grasping

Parietal area AIP and ventral premotor area F5 anchor the cortical circuit in monkey that transforms visual information on intrinsic properties of an object into hand movements for grasping it (see GRASPING MOVEMENTS: VISUOMOTOR TRANSFORMATIONS). Discharge in most F5 neurons correlates with an action rather than with the individual movements that form it, so that one may relate F5 neurons to various motor schemas (see SCHEMA THEORY) corresponding to the action associated with their discharge.

The FARS model (Fagg and Arbib, 1998) provides a computational account of the system centered on the AIP → F5 pathway (Figure 1): AIP cells encode “affordances”—visual features of the object relevant to action—for grasping and send (neural codes for) these on to area F5, which selects one of these for action. Inferotemporal cortex (IT) and prefrontal cortex (PFC) modulate F5’s selection of an affordance. However, the dorsal stream via AIP does not know “what” the object is, it can only see the object as a set of possible affordances. The ventral stream (from primary visual cortex to IT), by contrast, is able to recognize what the object is (see DISSOCIATIONS BETWEEN VISUAL PROCESSING MODES). This information is passed to PFC, which can then, on the basis of the current goals of the organism and the recognition of the nature of the object, bias F5 to choose the affordance appropriate to the task at hand. (See GRASPING MOVEMENTS: VISUOMOTOR TRANSFORMATIONS for recent neuroanatomical data suggesting that PFC may act on action selection at the level of parietal cortex rather than premotor cortex.)

Figure 1 gives only a partial view of the FARS model, which also provides mechanisms for sequencing actions. It segregates the F5 circuitry that encodes unit actions from the circuitry encoding a sequence, possibly the part of the supplementary motor area called pre-SMA (Rizzolatti, Luppino, and Matelli, 1998). The administration of the sequence (inhibiting extraneous actions, while priming imminent actions) is then carried out by the basal ganglia (see BASAL GANGLIA and SEQUENCE LEARNING).

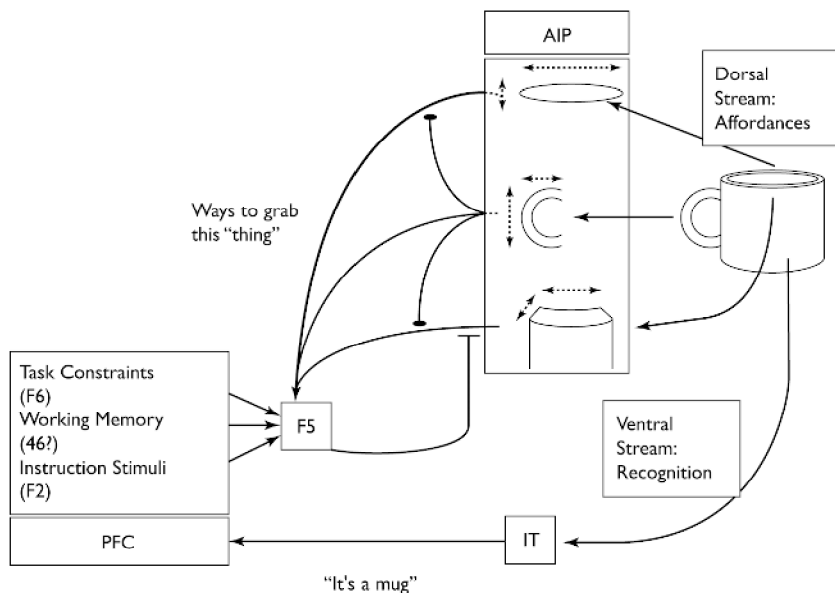


Figure 1. The role of IT (inferotemporal cortex) and PFC (prefrontal cortex) in modulating F5’s selection of an affordance from the repertoire forwarded by AIP.

The Mirror System for Grasping

Further study revealed a class of F5 neurons that discharged not only when the monkey grasped or manipulated objects, but also when the monkey observed the experimenter make a gesture similar to the one that, when actively performed by the monkey, involved activity of the neuron. Neurons with this property are called *mirror neurons* (Gallese et al., 1996). Mirror neurons respond only to an interaction between the agent and the object of an action. The simple presentation of objects, even when held by hand, does not evoke the neuron discharge.

Not all F5 neurons respond to action observation. We thus distinguish mirror neurons from *canonical neurons* in F5, which are active only when the monkey itself performs the relevant actions. Mirror neurons receive input from the PF region of parietal cortex encoding observations of arm and hand movements. This is in contrast to the canonical F5 neurons, which receive object-related input from AIP. It is the canonical neurons, with their input from AIP, that are modeled in the FARS model.

Bridging from Action to Language

A Mirror System for Grasping in Humans

The notion that a mirror system might exist in humans was tested by PET experiments, which showed that grasp observation significantly activated the superior temporal sulcus (STS), the inferior parietal lobule, and the inferior frontal gyrus (area 45). All activations were in the left hemisphere. The last area is of special interest: areas 44 and 45 in the left hemisphere of the human constitute Broca’s area. F5 in monkey is generally considered (see analysis by Matelli in Rizzolatti and Arbib, 1998) to be the homologue of Broca’s area in humans; i.e., it can be argued that these areas of monkey and human brain are related to the same region of the common ancestor. Thus, the cortical areas active during action observation in humans and monkeys correspond very well, indicating that there is a fundamental primate mechanism for action recognition: we argue that individuals recognize actions made by others because the neural pattern elicited in their premotor areas during action observation is similar to a part of that internally generated to produce a similar action. Note, however, that “understand-

ing” involves the cooperation of many brain systems and cannot be reduced to just the activity in a subset of F5 neurons.

Primate Vocalization

Monkeys exhibit a primate call system (a limited set of species-specific calls) and an orofacial (mouth and hand) gesture system (a limited set of gestures expressive of emotion and related social indicators). This communication system is *closed* in the sense that it is restricted to a specific repertoire. This is to be contrasted with the open nature of human languages. (Although each human language is open as to, e.g., nouns and verbs, it is [almost] closed with respect to prepositions and grammatical markers.) Strikingly, the neural substrate for primate calls is in a region of cingulate cortex distinct from F5, which we have seen to be the monkey homologue of Broca’s area in the human. One challenge, then, is to understand why it is F5, rather than the cingulate area already involved in monkey vocalization, that is homologous to the human’s frontal substrate for language. Note that the claim is not that Broca’s area is genetically preprogrammed for language, but rather that the development of a human child in a language community normally adapts this brain region to play a crucial role in language performance.

The Mirror System Hypothesis

What turns a movement into an action is that it is associated with a goal, so that initiation of the movement is accompanied by the creation of an expectation that the goal will be met. We distinguish “pragmatic action,” in which the hands are used to interact physically with objects or the bodies of other creatures, and “gestures” (both manual and vocal), whose purpose is communication. Our assumption is that monkeys use hand movements only for pragmatic actions. The mirror system allows other monkeys to understand these actions and act on the basis of this understanding. Similarly, the monkey’s orofacial gestures register emotional state, and primate vocalizations can also communicate something of the current situation of the monkey.

Stokoe (2001) provides a recent summary of the argument, rooted in the analysis of sign language, that communication based on manual gesture played a crucial role in human language evolution, preceding communication by speech. In this regard, we stress that the “openness” or “generativity” that some see as the hallmark of language is present in manual behavior, which can thus supply the evolutionary substrate for its appearance in language. With our understanding that the mirror system in monkeys is the homologue of Broca’s area in humans, we can now appreciate the mirror system hypothesis.

The mirror system hypothesis: Language evolved from a basic mechanism *not* originally related to communication: the *mirror system for grasping*, with its capacity to generate *and* recognize a set of actions. More specifically, Broca’s area in the human contains a mirror system for grasping that is homologous to the F5 mirror system of monkey, and this provides the evolutionary basis for *language parity*; i.e., an utterance means roughly the same for both speaker and hearer.

However, having a mirror system is not equivalent to having language. Monkeys have mirror systems but do not have language, and we expect that many species have mirror systems for varied socially relevant behaviors.

Simple and Complex Imitation Systems for Grasping

It is unclear whether the mirror system for grasping is sufficient for the copying of actions. It is one thing to recognize an action

using the mirror system and another thing to use that representation as a basis for repeating the action. In any case, the ability to copy *single* actions is just the first step toward imitation, since imitation involves “parsing” a complex movement into more or less familiar pieces and then performing the corresponding composite of (variations on) familiar actions. Myowa-Yamakoshi and Matsuzawa (1999) observed that chimpanzees typically took 12 trials to learn to “imitate” a behavior, and in doing so paid more attention to where the manipulated object was being directed rather than to the actual movements of the demonstrator. This may involve using one or both hands to bring two objects into relationship or to bring an object into relationship with the body. Thus the form of imitation reported for chimpanzees is a long and laborious process compared to the rapidity with which humans can acquire novel sequences. I have called this the contrast between “simple” imitation and “complex” imitation (Arbib, 2002) and assert that monkeys have neither, chimpanzees have simple imitation, and humans have complex imitation (not all primatologists accept this distinction; see IMITATION for further discussion).

If we assume (1) that the common ancestor of monkeys and apes had no greater imitative ability than present-day monkeys and (2) that the ability for simple imitation shared by chimps and humans was also possessed by their common ancestor, but that (3) only humans possess a talent for “complex” imitation, then we have established a case for the claim that brain mechanisms for simple imitation developed in the 15 million-year evolution from the common ancestor of monkeys and apes to the common ancestor of apes and humans, and that a complex imitation system—acquiring (longer) novel sequences of more abstract actions in a single trial—developed in the 5 million-year evolution from the common ancestor of apes and humans along the hominid line that led, in particular, to *Homo sapiens*. The argument, then, is that extension of the mirror system from recognition of single actions to imitation of compound actions was one of the key innovations in the brains of hominids relevant to language readiness.

A Manual-Based Communication System

Given a creature with the ability for complex imitation, how might further hominid evolution have yielded a manual-based communication system that could lead to the further evolution of brain and body mechanisms supporting language readiness? Our hypothetical sequence leading to manual gesture and beyond is the following:

1. Pragmatic action directed toward a goal object
2. Imitation of such actions
3. Pantomime in which similar actions are produced in the absence of any goal object

In terms of observable movements, imitation of an action and pantomime of an action may appear the same. However, imitation is the generic attempt to reproduce movements performed by another, whether to master a skill or simply as part of a social interaction. By contrast, pantomime is essentially communicative, performed with the intention of getting the observer to think of a specific action or event. Thus, even though the movements may be similar, the actions (movement + goal) are very different.

4. Abstract gestures divorced from their pragmatic origins (if such existed). In pantomime it might be hard to distinguish a grasping movement signifying “grasping” from one meaning “a [graspable] raisin,” thus providing an “incentive” for coming up with an arbitrary gesture to distinguish the two meanings.

This suggests that *arbitrary* symbols emerged when the communicative capacities of pantomiming were exhausted. This can be

illustrated with modern American Sign Language, in which, for example, noun/verb pairs may be differentiated by movement. For example (Stokoe, 2001), the AIRPLANE/FLY pair of signs uses the same handshape, but the noun has short, repeated movements, whereas the verb has a single prolonged movement.

We thus distinguish *complementary roles for imitation* in the posited evolution of manual-based communication: extending imitation to pantomime to provide “natural” gestures that may convey a situation to the observer; and extending the mirror system from the grasping repertoire to mediate imitation of gestures that provide “conventionalized” symbols that can reduce ambiguity and extend the semantic range.

My current hypothesis is that stages (2) and (3) and a rudimentary (presyntactic) form of (stage 4) were present in prehuman hominids, but that the explosive development of linked symbols that we know as language depended on cultural evolution, well after biological evolution had formed modern *Homo sapiens*.

The Path to Protospeech Is Indirect

Earlier we noted that the neural substrate for primate calls is in a region of cingulate cortex distinct from F5, which latter is the monkey homologue of Broca’s area in the human. Rizzolatti and Arbib (1998) suggest two evolutionary stages:

1. A *distinct* manuobrachial (hand-arm) communication system evolved (as just described) to complement the primate calls/orofacial communication system. At this stage, the “speech” area (i.e., the area of the hominid brain presumably homologous to monkey F5) mediated only orofacial and manuobrachial communication.
2. The manual-orofacial symbolic system then “recruited” vocalization. The association of vocalization with manual gestures allowed it to assume a more open referential character, yielding “protospeech” (but not full-blown spoken language). This yields the MSH explanation of why F5, rather than the primate call area, provides the evolutionary substrate for language readiness.

However, language and vocalization systems are nonetheless linked. Lesions centered in the anterior cingulate cortex and supplementary motor areas of the brain can also cause mutism in humans, similar to the effects produced in muting monkey vocalizations. Conversely, a patient with a Broca’s area lesion may nonetheless swear when provoked. But note that “emitting an imprecation” is more like a monkey vocalization than like the syntactically structured use of language. Lieberman (1991) suggests that the primate call made by an infant separated from its mother not only survives in the human infant, but in humans develops into the breath group, i.e., the pattern of breathing in and breathing out that is shaped to provide the contour for each continuous sequence of a spoken utterance. This suggests that the evolution of speech yielded the pathways for cooperative computation between cingulate cortex and Broca’s area, with cingulate cortex involved in breath groups and emotional shading and Broca’s area involved in providing (in concert with, e.g., the basal ganglia) motor control for rapid production and interweaving of elements of an utterance.

From Protospeech to Language

The cultural evolution of *Homo sapiens* may have involved an increased ability to name actions and objects to create a rapidly growing set of verb-argument structures, and the ability to compound those structures in diverse ways. Earlier I suggested that many grammatical structures would have been “postbiological” in their origin (Arbib, 2002). We might then see as ingenious human

discoveries that the one word *ripe* halves the number of fruit names to be learned, or that separating action names from object names requires one to learn only $m + n$ words (m nouns and n verbs) to be able to form $m*n*m$ of the most basic utterances.

The spread of these innovations resided in the ability of other humans not only to imitate the new actions and compounds of actions demonstrated by the innovators, but also to do so in a way that related increasingly general classes of symbolic behavior to the classes, events, behaviors, and relationships they were to represent. Indeed, consideration of the spatial basis for “prepositions” may help show how visuomotor coordination underlies some aspects of language, while variations in the use of corresponding prepositions, even in English and Spanish, show how the basic, functionally grounded semantic-syntactic correspondences have been overlaid by a multitude of later innovations and borrowings.

Toward a Mirror System–Based Neurolinguistics

The monkey needs many brain regions for the mirror system for grasping. We will need many more brain regions for a full neurolinguistic model that extends the linkages far beyond the F5 \approx Broca’s area homology. To set the stage for the future development of such a model, we briefly link our view of AIP and F5 in monkey to data on human abilities. Studies of the visual system of monkey led to the distinction between IT mechanisms for object recognition (“what”) and posterior parietal (PP) mechanisms for localizing objects (“where”). Others extended this to a dichotomy between human “what” (IT) and “how” (PP): a patient with damage to the IT pathway could grasp and orient objects appropriately for manipulating them but could not report, either verbally or by pantomime, how big an object was or what the orientation of a slot was; another patient with damage to the PP pathway could communicate the size of a cylinder but not preshape appropriately (see DISSOCIATIONS BETWEEN VISUAL PROCESSING MODES).

Let us now try to reconcile these observations with our mirror system–based approach to language. Our evolutionary theory suggests a progression from action to action recognition to language, as follows:

- | | |
|---|--------------------|
| 1. Object \rightarrow AIP \rightarrow F5 _{canonical} | pragmatics |
| 2. Action \rightarrow PF \rightarrow F5 _{mirror} | action recognition |
| 3. Scene \rightarrow Wernicke’s \rightarrow Broca’s | utterance |

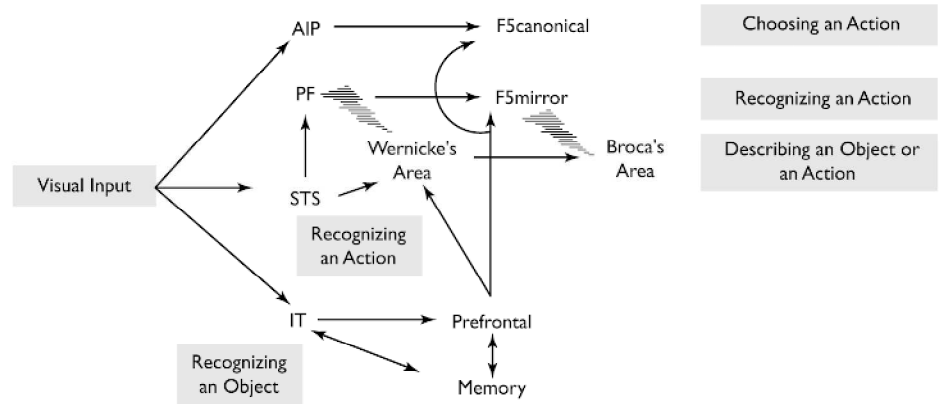
The “zero-order” model of the above PP/IT data is:

4. Parietal “affordances” \rightarrow preshape
5. IT “perception of object” \rightarrow pantomime or verbally describe size

However, step 5 implies that one cannot directly pantomime or verbalize a parietal affordance; one needs the “unified view of the object” (IT) before one can communicate attributes. The problem with this is that the “language” path as shown in step 3 is independent of the IT system. To resolve this paradox, we note the experiments of Bridgeman and his colleagues (DISSOCIATIONS BETWEEN VISUAL PROCESSING MODES). When an observer sees a target in one of several possible positions and a frame either centered before the observer or deviated left or right, verbal judgments of the target position are altered by the background frame’s position, but “jabbing” at the target never misses. The point is that communication must be based on the size estimate generated by IT, presumably for overall planning of movement, and not on that generated by PP, which provides precise parameters for motor control.

Given these data, we may now recall (Figure 1) that although AIP extracts a set of affordances, it is IT and PFC that are crucial to F5’s selection of the affordance to execute, and then offer the scheme shown in Figure 2 (from Arbib in Cangelosi and Parisi,

Figure 2. An early pass on a mirror system-based neurolinguistics. (From Arbib, 2002.)



2001). Here we emphasize the crucial role of IT-mediated functioning of PFC in the activity of Broca's area. This is the merest of sketches. For example, we do not tease apart the roles of different subdivisions of PFC in modulating $F5_{canonical}$, $F5_{mirror}$, and Broca's area. However, the crucial suggestion is that, just as $F5_{mirror}$ receives its parietal input from PF rather than AIP, so Broca's area receives its size data as well as object identity data from IT via PFC, rather than via a side path from AIP. This is just the beginning.

Discussion

The approach taken here has rich implications for the study of human evolution and NEUROLINGUISTICS (q.v.).

1. It is a mistake to assume that features common to modern-day languages must be biological in nature. Work on brain mechanisms of language must seek to distinguish the biological givens of language readiness from the cultural extensions that define the world's languages today.
2. We must be open to multiple hypotheses about the nature of hominid "prelanguage," seeking to evaluate the claims of Bickerton's protolanguage hypothesis against the claims for fractionation and imitation-based symbolization advanced here.
3. MSH offers a clear explanation for why Broca's area is homologous to an area for grasping in the human brain rather than to the cingulate area involved in primate vocalizations.
4. MSH also explains why language is multimodal, so that a deaf child can acquire sign language as readily as a hearing child acquires speech.

Even the canonical system (the FARS model) involves multiple regions in the monkey's brain (including AIP, $F5$, IT, pre-SMA, and basal ganglia), and the mirror system involves many more (including STS and PF). As we seek to understand the extensions of these that mediate complex imitation and undergird language readiness, we see that there is far more to the human brain's unique capability to master language than "F5 becomes Broca's area to provide parity." Building on Figure 2, or on other hypotheses grounded in the comparative study of primate brains, to develop a rich model of language readiness—and developing an integrated view of syntax and semantics to build upon it—will require both analysis of neurological data and subtle modeling that links neurolinguistics to the basic neural mechanisms for the recognition of the interactions of actors and objects, and for the elaboration of suitable motor plans for interacting with the environment so perceived. A further challenge is to explore the extent to which MSH

constrains the structure of modern languages, despite the great variations wrought by cultural evolution. One such example (Arbib and Jean-Roger Vergnaud, in prep.) is the assertion that sentences are canonically structured as

$$[_s NP_s [_{VP} V [_{VP} \times NP_o]]]$$

v-V being the analysis of the verb in the sentence, because the canonical system of $F5$ binding action to object underlies the merging of the V-component with the object NP_o , while the mirror system of $F5$ binding agent to action underlies the merging of the subject NP_s with the complex v-VP.

Road Maps: Linguistics and Speech Processing; Neuroethology and Evolution

Related Reading: Evolution of Artificial Neural Networks; Evolution of the Ancestral Vertebrate Brain; Grasping Movements: Visuomotor Transformations; Imitation; Language Evolution and Change; Neuroethology, Computational

References

- Arbib, M. A., 2002, The mirror system, imitation, and the evolution of language, in *Imitation in Animals and Artifacts* (C. Nehaniv and K. Dautenhahn, Eds.), Cambridge, MA: MIT Press, pp. 229–280. ♦
- Bickerton, D., 1995, *Language and Human Behavior*, Seattle: University of Washington Press.
- Cangelosi, A., and Parisi, D., Eds., 2001, *Simulating the Evolution of Language*, London: Springer-Verlag.
- Chomsky, N., 1975, *Reflections on Language*, New York: Pantheon.
- Deacon, T. W., 1997, *The Symbolic Species: The Co-evolution of Language and the Brain*, New York: Norton.
- Fagg, A. H., and Arbib, M. A., 1998, Modeling parietal-premotor interactions in primate control of grasping, *Neural Netw.*, 11:1277–1303.
- Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G., 1996, Action recognition in the premotor cortex, *Brain*, 119:593–609.
- Lieberman, P., 1991, *Uniquely Human: The Evolution of Speech, Thought, and Selfless Behavior*, Cambridge, MA: Harvard University Press.
- Myowa-Yamakoshi, M., and Matsuzawa, T., 1999, Factors influencing imitation of manipulatory actions in chimpanzees (*Pan troglodytes*), *J. Comp. Psychol.*, 113:128–136.
- Rizzolatti, G., and Arbib, M. A., 1998, Language within our grasp, *Trends Neurosci.*, 21:188–194. ♦
- Rizzolatti, G., Luppino, G., and Matelli, M., 1998, The organization of the cortical motor system: New concepts, *Electroencephalogr. Clin. Neurophysiol.*, 106:283–296.
- Semendeferi, K., Lu, A., Schenker, N., and Damasio, H., 2002, Humans and great apes share a large frontal cortex, *Nature Neurosci.*, 5:272–276.
- Stokoe, W. C., 2001, *Language in Hand: Why Sign Came Before Speech*, Washington, DC: Gallaudet University Press. ♦
- Wilkins, W. K., and Wakefield, J., 1995, Brain evolution and neurolinguistic preconditions, *Behav. Brain Sci.*, 18:161–226.

Language Processing

Richard Shillcock

Introduction

Language has provided some of the most important opportunities and challenges for connectionist cognitive modeling. There is a wealth of human behavioral data, from cognitive psychology experiments, from observations of children acquiring language, and from studies of developmental or acquired language disorders. There are also comprehensive insights from formal domains, such as phonology and syntax. Since McClelland and Rumelhart's (1981) Interactive Activation Model of visual word recognition, connectionist modeling has been applied to most areas of language processing. For instance, neural network architectures have been employed to discover syllable structure from waveforms, to manage constraint satisfaction with different categories of linguistic information in parsing, and to simulate data from historical linguistics and language typology. Even in formal syntax, the connectionist paradigm shift has provoked notions such as the graded application of rules (see OPTIMALITY THEORY IN LINGUISTICS). In the following paragraphs we review a number of models that capture detailed behavioral data and simulate the brain's discovery and manipulation of structure in spoken and written language.

This modeling was conducted against the backdrop, within psychology and cognitive science, of two related concerns: modularity and top-down feedback. Researchers have asked whether the functional architecture of cognition is modular: Are certain kinds of processing autonomous and encapsulated from other kinds of information, or do different types of information freely interact? The paradigm example has been the claimed autonomy of syntactic processing. Researchers have also asked whether genuine top-down feedback occurs: Is more sophisticated, higher-order, contextual information ever used to enhance the processing of lower-order, more peripheral representations, or is cognition fundamentally "bottom-up"? The paradigm example here has been the relationship between a word and its constituent letters or speech segments (see SPEECH PROCESSING: PSYCHOLINGUISTICS).

Many classical box-and-arrow cognitive models have become very elaborate in an attempt to accommodate additional human data, and have often been limited in their ability to generate interesting emergent behavior that might be tested against human subjects. Finally, there has been a fierce debate over the nature of representation in the brain, the status of rules, and the extent of any specific genetic endowment underlying human language learning. Despite the insights gained from the formal study of language, the formal approach has allowed researchers to say relatively little about learning and development. Thus, the "connectionist program" for understanding language has concentrated on the process of *change*, exploring topics such as language development (see LANGUAGE ACQUISITION; PAST TENSE LEARNING); language breakdown (see LESIONED NETWORKS AS MODELS OF NEUROPSYCHOLOGICAL DEFICITS; NEUROLINGUISTICS); the dynamics of representation in complex systems, which themselves may be receiving changing input (see SPEECH PROCESSING: PSYCHOLINGUISTICS); and, more recently, the evolution of language (see LANGUAGE EVOLUTION AND CHANGE). Despite the fact that connectionist models of language have often been built to cover restricted domains, such as learning the past tense, a common goal has been to show the *emergence* of complex linguistic behavior from more basic foundations, which may not themselves be specifically linguistic in nature.

Lexical Processing

Reading Single Words

Models of single-word reading and pronunciation have perhaps been the most compelling of any connectionist models of cognition. McClelland and Rumelhart's Interactive Activation Model (IAM) is an enduring example of a large-scale model producing illuminating and testable emergent behaviors through the management of very large numbers of interactions. It identifies four-letter words from an input of visual features in the four-letter positions. This input level is connected to a level containing a representation of each letter in each position, which is connected to a level containing a representation of each word. The model was handwired so that excitatory connections between levels cause features to activate the relevant letter representations, which then activate all of the words in which they occur. Each activated word node sends supportive activation to its constituent letters. Inhibitory connections between and within levels suppress inconsistent representations. This localist model accurately simulated human data concerning the perception of letters in briefly presented words, such as the apparent top-down support of "gangs" of similar words for shared letters in particular positions. The model has been incorporated as a proper part of Coltheart and colleagues' (1993) symbolic/connectionist hybrid model of pronunciation.

Recognizing Spoken Words

Top-down feedback similar to the IAM's appeared in McClelland and Elman's TRACE model of spoken word recognition, in which an interactive activation network was replicated across subsequent time slices to capture the flow of activation from phonetic features to segments to words. Interesting behaviors were observed, such as the trading of different cues to segment identity and an emergent segmentation of the continuous speech stream into a single string of words simply by the activation of all possibly implicated words, combined with winner-take-all competitions. Top-down feedback in TRACE is computationally effective, but in the long-running controversy over the existence of such feedback, Norris, McQueen, and Cutler (2000) have argued that it is unwarranted and unnecessary in a model of spoken word recognition. In related models of spoken word recognition and of phoneme recognition in speech (Norris et al., 2000), interactive activation principles have been employed in purely bottom-up architectures.

Distributed Lexical Representations

The modeling of normal and impaired single-word reading is the flagship of the connectionist cognitive modeling enterprise (Seidenberg and McClelland, 1989; Plaut et al., 1996). Seidenberg and McClelland (1989) characterized lexical processing in terms of a triangle of mappings between orthographic, phonological, and semantic representations, using three-layer networks trained by the backpropagation of error. (Only in later developments of this model has semantic processing been implemented.) An identity mapping from orthographic representations to orthographic representations was used to model the capturing of a secure visual representation of the word. The model did not possess a

discretely structured lexicon; instead, information about a particular word was distributed across the many weighted connections between the three sets of representational units as the model learned the words in its training set. The goal has been to simulate the reading of regular words, irregular words (e.g., *pint*), and nonwords (e.g., *tenk*) within the same homogeneous architecture, and to explore an emergent division of labor between orthographic and semantic contributions to the pronunciation of different word types (see READING). An important improvement in later versions of this model involved employing local orthographic and phonological representations that respected the formal distinctions of the onset, nucleus, and coda of monosyllabic words (e.g., the *d*, *e*, and *sk* of *desk*). This innovation illustrates a criticism heard on both sides of the debate over connectionist cognitive modeling. Critics have stated that specifying the input and output representations relies on the formal insights of the classical, symbol-processing tradition and takes the crucially hard work out of solving the problem. Quartz and Sejnowski (1997), arguing a “constructivist” case for concentrating on the role of activity and growth in development, make the related point that it is the identification of the problem in the first place that is critical in learning and development. We might add that employing representations developed by formal linguists, such as phonemes, distinctive features, or syntactic categories, brings both advantages and disadvantages; all such categories involve ambiguity at their boundaries, even in the formal domains for which they were developed. All symbolic distinctions made about natural language are graded, in the limit.

Further improvements in the modeling of lexical processing have come from using recurrent architectures trained to settle into steady states, each corresponding to a desired output. Such attractor networks allow a feedforward model to map initially to an approximate output, which is then “cleaned up” appropriately if the initial approximation falls within the “basin of attraction” of the desired output. This innovation has been crucial in mapping between orthographic and semantic representations (Hinton and Shallice, 1991), where the relationship is essentially arbitrary. It has also allowed better generalization in the quasi-regular mapping between orthographic and phonological form, as such attractors have been shown to behave componentially in generating pronunciations.

The modeling of lexical processing has been important because of its psychologically realistic scale. The benchmark has been the pronunciation of a more or less complete slice of the lexicon, such as all the four-letter words or all the monosyllabic words of English. (Modeling with polysyllabic words has been limited.) A further aspect of the models’ scale is that they involve the full repertoire of letters and speech segments. In these respects, such modeling is psychologically grounded in a way that cannot be claimed for small models with unrepresentative training sets. The modeling of other language domains, such as morphology, syntax, or semantics, has typically not achieved this rich, realistic scale and coverage.

The folk interpretation of the mental lexicon as a dictionary suggests a discrete lexical entry for each word, and it is surprising that models based on superpositional storage can achieve so much. Models of visual and spoken word recognition employing distributed representations of words have largely superseded the localist, hardwired models such as the IAM and TRACE, which cannot capture the phenomena of learning and development and that are also relatively large architectures to build if they are to achieve wide coverage. Nevertheless, both types of connectionist model continue to generate important, experimentally testable predictions, and there has been eloquent advocacy of the merits of localist representation.

Less attention has been given to semantic processing in lexical models, partly because the central ability of connectionist models to generalize is maladaptive, given the arbitrary relationship between form and meaning, and partly because of the inherent difficulties of representing the meanings of words. However, when semantic representations have been studied, they have been based either on hand-crafted (sometimes dictionary-derived) semantic features (e.g., “+edible” or “+living”), or on corpus-derived context vectors, which specify the lexical contexts in which words are found. It is easier to see the latter type of semantic representation as being psychologically grounded: the context vectors are derived from very large corpora of real language, and wide coverage of the lexicon is more feasible than with an approach that uses semantic features.

Word Production

Research on word production has been overwhelmingly concerned with slips of the tongue and with dysphasia resulting from strokes, but there has been a growing interest in the priming of syntactic structure. Connectionist modeling of word production has not been extensive and has centered on Dell’s interactive activation model (Dell, 1986). In this model, activation initially flows downward from a lexical node to levels representing linguistic units—syllables, onsets and rimes, phonemes, and phonological features. Activation also feeds back upward, and after a set time the most activated nodes are selected for the onset, vowel, and coda slots. In a later version, semantic units activate word units and, in a second phase, word units activate phoneme units. Dell and colleagues have simulated both slips of the tongue and dysphasic errors within this framework, and have also developed learning models based on recurrent networks to avoid the limitations of hand-coded interactive activation models.

Learning Rules

Researchers have explored the analogy between a connectionist model learning the structure of a linguistic domain and an infant acquiring language. Formal theories of language acquisition had led to the conclusion that a universal grammar had to be genetically specified in some detail. Within this approach, symbol-based rules are explicitly implemented. Connectionist modeling has held out the prospect that language learning could arise from more general-purpose cognitive processing, with rule-like behavior emerging from distributed representations. The learning of morphology has attracted the most research, but attention has recently centered on the abilities of very young infants to learn the regularities present in very simple “artificial grammars” and on the capabilities of connectionist models to simulate these data.

Rumelhart and McClelland began this debate about development with their model of past tense acquisition (see PAST TENSE LEARNING). Their model attracted robust criticism, but later modeling of morphological processing, even involving the apparently problematic case of minority default forms in German, has revealed a more detailed developmental picture that has supported Rumelhart and McClelland’s original insight that a quasi-regular mapping can be captured by a model with a single mechanism operating with distributed representations of words. However, the broader debate continues, with one side exploring the capacity of models with a homogeneous architecture to account for the data while the other side argues for “dual-route” models containing explicit rules and stored exceptions to those rules.

Higher-Level Processing

Higher-level, sentential processing involves syntax and semantics and immediately differs from lexical processing in terms of productivity: a speaker can produce an infinite number of different sentences. Syntactic processing is the paradigm example of symbolic behavior: the usage of the word *John* is determined by the category to which it belongs, proper nouns, rather than by features intrinsic to that particular word, so that being able to process *John sees Kim* necessarily implies being able to process *Kim sees John*. It has been claimed that connectionist models cannot transcend their inherent capacities to associate and to generalize between similar items so as to be able to capture the productivity and systematicity of natural language. Nevertheless, a variety of models have been developed to show that the principles of the connectionist modeling of cognition can be applied to sentence processing.

There is a long history of using the constraint satisfaction abilities of localist connectionist models to parse strings of words when syntactic and semantic categories are given. Such approaches implement insights such as the role of the frequency of particular transitions or the competition between possible parses. However, they take for granted the discreteness of the given formal categories, they can say little about development, and they cannot arbitrate the psychological issue of whether any particular category or type of information (typically syntax) has priority in processing.

An important departure came with Elman's simple recurrent network (SRN) and the demonstration that such models could discover and use syntactically relevant information in sentences (Elman, 1990). The SRN was developed to take account of temporal context in a sequence of inputs with no length restriction. In a three-layer network, the state of the hidden units at a particular point in time is recycled back to the hidden units simultaneously with the new input at the next point in time. Thus, the hidden units are affected by the previous inputs and states of the network. Elman required SRN models to predict the identity of the next word in a novel sentence as a means of testing whether the models had learned syntactic and semantic generalizations from the training corpus. Elman and others have shown, using this approach, that SRNs can capture some of the processing behavior observed in human readers, involving difficulties with particular construction types. Christiansen and Chater have shown that such models are capable of behavior that resembles the human ability to use words in accordance with their constituent nature, such as using *boy* in a novel syntactic position appropriate for a noun.

These approaches to higher-level structure in language have been shown to be capable of at least beginning the process of language learning (see CONSTITUENCY AND RECURSION IN LANGUAGE). Further work has been directed to extending the complexity of the input; more construction types have been covered, and the effects of memory constraints on syntactic processing have been revealed, with implications for the observed distribution of different language types.

An influential approach to modeling adult parsing competence has been based on Pollack's Recursive Auto-Associative Memory (RAAM), in which the hidden units of the model are made to develop distributed representations of successive layers of a parse tree. The model is required to autoassociate these patterns, which are provided by the modeler. The hidden unit activations may be stored and read back off a stack, when they may be used to recreate the parse across the output units. In addition, the trained model may generalize its parsing behavior to novel inputs. A recurrent version of this model has been developed that, like the SRN, discovers sequential dependencies in its inputs that correlate with syntactic roles.

In general, parsing models that have been claimed to learn structured syntactic relationships between words also discover a variety

of other types of information relevant to parsing. In the real world, potential clues to syntactic structure may include prosody, phonosyntactic regularities, punctuation, semantic attributes of words, argument structure, discourse context, and real-world knowledge; it is a strong claim to say that any potentially useful information is ignored by the brain in assigning a parse. Some brain imaging studies suggest autonomous syntactic processing of unproblematic input and fast interaction with other types of information when ambiguity is encountered. The debate about the autonomy of syntax continues even with the advent of sophisticated imaging techniques (see IMAGING THE GRAMMATICAL BRAIN).

Overall, the period from the mid-1980s has seen an explosion in the increasingly sophisticated application of connectionist methods to the discovery and/or management of higher-level structure in language, from syntax and morphosyntax through to semantics and discourse structure. Researchers have simulated processing in such subdomains as parsing, variable binding, question answering, sentence generation, topic identification, anaphora resolution, application of world knowledge to language understanding, translation, grammaticality judgment, text compression, information retrieval, and discourse understanding. Some researchers have dealt with the more ambitious combinations of tasks using completely connectionist modular architectures, and others have augmented connectionist architectures with more traditional computational approaches, creating hybrid models. (For coverage of some of this growing field, see Reilly and Sharkey, 1992; Wermter, Riloff, and Scheler, 1996; Dale, Moisl, and Somers, 2000; and Christiansen and Chater, 2001.)

Finally, the last decade has seen growing interest in modeling the evolution of language. Connectionist architectures have been used, along with other statistical models, to simulate the evolution, by iterative learning, of simple "languages" by successive generations of individuals. The goal has been to elicit the emergence of compositional structure and other attributes of natural language.

Starting Small

One set of studies addressing development has attracted wide attention across a range of disciplines. Elman captured in network terms the intuition from language learning research that the cognitive constraints found in the developing infant may actually be advantageous to learning. Elman trained an SRN to learn syntactic dependencies separated by different numbers of intervening words in a sequential input and found that constraints that were intended to mimic the memory and attention limitations of infants acted to structure the training regimen by focusing learning on the shorter dependencies before the longer ones. The model that began with constraints that were subsequently relaxed outperformed the model that began with "adult" abilities. However, even though the notion of "starting small" still has deep implications for understanding development, Rohde and Plaut have shown that this principle may not be so unambiguously demonstrated using SRNs.

Anatomical Reality

Models have not typically incorporated observable anatomical detail beyond the general claims regarding interactivity, distributed representations, and superpositional storage (although some researchers have explored giving priority to short connections in cortical processing). Arguably there is little, if any, discrete brain anatomy exclusively dedicated to language processing, unlike the areas and pathways responsible for vision, for instance. Some neuroimaging research suggests that although some language activities may be closely associated with certain brain areas, it may be truer to say that those areas specialize in particular subtasks (e.g., Broca's area and sequence storage) that are not exclusively linguistic. Fur-

thermore, task difficulty seems to affect the configurations of activation seen (e.g., Broca's area may be only minimally activated by syntactically undemanding language), perhaps reflecting the redundancy present in language. These observations may mean that for a very long time, our best understanding of language processing may still come from relatively high-level connectionist modeling.

However, the largest anatomical distinction, the hemispheric division of the brain, indisputably has an impact on language processing. Reggia, Goodall, and Shkuro have modeled hemispheric lateralization of phonological processing during development, and Shillcock and Monaghan (2001) have modeled single-word reading based on the observation that the fovea in the human retina is vertically split and projects contralaterally to the two hemispheres.

The issue of anatomical reality is closely connected to attempts to model cognitive impairment by lesioning trained connectionist models. Indeed, a critical part of the validation of models of normal processing has been to study their ability to capture impaired processing. The modeling of dyslexia is the most developed part of this field. Some of the lesions applied to the relevant models may be interpreted anatomically, but the range of possible instantiations of such damage is large. Thus, "impairment to the orthographic representations" could refer to anything from ocular problems to hemispheric desynchronization. Finally, a different approach to anatomical specificity comes from Miikkulainen (1997), who shows how the development and subsequent lesioning of a self-organizing feature map of word meanings can result in the category-specific impairment observed in deep dyslexia, in which particular semantic categories (e.g., furniture, tools, fruit) are disproportionately affected.

Discussion

Perhaps the most important outcome of this expanding field of research is that testable predictions have been made about normal and impaired language processing in human subjects. The resulting experiments tell us more about the phenomena concerned. Some of the models represent the state-of-the-art theory about their domain, vying with nonconnectionist models for best coverage of the data; other models remain principled existence proofs of the application of soft constraint satisfaction or distributed representations, for instance, to language processing problems. However, the major debates—over top-down feedback, over the capacity of connectionist models to capture the productivity and systematicity of human language, and over the degree of modularity in language processing—have not been settled by this combination of modeling and human experimentation. The terms of the debates may well change before any such resolution is manifest. For instance, much still remains to be understood about the nature of the brain's pervasive recurrent connectivity. In addition, we are becoming aware of the speed with which visual input makes contact with sophisticated stored cortical representations.

The most secure results have arguably come from modeling that possesses psychologically realistic dimensions. Some aspect of the model might be full-scale; for instance, the number of words in a lexical model might be large enough to approach the real ambiguities of word recognition. Or the model might contain the full repertoire of representations found in a particular domain, such as the full range of phonemes. Alternatively, the model might contain a comprehensive range of *types* of representation. For instance, lexical models containing implemented semantic representations are more convincing than those that do not. Plaut and Shallice's (1993) model of deep dyslexia captures seemingly disparate data by virtue of the range of representations it contains: it simulates visual, semantic, and mixed visual/semantic errors, together with category-specific errors and a concreteness effect (see LESIONED NETWORKS AS MODELS OF NEUROPSYCHOLOGICAL DEFICITS). A

further dimension of psychological reality involves how the model is tested. For example, in modeling the pronunciation of a written word, the error over the representation of the whole word at the output of the model may be an inappropriate measure of the difficulty of naming that word if pronunciation can commence on the strength of the processing of the first part of the word alone. Finally, models of language processing can be psychologically grounded by extending them to other languages. Individual human languages differ substantially as to where their complexity resides, and lexical, morphological, and syntactic categories may not be closely comparable across languages. Thus, it is possible to apply some of the insights present in connectionist models of the pronunciation of English words to the pronunciation of Chinese characters and words (indeed, Perfetti and Tan discuss a nonimplemented interaction activation model), but the orthographies are so different that only the major themes of the modeling of English pronunciation survive. Nonetheless, such a cross-linguistic perspective can provide valuable insights into the nature of language impairments and into the parameter space within which models of normal processing may exist.

When a model comprehensively covers a domain, the connectionist approach may have more in common with the conventional statistical analysis of that domain. Connectionist modeling may allow the researcher to avoid making certain representational decisions, and lesioning the trained model can be a convenient means of producing a picture of impairment. Nonetheless, the behaviors of such models will typically be relatively opaque, and they are usually most convincingly explained in terms of the statistics of the training regimen, once the researcher knows the relevant statistic to look for. Accordingly, connectionist modeling and the conventional statistical exploration of language corpora may often be complementary: the former demonstrates mechanisms but often cannot be full-scale, and the latter confirms that the training regimen was representative of the language in its fullest extent.

Finally, the relationships between production and perception, and between spoken and written language, are critical areas for further research. The continuing uncertainty about just what representations and processes are shared across these major divides speaks to the difficulties of these issues. It is also testament to the flexibility of the brain in incorporating the daunting cognitive task of reading, a recent cultural innovation, into the brain's older mastery of spoken communication.

In summary, there has been important progress in many areas of connectionist-based research into language processing, and this modeling continues to influence psychological and neuropsychological experimentation and observation.

Road Map: Linguistics and Speech Processing

Related Reading: Constituency and Recursion in Language; Language Acquisition; Neurolinguistics; Optimality Theory in Linguistics; Past Tense Learning

References

- Christiansen, M. H., and Chater, N., 2001, *Connectionist Psycholinguistics*, Westport, CT: Ablex. ♦
- Coltheart, M., Curtis, B., Atkins, P., and Haller, M., 1993, Models of reading aloud: Dual-route and parallel-distributed-processing approaches, *Psychol. Rev.*, 100:589–608.
- Dale, R., Moisl, H., and Somers, H., 2000, *Handbook of Natural Language Processing*, New York: Marcel Dekker. ♦
- Dell, G. S., 1986, A spreading activation theory of retrieval in language production, *Psychol. Rev.*, 93:283–321.
- Elman, J. L., 1990, Finding structure in time, *Cognit. Sci.*, 14:179–211.
- Hinton, G. E., and Shallice, T., 1991, Lesioning an attractor network: Investigations of acquired dyslexia, *Psychol. Rev.*, 98:74–95.

- McClelland, J. L., and Rumelhart, D. E., 1981, An interactive activation model of context effects in letter perception: Part 1. An account of basic findings, *Psychol. Rev.*, 88:375–407.
- Miikkulainen, R., 1997, Dyslexic and category-specific aphasic impairments in a self-organizing feature map model of the lexicon, *Brain Lang.*, 59:334–366.
- Norris, D., McQueen, J. M., and Cutler, A., 2000, Merging information in speech recognition: Feedback is never necessary, *Behav. Brain Sci.*, 23:352–363.
- Plaut, D. C., McClelland, J. L., Seidenberg, M. S., and Patterson, K., 1996, Understanding normal and impaired word reading: Computational principles in quasi-regular domains, *Psychol. Rev.*, 103:56–115.
- Plaut, D. C., and Shallice, T., 1993, Deep dyslexia: A case study of connectionist neuropsychology, *Cognit. Neuropsychol.*, 10:377–500.
- Quartz, S. R., and Sejnowski, T. J., 1997, The neural basis of cognitive development: A constructivist manifesto, *Behav. Brain Sci.*, 20:537–596.
- Reilly, R. G., and Sharkey, N. E., 1992, *Connectionist Approaches to Natural Language Processing*, Hove, Engl.: Erlbaum. ♦
- Seidenberg, M. S., and McClelland, J. L., 1989, A distributed, developmental model of word recognition and naming, *Psychol. Rev.*, 96:523–568.
- Shillcock, R. C., and Monaghan, P., 2001, The computational exploration of visual word recognition in a split model, *Neural Computat.*, 13:1171–1198.
- Wermter, S., Riloff, E., and Scheler, G., 1996, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, New York: Springer-Verlag. ♦

Layered Computation in Neural Networks

Hanspeter A. Mallot

Introduction

Layering is a common architectural feature of many neural subsystems, both in vertebrate and in invertebrate brains. It is best studied in the mammalian neocortex but can be found in regions as diverse as the optic tectum, the avian visual wulst, or the cephalopod optic lobe. In a broader sense, layered neural areas with strong vertical connectivity and topographic organization of input and output may be called *cortical* in all these different structures.

Neural layers are characterized by various anatomical and physiological parameters, such as relative abundance of cell classes, soma size, pharmacology, and both intrinsic and interarea connectivity (Braitenberg and Schüz, 1991; see also NEUROANATOMY IN A COMPUTATIONAL PERSPECTIVE). These parameters remain constant within a two-dimensional sheet but vary between sheets. In contrast, the “layers” of artificial neural networks are defined by topology only (block structure of the connectivity matrix), leaving no room for geometrical concepts such as two-dimensional extent. Some important properties of cortical layering are as follows:

1. Both intralayer and interlayer connectivity are largely determined by spatial constraints, such as nearness. In particular, the two-dimensional topology of layers is important.
2. Connections between two neurons can be mediated by multiple synapses located in different layers. If detailed timing is considered, this multiplicity of connections can be functionally significant, owing to differences in propagation time.
3. Layers can be void of nerve cell somata, mediating fiber contacts of neurons whose somata are located elsewhere (e.g., the molecular layer of the neocortex).
4. Feedback connections can occur even within each layer.

This article deals with three aspects of layering: quantitative descriptions of layered or cortical organization, the activation dynamics of network models incorporating this organization, and applications to problems of information processing and computation. Although the concepts are general, examples will repeatedly be drawn from (primate) visual cortex.

Quantitative Anatomy

Uniformity and Continuous Models

Given the vast numbers of cells found in layered cortical areas, it seems appropriate to build spatially continuous models (neural

“fields”) where each point in space corresponds to a neuron (Korn and von Seelen, 1972; Amari, 1977; see also the references in Mallot and Giannakopoulos, 1992). Besides being a good approximation of large neuron numbers, the continuous description allows for a natural modeling of position and distance of neurons. In a continuous model, a “neuron” consists of (1) a point on the sheet at the position of its soma, (2) a cloud or density function of postsynaptic (dendritic) sites specifying the input sensitivity for each point on the sheet, (3) a density function of postsynaptic (axonal) sites, and (4) an appropriate activation function (see the discussion that follows). The usefulness of these model features rests on two assumptions:

- The strength (or likelihood) of a connection between two neurons is proportional to the overlap of their dendritic and axonal clouds and the efficiencies of the presynaptic and postsynaptic sites involved. This assumption is discussed at length by Braitenberg and Schüz (1991), who have termed it “Peters’s rule” (see also Peters, 1985:64ff).
- Intrinsic connectivity is largely uniform; i.e., the fiber clouds of the neurons are shifted versions of each other.

While space variance (nonuniformity) presents a problem to the sketched continuous approach, some common cases can be dealt with rather easily: e.g., by topographic maps between brain areas and modulations of neuron density within single layers. Topographic mapping can be modeled in terms of piecewise, continuous, point-to-point coordinate transforms. Explicit mathematical functions have been derived from known or assumed distributions of areal magnification by integrating the “mapping-magnification equation” in one or two dimensions (Schwartz, 1980, log z ; Mallot and Giannakopoulos, 1992). Point-to-point models have also been proposed for columnar input patterns, such as ocular dominance stripes (Mallot, von Seelen, and Giannakopoulos, 1990). An important case of intrinsic space variance is the columnar pattern of cell densities such as the one revealed by the cytochrome oxidase stain. In the continuous model, this can be accounted for by a space-variant density factor.

Populations, Layers, and Areas

In the continuous approach, the unit of modeling is the neural population, i.e., a set of neurons from the same anatomical class with a space-invariant connectivity pattern. Examples of such populations

in the visual system (see VISUAL CORTEX: ANATOMICAL STRUCTURE AND MODELS OF FUNCTION) are the spiny stellate cells in layer 4a, the GABAergic cells in layer 3, the pyramidal cells of layer 6 connecting to the lateral geniculate nucleus, and so on. The neural population is characterized by a number of variables that fall into three groups. *Anatomical variables* include the dendritic and axonal fiber clouds, $\delta(\mathbf{x})$, $\alpha(\mathbf{x})$; cell density, $\rho(\mathbf{x})$; and topographic output maps, $\mathcal{R}(\mathbf{x})$. Physiological variables include a nonlinear compression function, $f(u)$; time delays for the propagation of activity, T ; synaptic integration time, τ ; and gain factors. Activity is described by three *state variables*: input, $s(\mathbf{x}, t)$; potential, $u(\mathbf{x}, t)$; and output, $e(\mathbf{x}, t)$, where $\mathbf{x} \in \mathbb{R}^2$.

When connecting neural populations into networks, it is convenient (although slightly redundant) to keep the two separate state variables for input and output, the somatic activity $e(\mathbf{x}, t)$, and the “synaptic” activities $s(\mathbf{x}, t)$. The idea is that the output of one population is not the immediate input to some other population. Rather, several outputs from different populations are accumulated into one distribution of presynaptic activity, which then feeds into all neural populations with an appropriate dendritic port. We call the support of the presynaptic activity *connection planes*; they are indexed by $l \in \{1, \dots, L\}$ in Equations 1 and 3 in the next section. Connection planes are reminiscent of the *blackboard* structure in multi-agent computer systems (see MULTIAGENT SYSTEMS) in that they collect activity from several populations without keeping track of the original source of the activity. When in turn a neural population “reads” from the connection plane, it cannot know whose activity it is reacting to; this lack of labeling of the activities is a direct consequence of Peters’s rule.

Using the idea of connection planes, we can now give a definition of the terms *layer* and *area*. A *layer* is a connection plane together with all neuron populations whose somata are located in that plane. (The number of populations in a layer may be zero, as is the case in the molecular layer of the cortex. There may also be just one population allowing for specific circuitry; in this case, the pooling effect of the connection planes is bypassed and the distinction between layers and populations becomes obsolete.) An *area* is a set of layers connected without topographic maps (other than the identity). That is to say, in the case of modeling a visual system, just one retinotopic map, \mathcal{R}_i , is assigned to each visual area A_i , which is valid for all its layers. For projections between different areas—e.g., from A_i to A_j —a mapping of the form $\mathcal{R}_j \circ \mathcal{R}_i^{-1}$ is required to connect points representing the same retinal location.

Activation Dynamics

Network Equations

Consider a network of P neural populations (index p , state variables $e_p(\mathbf{x}, t)$ and $u_p(\mathbf{x}, t)$) connected via L connection planes (index l , state variable $s_l(\mathbf{x}, t)$). The activation dynamics of the resulting network can be formulated in three steps (cf. Figure 1 and Mallot and Giannakopoulos, 1992):

1. Dendritic summation ($s_l \rightarrow u_p$): For each population p , inputs from different connection planes s_1, \dots, s_L are accumulated according to dendritic arborizations, δ_{pl} ; delays, T_l^δ , and synaptic integration time, τ :

$$\frac{\partial}{\partial t} u_p(\mathbf{x}, t) = -\frac{u_p(\mathbf{x}, t)}{\tau} + \sum_{l=1}^L \int s_l(\mathbf{x}', t - T_l^\delta) \delta_{pl}(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (1)$$

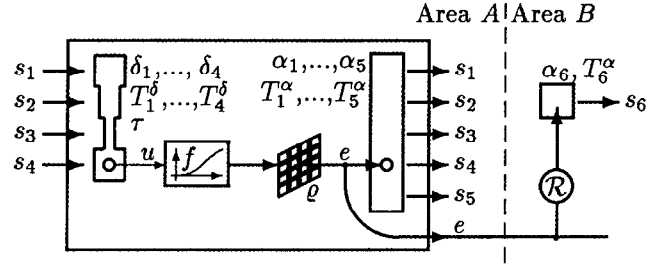


Figure 1. Activation transfer function of a neural population modeling dendritic summation, somatic point operations, and axonal spreading of activity. In Equations 1–3 (explained in the text), an additional index p is used to distinguish between different populations.

2. Somatic point operations ($u_p \rightarrow e_p$): The resulting intracellular potential is passed through a nonlinearity f_p and locally weighted with the density of the cell population $\rho_p(\mathbf{x})$. For example, if we consider a cell population in the magnocellular stream, $\rho_p(\mathbf{x})$ reflects the pattern of cytochrome oxidase blobs (see VISUAL CORTEX, ANATOMICAL STRUCTURE AND MODELS OF FUNCTION)

$$e_p(\mathbf{x}, t) = \rho_p(\mathbf{x}) f_p(u_p(\mathbf{x}, t)) \quad (2)$$

3. Axonal spread ($e_p \rightarrow s_l$): The resulting excitation is spread over the axonal densities α_{lp} and added to the activity of the connection layer to which the axon projects, again with appropriate delays (propagation times) T_l^α . For axons from population p projecting to a connection layer l in another cortical area, a point-to-point mapping \mathcal{R}_{lp} has to be considered.

$$s_l(\mathbf{x}, t) = \sum_{p=1}^P \int e_p(\mathbf{x}', t - T_l^\alpha) \alpha_{lp}(\mathbf{x} - \mathcal{R}_{lp}(\mathbf{x}')) d\mathbf{x}' \quad (3)$$

Integro-differential equations of the leaky integrator type, such as the equations just given, have been studied extensively. One important special case is the interaction of two populations—one excitatory and one inhibitory—with all of the space-variant terms in Equations 1–3 omitted (e.g., Amari, 1977; Ermentrout and Cowan, 1979; Chipalkatti and Arbib, 1988; Murray, 1989). The formulation given here allows for space variances, both in the inter-area connections (topographic mapping functions \mathcal{R}_{lp} in Equation 3) and in the cell densities (ρ_p in Equation 2). In addition, cell populations can be multiply connected via different cortical layers so that each path has its own spatiotemporal characteristic (Krone et al., 1986; Mallot et al., 1990; Mallot and Giannakopoulos, 1992).

Receptive Fields and Point Images

When stimulated with an external signal, $s_{\text{ext}}(\mathbf{x}, t)$, the network reacts with a distribution of activity $e(\mathbf{y}, t)$ that corresponds to the neural representation of the stimulus. As an example, let \mathbf{x} denote retinal and \mathbf{y} cortical coordinates. In neurophysiological experiments, the relation between stimulus and excitation is often described by two so-called characteristic functions that can easily be modeled in continuous neural networks:

- The point image, point spread function, or impulse response, $p_{ps}(\mathbf{y}, t)$, is the distribution of activity resulting from stimulation

- with a spatiotemporal Dirac function, $\delta(\mathbf{x}, t)$, e.g., a briefly flashed spot of light in the visual system. If the system were linear, space invariant, and stationary, responses to arbitrary stimuli could be predicted by superposition of appropriately shifted, delayed, and weighted impulse responses (convolution).
- The receptive field profile $p_{rf}(\mathbf{x}, t)$ of a cortical unit at position \mathbf{y} describes the influence that each input site \mathbf{x} at each instant in time has on the unit in question. In linear, space-invariant, and stationary systems, point spread function and receptive field are identical up to a mirroring in spatial and temporal coordinates (e.g., while p_{rf} “looks backward in time,” p_{ps} “looks forward”). In general linear systems, they are the kernels of adjoint operators (Mallot et al., 1990).

Point images and receptive fields are most useful in linear systems, where they completely describe the stimulus-response behavior by way of superposition. In order to interpret neurophysiological measurements of these functions, nonlinear approaches are required. Possible choices are (1) cascades of linear systems with stationary nonlinearities, (2) Wiener-Volterra expansions (usually terminated after order 2), and (3) nonlinear network equations, such as presented earlier in Equations 1–3.

Receptive Field Properties

Figure 2 shows four steps for increasing realism in the modeling of receptive fields. Figure 2A illustrates the space- and time-

invariant feedforward system, where the spatiotemporal version of linear systems theory can be applied (Korn and von Seelen, 1972). Since the early work on lateral inhibition in the compound eye of the chelicerate *Limulus*, a number of filter functions have been discussed as models of both receptive fields and spatial vision (difference of Gaussians, various derivatives of Gaussians, Gabor functions). Many of these are now used as filters in image processing. In the feedforward case, the spatial and temporal parts of the neural “filter function” are usually considered separable, equivalent to a cascade of two steps, one of which is temporal only and the other of which is spatial only. Simple nonseparability can be introduced by adding several of these cascades (e.g., one for the on-center and another for the off-surround of a retinal ganglion cell; see Dinse, Krüger, and Best, 1990). One important computational application of the resulting spatiotemporal filters is the processing of visual motion (Korn and von Seelen, 1972). Interestingly, many more can be found, if receptive fields specifically responding to other stimulus parameters (orientation, velocity, spectra, etc.) are considered (Adelson and Bergen, 1991; Mallot et al., 1990).

Parts B and C of Figure 2 show simple extensions of the space-invariant feedforward situation. In Figure 2B, many layers with feedback connections are considered (Krone et al., 1986). The main effects of this architecture include (1) increased width of receptive fields, since point stimuli can be signaled through the entire network by feedback connections, and (2) full nonseparability of spatial and temporal aspects of the receptive field. The first of these effects has been used by Horn (1974) in the deconvolution step of the “retinex” scheme for recovering lightness from image intensities. This article also introduces the idea of resistive networks for image processing which links the continuous Equations 1–3 to discrete implementations, as well as to diffusion-type equations in which spatial interaction is modeled by partial derivatives rather than by integral kernels.

In Figure 2C, the feedforward situation is extended by allowing for space variance by retinotopic mapping (Mallot et al., 1990). The combination of mapping and feedback illustrated in Figure 2D has not yet been studied in detail. Its activation dynamics is described by Equations 1–3 cited earlier.

Information Processing Capabilities of Neural Layers

The filter operations discussed earlier exploit the neighborhood relations in a neural layer. Other features that can be used for computational purposes are nonlinear activation dynamics and neural maps. Some examples include the following:

1. *Lateral cooperativity.* Lateral interactions between neural activities in a layer can have cooperative and/or inhibitory effects, leading to filling-in or related kinds of shaping of the activity pattern (Murray, 1989). One important application of this principle is the solution of the correspondence problem in stereo-vision by means of cooperative dynamics in a disparity map (for a review, see Blake and Wilson, 1991; see also Chipalkatti and Arbib, 1988, and STEREO CORRESPONDENCE). Other examples of nonlinear lateral interactions include various winner-take-all or nonmaximum-suppression schemes that are widely used in artificial neural networks.
2. *Topographic mapping.* While the continuity of neural representations is a prerequisite for neighborhood operations, such as filtering, the smooth distortions introduced by topographic mapping can simplify subsequent information processing tasks. Examples include the allocation of cortical neurons to different parts of the visual field (fovea/periphery), and the simplified processing of images with systematic space variances, such as optic flow patterns. The optic flow resulting from translation in a plane can be compensated for by so-called inverse perspective

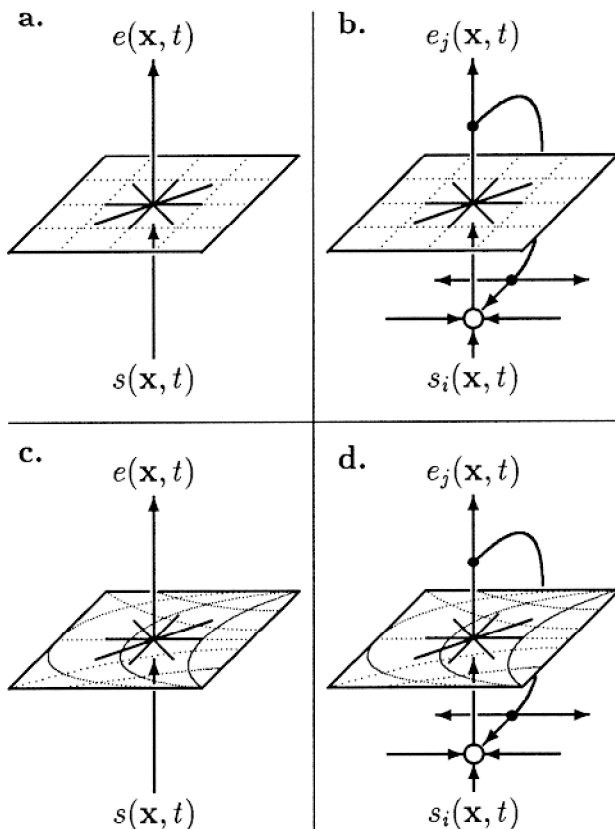


Figure 2. Continuous layers as models of receptive fields. A, Space-invariant, feedforward (spatiotemporal convolution). B, Space-invariant network of continuous layers. C, Space-variant, feedforward. D, Space-variant network of continuous layers.

mapping of the input images. Obstacles in the way of the observer lead to uncompensated changes in the flow field that are easily detected. A review of applications of topographic mapping to image processing problems is given elsewhere (Mallot et al., 1990).

3. *Feature maps and population coding.* While the examples presented so far apply to sensory input, analogous results have been obtained for motor pathways. Here, the distribution of activity on a neural layer has to be interpreted in terms of the “motor fields” of its active neurons. If this is done, the flow of activity in the appropriate motor areas predicts the initiated movements.

Discussion

The type of cortex model sketched in this article has the advantage of modeling a prominent structural unit of the vertebrate nervous system, the neural layer, on a rather high level. It can easily deal with geometrical features, such as maps, columns, dendritic and axonal arborization patterns, varying cell densities, and the like. This level of modeling is required to understand large-scale activation dynamics of cortical networks as have been made accessible by recently developed imaging techniques. The continuity limit seems appropriate when entire cortical areas are to be represented in a neural network model. On the other hand, it is not very well suited to model properties that differ from one cell to the next. For example, synaptic plasticity is not easily included. It is, therefore, most useful for the modeling of rather short time scales, where plasticity may be excluded, and for systems in steady states.

[Reprinted from the First Edition]

Road Map: Biological Networks

Background: I.3. Introducing the Neuron

Related Reading: Directional Selectivity; Gabor Wavelets and Statistical Pattern Recognition; Pattern Formation, Neural; Thalamus

References

- Adelson, E. H., and Bergen, J. R., 1991, The plenoptic function and the elements of early vision, in *Computational Models of Visual Processing* (M. S. Landy and J. A. Movshon, Eds.), Cambridge, MA: MIT Press, pp. 3–20. ♦
- Amari, S.-I., 1977, Dynamics of pattern formation in lateral-inhibition type neural fields, *Biol. Cybern.*, 27:77–87.
- Blake, R., and Wilson, H. R., 1991, Neural models of stereoscopic vision, *Trends Neurosci.*, 14:445–452. ♦
- Braitenberg, V., and Schüz, A., 1991, *Anatomy of the Cortex: Statistics and Geometry*, Berlin: Springer-Verlag. ♦
- Chipalkatti, R., and Arbib, M. A., 1988, The cue integration model of depth perception: A stability analysis, *J. Math. Biol.*, 26:235–262.
- Dinse, H. R., Krüger, K., and Best, J., 1990, A temporal structure of cortical information processing, *Concepts Neurosci.*, 1:199–238.
- Ermentrout, G. B., and Cowan, J. D., 1979, A mathematical theory of visual hallucination patterns, *Biol. Cybern.*, 34:137–150.
- Horn, B. K. P., 1974, Determining lightness from an image, *Comput. Vis. Graph. Image Proc.*, 3:277–299.
- Korn, A., and von Seelen, W., 1972, Dynamische Eigenschaften von Nervennetzen im visuellen System, *Kybernetik*, 10:64–77.
- Krone, G., Mallot, H. A., Palm, G., and Schüz, A., 1986, The spatio-temporal receptive field: A dynamical model derived from cortical architectonics, *Proc. R. Soc. Lond. B Biol. Sci.*, 226:421–444.
- Mallot, H. A., and Giannakopoulos, F., 1992, Activation dynamics of space-variant continuous networks, in *Neural Network Dynamics* (J. G. Taylor, E. R. Caianiello, R. M. J. Cotterill, and J. W. Clark, Eds.), Berlin: Springer-Verlag, pp. 341–355.
- Mallot, H. A., von Seelen, W., and Giannakopoulos, F., 1990, Neural mapping and space-variant image processing, *Neural Netw.*, 3:245–263.
- Murray, J. D., 1989, *Mathematical Biology*, Berlin: Springer-Verlag, chap. 16. ♦
- Peters, A., 1985, Visual cortex of the rat, in *Cerebral Cortex*, vol. 3, *Visual Cortex* (A. Peters and E. G. Jones, Eds.), New York: Plenum Press, pp. 19–80. ♦
- Schwartz, E. L., 1980, Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding, *Vis. Res.*, 20:645–669.

Learning and Generalization: Theoretical Bounds

Ralf Herbrich and Robert C. Williamson

Introduction

The fundamental difference between a system that learns and one that merely memorizes is that the learning system *generalizes* to unseen examples. In order to understand the performance of learning machines and to gain insight helpful for designing better ones, it is useful to have theoretical bounds on the generalization ability of the machines. The determination of such bounds is the subject of this article. In order to formulate the bounds it is first necessary to formalize the learning problem and turn the question of how well a machine generalizes into a mathematical question. In the next section we introduce one possible formalization, the one adopted in the field of statistical learning theory.

Formalization of the Learning Problem

To study the learning problem in a mathematical framework, we assume the existence of an *unknown* distribution $\mathbf{P}_{\mathbf{X}\mathbf{Y}}$ over an *input space* \mathcal{X} (e.g., \mathbb{R}^n) and an *output space* \mathcal{Y} (e.g., $\{0, 1\}$). We are given only a *sample* $\mathbf{z} = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m = \mathcal{Z}^m$, which is assumed to be drawn *iid* (independent identically

distributed) from $\mathbf{P}_{\mathbf{X}\mathbf{Y}}$; we define $\mathbf{P}_{\mathbf{Z}} := \mathbf{P}_{\mathbf{X}\mathbf{Y}}$. (In this article, random variables are always written sans-serif, e.g., \mathbf{X} .)

In an attempt to discover the unknown relation $\mathbf{P}_{\mathbf{Y}|\mathbf{X}=\cdot}$ between inputs and outputs, a *learning algorithm* \mathcal{A} chooses a deterministic *hypothesis* $h: \mathcal{X} \rightarrow \mathcal{Y}$ solely on the basis of a given training sample $\mathbf{z} \in \mathcal{Z}^m$. Formally,

$$\mathcal{A} : \bigcup_{i=1}^{\infty} \mathcal{Z}^i \rightarrow \mathcal{H},$$

where $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is the *hypothesis space* used by the algorithm. (Recall that $\mathcal{Y}^{\mathcal{X}}$ denotes the set of maps from \mathcal{X} to \mathcal{Y} .) Some of the bounds take account of more information regarding \mathcal{A} than just \mathcal{H} .

The performance of the learning algorithm is judged according to a *loss function* $l: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$, which measures the cost of the prediction \hat{y} if y is the correct output. The choice of the loss function is a key part of the formal specification of the learning problem. The *learning problem* is to find an hypothesis, $h: \mathcal{X} \rightarrow \mathcal{Y}$, such that the *expected risk*, $R[h] := \mathbf{E}_{\mathbf{X}\mathbf{Y}}[l(h(\mathbf{X}), \mathbf{Y})]$, is minimized.

Pattern recognition. In this case, $|\mathcal{Y}| < \infty$. Typically one is interested in the misclassification error $\mathbf{P}_{\mathbf{X}\mathbf{Y}}(h(\mathbf{X}) \neq \mathbf{Y})$. This can be

modeled by the *zero-one loss*, $l_{0-1}(\hat{y}, y) := \mathbb{I}_{\hat{y} \neq y}$. (Here \mathbb{I} denotes the indicator function.) More complex loss functions are obtained by using a cost matrix $\mathbf{C} \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$.

Function learning. Here, $\mathcal{Y} = \mathbb{R}$. The classical regression scenario utilizes squared loss, $l_2(\hat{y}, y) := (\hat{y} - y)^2$. Other loss functions are the ℓ_1 loss function, $l_1(\hat{y}, y) := |\hat{y} - y|$, and the ε -insensitive loss, $l_\varepsilon(\hat{y}, y) := \max\{|\hat{y} - y|, \varepsilon\} - \varepsilon$.

If we knew \mathbf{P}_Z , the solution of the learning problem would be straightforward:

$$h_{\text{opt}}(x) := \operatorname{argmin}_{y \in \mathcal{Y}} \mathbf{E}_{Y|X=x}[l(y, Y)]. \quad (1)$$

The fact that h_{opt} cannot be identified only on the basis of the training sample \mathbf{z} is the motivation for studying *theoretical bounds* on the generalization error of learning algorithms. These bounds are only valid for most random draws of the training sample. Formally, they read as follows:

$$\mathbf{P}_Z(R[\mathcal{A}(\mathbf{Z})] \leq \varepsilon_{\mathcal{A}}(\mathbf{Z}, \dots, \delta)) \geq 1 - \delta. \quad (2)$$

In the analysis of such bounds, it is convenient to think of the loss function induced function class

$$\mathcal{L}_{\mathcal{H}} := \{(x, y) \mapsto l(h(x), y) \mid h \in \mathcal{H}\}.$$

For simplicity, we will mostly consider the pattern recognition case and the zero-one loss; the reasoning in the function learning case is conceptually similar.

Consistency of Learning Algorithms

Consistency is a property of a learning algorithm that guarantees that in the limit of an infinite amount of data, the learning algorithm will achieve the minimum possible expected risk. The definition is relative to a fixed hypothesis space $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and requires

$$\forall c > 0: \lim_{m \rightarrow \infty} \mathbf{P}_Z(R[\mathcal{A}(\mathbf{Z})] - \inf_{h \in \mathcal{H}} R[h] > c) = 0. \quad (3)$$

For the results stated below (Vapnik, 1998), a more complex notion of *nontrivial consistency* is needed. In particular, this notion requires that Equation 3 holds even if $\mathcal{H}_c := \{h \in \mathcal{H} \mid R[h] \geq c\}$ for all $c \in \mathbb{R}$. Note that in this case, $\inf_{h \in \mathcal{H}_c} R[h] = c$. It is known that for the class of *empirical risk minimization (ERM) algorithms*

$$\mathcal{A}_{\text{ERM}}^{\mathcal{H}}(\mathbf{z}) := \operatorname{argmin}_{h \in \mathcal{H}} \underbrace{\frac{1}{m} \sum_{i=1}^m l(h(x_i), y_i)}_{\hat{R}[h, \mathbf{z}] \text{ (the empirical risk)}}$$

consistency is equivalent to uniform one-sided convergence of empirical risks to expected risk; that is,

$$\forall \varepsilon > 0: \lim_{m \rightarrow \infty} \mathbf{P}_Z(\sup_{h \in \mathcal{H}} (R[h] - \hat{R}[h, \mathbf{Z}]) > \varepsilon) = 0 \quad (4)$$

A slightly stronger condition than that in Equation 4, namely uniform two-sided convergence, is equivalent to

$$\forall \varepsilon > 0: \lim_{m \rightarrow \infty} \frac{\ln(\mathbf{E}_Z[\mathcal{N}(\varepsilon, \mathcal{L}_{\mathcal{H}}, \mathbf{Z})])}{m} = 0 \quad (5)$$

where $\mathcal{N}(\varepsilon, \mathcal{L}_{\mathcal{H}}, \mathbf{z})$ is the *covering number* of $\mathcal{L}_{\mathcal{H}}$ on the sample \mathbf{z} at scale ε . This is the smallest number of functions $\hat{g}: \mathcal{X} \rightarrow \mathbb{R}$ such that for every induced loss function $g \in \mathcal{L}_{\mathcal{H}}$ there exists a function \hat{g} with

$$\frac{1}{m} \sum_{i=1}^m |g(z_i) - \hat{g}(z_i)| \leq \varepsilon.$$

In the case of the zero-one loss, l_{0-1} , the covering number $\mathcal{N}(1/m, \mathcal{L}_{\mathcal{H}}, \mathbf{z})$ equals the number of different error patterns $(g(z_1), \dots, g(z_m)) \in \{0, 1\}^m$ incurred by induced loss functions $g \in \mathcal{L}_{\mathcal{H}}$.

This *characterization* result (that consistency of $\mathcal{A}_{\text{ERM}}^{\mathcal{H}}$ is “almost” equivalent to Equation 5) is the justification for the central place that covering numbers play in statistical learning theory. It is important to note that the results are only for $\mathcal{A}_{\text{ERM}}^{\mathcal{H}}$. It is still an open problem to characterize consistency for algorithms other than $\mathcal{A}_{\text{ERM}}^{\mathcal{H}}$, and thus it is not known what their “right” technical parameters are.

Theoretical Bounds for Learning Algorithms

The starting point of all the analysis presented here is the observation that for a *fixed* hypothesis $h: \mathcal{X} \rightarrow \mathcal{Y}$ (and induced loss function $g, g((x, y)) := l(h(x), y)$), we know that

$$\begin{aligned} \mathbf{P}_Z(R[h] - \hat{R}[h, \mathbf{Z}] > \varepsilon) &= \mathbf{P}_Z\left(\mathbf{E}_Z[g(\mathbf{Z})] - \frac{1}{m} \sum_{i=1}^m g(\mathbf{Z}_i) > \varepsilon\right) \\ &< \exp(-c \cdot m\varepsilon^\beta) \end{aligned} \quad (6)$$

where c is some constant and $\beta \in [1, 2]$, if the loss is bounded or has bounded moments. This is due to well-known results in large deviation theory (see Devroye and Lugosi, 2001, chap. 1).

The second tool is the *union bound*, which states that for events A and B ,

$$\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B) - \mathbf{P}(A \cap B) \leq \mathbf{P}(A) + \mathbf{P}(B)$$

As a consequence, if we consider a hypothesis space of finite size, say n , then the chance that for at least one of the hypotheses the expected risk is larger than the empirical risk by more than ε is of order $n \cdot \exp(-m\varepsilon^\beta)$. The general application of this simple inequality for learning theory is that given n *high-probability bounds* $Y_i: \mathcal{Z}^m \times \dots \times [0, 1] \rightarrow \{\text{false}, \text{true}\}$ such that

$$\forall i \in \{1, \dots, n\}: \forall \delta \in [0, 1]: \mathbf{P}_Z(Y_i(\mathbf{Z}, \dots, \delta)) \geq 1 - \delta, \quad (7)$$

then

$$\forall \delta \in [0, 1]: \mathbf{P}_Z\left(Y_1\left(\mathbf{Z}, \dots, \frac{\delta}{n}\right) \wedge \dots \wedge Y_n\left(\mathbf{Z}, \dots, \frac{\delta}{n}\right)\right) \geq 1 - \delta.$$

There are two conceptual simplifications that aid the study of the generalization performance of learning algorithms:

1. **Algorithm independence:** Motivated by Equation 4, consider the uniform convergence and bound this probability. This automatically gives a bound which holds for all hypotheses, including the one learned with a given learning algorithm. Although this is a very crude step, it has largely dominated statistical learning theory for the past 30 years; the whole analysis is independent of the learning algorithm used except via \mathcal{H} .
2. **Data independence:** If the training sample is entering the bound only via the empirical risk, we call the analysis *sample independent*, as we are unable to exploit the serendipity of the training sample to obtain a better bound.

Algorithm-Independent Bounds

Algorithm-independent analysis has historically been the most common. Below we examine the Vapnik-Chervonenkis framework, data-dependent structural risk minimization, and the PAC-Bayesian framework.

The Vapnik-Chervonenkis framework. The Vapnik-Chervonenkis (VP) framework, established in 1971, studies $\mathcal{A}_{\text{ERM}}^{\mathcal{H}}$ via uniform

convergence (see Vapnik, 1998, and Anthony and Bartlett, 1999, for more details). The bounds are sample independent in the sense defined above. The only extra tool required is the *basic lemma*. This result makes precise the idea that whenever it is likely that two empirical risks measured on a training sample and a *ghost sample* (another sample of the same size drawn independently) are close to each other, then it must also be likely that the empirical risk on a training sample is close to the expected risk. A result of this is a generalization bound in terms of $\mathbf{E}_{\mathbf{Z}^{2m}}[\mathcal{N}(1/2m, \mathcal{L}_{\mathcal{H}}, \mathbf{Z})]$, where the $2m$ is a consequence of the basic lemma. However, this is still not really useful, since computing $\mathbf{E}_{\mathbf{Z}^{2m}}[\mathcal{N}(1/2m, \mathcal{L}_{\mathcal{H}}, \mathbf{Z})]$ requires knowledge of the distribution $\mathbf{P}_{\mathbf{Z}}$. For l_{0-1} loss, use is made of the inequalities

$$\mathbf{E}_{\mathbf{Z}^{2m}}\left[\mathcal{N}\left(\frac{1}{2m}, \mathcal{L}_{\mathcal{H}}, \mathbf{Z}\right)\right] \leq \sup_{z \in \mathcal{Z}^{2m}} \mathcal{N}\left(\frac{1}{2m}, \mathcal{L}_{\mathcal{H}}, z\right) \leq \left(\frac{2em}{d_{\mathcal{H}}}\right)^{d_{\mathcal{H}}},$$

where $d_{\mathcal{H}}$ is known as the *VC-dimension* of \mathcal{H} :

$$d_{\mathcal{H}} := \max \left\{ m \in \mathbb{N} \mid \sup_{z \in \mathcal{Z}^{2m}} \mathcal{N}\left(\frac{1}{2m}, \mathcal{L}_{\mathcal{H}}, z\right) = 2^m \right\}.$$

The generalization bound for the zero-one loss l_{0-1} then reads as follows: *With probability at least $1 - \delta$ over the random draw of the training sample $z \in \mathcal{Z}^m$, for all hypotheses $h \in \mathcal{H}$, $R[h] \leq \varepsilon_{\text{VC}}(z, d_{\mathcal{H}}, \delta)$, where*

$$\varepsilon_{\text{VC}}(z, d_{\mathcal{H}}, \delta) := \hat{R}[h, z] + \underbrace{\sqrt{\frac{8}{m} \left(d_{\mathcal{H}} \ln \left(\frac{2em}{d_{\mathcal{H}}} \right) + \ln \left(\frac{4}{\delta} \right) \right)}}_{\text{effective complexity}}. \quad (8)$$

The key term in this bound is labeled the *effective complexity* and in this case is essentially determined by the VC-dimension $d_{\mathcal{H}}$. Note that for general loss functions $L: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$, similar results are obtained by studying the family $\{(\hat{y}, y) \mapsto \mathbb{I}_{L(\hat{y}, y) > \theta} \mid \theta \in \mathbb{R}\}$ of zero-one loss functions.

There are many results bounding the VC-dimension for specific hypothesis spaces (see VAPNIK-CHERVONENKIS DIMENSION OF NEURAL NETWORKS and PAC LEARNING AND NEURAL NETWORKS). Since the result in Equation 8 is uniform, it automatically provides a bound on the generalization error of any algorithm that chooses its hypotheses from some fixed hypothesis space \mathcal{H} .

Data-dependent structural risk minimization. An application of the union bound allows the combination of several VC bounds for different hypothesis spaces $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots \subseteq \mathcal{H}_k \subseteq \mathcal{Y}^{\mathcal{X}}$. This is the idea underlying *structural risk minimization* (SRM): using the combination of VC bounds, an SRM algorithm aims to minimize the bound directly. It is thus applicable to regularized risk-minimization learning algorithms. The bound, however, requires that the series of hypothesis spaces be defined *independently* of the training sample. Hence, we cannot directly use the training sample to control the effective complexity (only implicitly via the resulting training error).

We can relax this assumption by introducing an ordering among the hypotheses to be covered for a given sample $z \in \mathcal{Z}^m$. Such a function, $L: \cup_{i=1}^{\infty} \mathcal{Z}^i \times \mathcal{H} \rightarrow \mathbb{R}$, is called a *luckiness* (see Shawe-Taylor et al., 1998). For each luckiness function it is required that a value measured on the training sample allows one to bound the covering number on the training sample and ghost sample of hypotheses that increase the luckiness. This property is called *probable smoothness* with respect to a function $\omega: \mathbb{R} \times \mathbb{N} \times [0, 1] \rightarrow \mathbb{N}$.

The main result (which is data dependent in the sense used above) for the zero-one loss l_{0-1} reads as follows: *For all luckiness functions L that are probably smooth with respect to ω , with probability at least $1 - \delta$ over the random draw of the training sample*

$z \in \mathcal{Z}^m$, for all hypotheses $h \in \mathcal{H}$ such that $\hat{R}[h, z] = 0$, $R[h] \leq \varepsilon_{\text{DSRM}}(z, h, \omega, L, \delta)$ where

$$\varepsilon_{\text{DSRM}}(z, h, \omega, L, \delta) := \frac{2}{m} \times \underbrace{\left(\log_2 \left(\omega \left(L(h, z), m, \frac{\delta}{2m} \right) \right) \right)}_{\text{effective complexity}} + \log_2 \left(\frac{2m}{\delta} \right). \quad (9)$$

The result can also be stated for non-zero training error and general loss functions shown by Equation 8. Each probably smooth luckiness function defines a data-dependent structuring $\mathcal{H}_1(z) \subseteq \mathcal{H}_2(z) \subseteq \dots \subseteq \mathcal{H}_m(z) \subseteq \mathcal{H}$ of the hypothesis space \mathcal{H} by

$$\mathcal{H}_i(z) := \left\{ h \in \mathcal{H} \mid \omega \left(L(h, z), m, \frac{\delta}{2m} \right) \leq 2^i \right\}.$$

The choice of the luckiness function is not unique; it is best compared to the choice of a prior in a Bayesian analysis (see BAYESIAN METHODS AND NEURAL NETWORKS).

PAC-Bayesian framework. The PAC-Bayesian framework (McAllester, 1998) studies only Bayesian learning algorithms. The main ideas are very similar to the luckiness framework. One of the motivations is to capture an important feature of Bayesian confidence intervals—their width depends on the sample itself and not just its size.

A direct application of the union bound with factors different from $1/n$ leads to the following result: *For all measures $\mathbf{P}_{\mathbf{H}}$ and $\mathbf{P}_{\mathbf{Z}}$, with probability at least $1 - \delta$ over the random draw of the training sample $z \in \mathcal{Z}^m$, for all hypotheses $h \in \mathcal{H}$ such that $\mathbf{P}_{\mathbf{H}}(h) > 0$, $R[h] \leq \varepsilon_{\text{PB}}(z, h, \mathbf{P}_{\mathbf{H}}, \delta)$, where*

$$\varepsilon_{\text{PB}}(z, h, \mathbf{P}_{\mathbf{H}}, \delta) := \hat{R}[h, z] + \underbrace{\sqrt{\frac{1}{2m} \left(\ln \left(\frac{1}{\mathbf{P}_{\mathbf{H}}(h)} \right) + \ln \left(\frac{1}{\delta} \right) \right)}}_{\text{effective complexity}}.$$

If the likelihood function $\mathbf{P}_{\mathbf{Z}|\mathbf{H}=h}((x, y))$ equals $\mathbb{I}_{h(x)=y}$, then the bound maximizer is given by the *maximum a posteriori* estimator $h_{\text{MAP}} := \arg\max_{h \in \mathcal{H}} \mathbf{P}_{\mathbf{H}}(h)$.

Using a tool known as the *quantifier reversal lemma*, it is possible to study the *Gibbs classification strategy*, which uses a randomly drawn hypothesis for each new data point to be classified:

$$\mathcal{A}_{\text{Gibbs}}^H(x) := h(x), \quad h \sim \mathbf{P}_{\mathbf{H}|\mathbf{H} \in \mathcal{H}}$$

The quantifier reversal lemma is a high-probability equivalent of the union bound: *Given n high-probability bounds Y_i (see Equation 7) and any distribution \mathbf{P}_1 over the numbers $\{1, \dots, n\}$,*

$$\forall \alpha \in [0, 1]: \forall \delta \in [0, 1]:$$

$$\mathbf{P}_{\mathbf{Z}^m}(\mathbf{P}_1(Y_1(\mathbf{Z}), \dots, \alpha\delta) \geq 1 - \alpha) \geq 1 - \delta.$$

The proof is very simple and makes use of Markov's inequality. Noticing that for all loss functions $L: \mathcal{Y} \times \mathcal{Y} \rightarrow [0, 1]$,

$$R[\mathcal{A}_{\text{Gibbs}}^H] = \mathbf{E}_{\mathbf{H}|\mathbf{H} \in \mathcal{H}}[R[H]] \leq c \cdot \mathbf{P}_{\mathbf{H}|\mathbf{H} \in \mathcal{H}}(R[H] \leq c) + 1 \cdot \mathbf{P}_{\mathbf{H}|\mathbf{H} \in \mathcal{H}}(R[H] > c)$$

it is possible to prove the following result: *Given a prior measure $\mathbf{P}_{\mathbf{H}}$, with probability at least $1 - \delta$ over the random draw of the training sample $z \in \mathcal{Z}^m$, for all subsets $H \subseteq \mathcal{H}$, the generalization error of the Gibbs classification strategy $\mathcal{A}_{\text{Gibbs}}^H$ satisfies*

$$R[\mathcal{A}_{\text{Gibbs}}^H(z)] \leq \mathbf{E}_{\mathbf{H}|\mathbf{H} \in \mathcal{H}}[\hat{R}[H, z]] + \underbrace{\sqrt{\frac{1}{2m} \left(\ln \left(\frac{1}{\mathbf{P}_{\mathbf{H}}(H)} \right) + \ln \left(\frac{m^2}{\delta} \right) \right)}}_{\text{effective complexity}} + \frac{1}{m}.$$

The effective complexity scales inversely with $\mathbf{P}_H(H)$, which in the case of the likelihood function $\mathbf{P}_{Z|H=h}((x, y)) = \mathbb{I}_{h(x)=y}$ and the Bayesian posterior $\mathbf{P}_{H|Z^m=z}$ equals the *evidence* $\mathbf{E}_H[\mathbf{P}_{Z^m|H=h}(z)]$ (see BAYESIAN METHODS AND NEURAL NETWORKS). The complexity term is minimized if we choose H such that $\mathbf{P}_H(H) = 1$. However, for a small overall bound value, it is also required that the expected empirical risk $\mathbf{E}_{H|H \in \mathcal{H}}[\hat{R}[H, z]]$ be small. It is worth mentioning that the results are still algorithm independent, since they hold not only for the Bayesian posterior but for all hypotheses $h \in \mathcal{H}$ and all subsets $H \subseteq \mathcal{H}$.

Algorithm-Dependent Bounds

We now summarize three distinct but related approaches to the analysis of learning algorithms that utilize particular properties of the algorithm apart from the space \mathcal{H} it draws its hypotheses from.

The compression framework. The compression framework (Floyd and Warmuth, 1995) is based on the idea that a good learning algorithm is able to reconstruct its hypothesis using only a small fraction of the training sample z . It is assumed that the learning algorithm can be written as

$$\mathcal{A}(z) := \mathcal{R}(z_{\mathcal{C}(z)}) \quad (10)$$

where $\mathcal{C}: \cup_{m=1}^{\infty} \mathcal{Z}^m \rightarrow \mathcal{J}$ maps the training sample to indices $\mathbf{i} \in \mathcal{J}$, $\mathcal{J} = \{(i_1, \dots, i_n) \mid n \in \mathbb{N}, i_1 \neq \dots \neq i_n\}$, $z_{\mathbf{i}} := (z_{i_1}, \dots, z_{i_n})$, and $\mathcal{R}: \cup_{m=1}^{\infty} \mathcal{Z}^m \rightarrow \mathcal{Y}^{\mathcal{X}}$ computes the final hypothesis using only the subsample indexed by $\mathcal{C}(z)$. A typical example of such an algorithm is the perceptron learning algorithm (see PERCEPTRONS, ADALINES, AND BACKPROPAGATION), which can reconstruct its hypothesis using only the training patterns on which it needed to update the weight vector.

The mathematical tool needed to study this class of learning algorithms is again the union bound:

$$\begin{aligned} \mathbf{P}_{Z^m}[R[\mathcal{A}(Z)] - \hat{R}[\mathcal{A}(Z), Z] > \varepsilon] \\ \leq \mathbf{P}_{Z^m}[\exists \mathbf{i} \in \mathcal{J}: R[\mathcal{R}(z_{\mathbf{i}})] - \hat{R}[\mathcal{R}(z_{\mathbf{i}}), Z] > \varepsilon] \\ \leq \sum_{\mathbf{i} \in \mathcal{J}} \mathbf{P}_{Z^m}[R[\mathcal{R}(z_{\mathbf{i}})] - \hat{R}[\mathcal{R}(z_{\mathbf{i}}), Z] > \varepsilon] \end{aligned}$$

Interestingly, for any index vector \mathbf{i} the sample $z_{\mathbf{i}}$ is an iid test sample on which the fixed hypothesis $\mathcal{R}(z_{\mathbf{i}})$ is assumed to have a difference in empirical and expected risk of more than ε . Using Equation 6—which holds independent of \mathbf{i} —and the fact that there are no more than $\binom{m}{d} \leq (em/d)^d$, $d = |\mathbf{i}|$, many different index sets for a training sample z of size m , leads to the main result of the compression framework: *For the zero-one loss l_{0-1} and any learning algorithm that can be written in the form of Equation 10, with probability at least $1 - \delta$ over the random draw of the training sample $z \in \mathcal{Z}^m$, $R[\mathcal{A}(z)] \leq \varepsilon_{\text{cr}}(z, l_{0-1}, \delta)$, where for $d = |\mathcal{C}(z)|$*

$$\begin{aligned} \varepsilon_{\text{cr}}(z, d, \delta) := & \frac{m}{m-d} \cdot \hat{R}[\mathcal{A}(z), z] \\ & + \sqrt{\frac{1}{2m-d} \left(d \ln \left(\frac{em}{d} \right) + \ln \left(\frac{m^2}{\delta} \right) \right)}. \quad (11) \end{aligned}$$

effective complexity

A similar result can be stated for general loss functions. Note that this bound is data dependent, since $l_{\mathcal{C}(z)}$ depends both on the learning algorithm \mathcal{A} and on the training sample z .

The compression framework has its roots in the theory of on-line learning (Littlestone, 1988). An *on-line learning algorithm* proceeds in trials. In each trial, the algorithm is presented with a training sample $x_i \in \mathcal{X}$ and makes a prediction $\hat{y} \in \mathcal{Y}$. It then receives the desired output $y_i \in \mathcal{Y}$ and incurs a mistake whenever $\hat{y} \neq y_i$. The performance measure of an on-line learning algorithm

is the number of mistakes it incurs on a training sample z . If the on-line algorithm is *mistake driven*, that is, if it only updates the hypothesis whenever a mistake is incurred, then any mistake bound is also an upper bound on $l_{\mathcal{C}(z)}$. This scheme allows the determination of generalization error bounds for on-line learning algorithms applied in batch mode (see, e.g., Cesa-Bianchi et al., 1997).

The Algorithmic Stability Framework. In the algorithmic stability framework (Bousquet and Elisseeff, 2001), it is assumed that any additional training example has a limited influence on the function learned insofar as the prediction on any possible test point is concerned. Such algorithms are called *uniformly stable* and have the property that for all $i \in \{1, \dots, m\}$:

$$\forall z \in \mathcal{Z}^m : \forall (x, y) \in \mathcal{Z} : |l(\mathcal{A}(z)(x), y) - l(\mathcal{A}(z_{\mathbf{i}})(x), y)| \leq \beta(m),$$

where $z_{\mathbf{i}} := (z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_m)$. The $\beta(\cdot)$ -stability of learning algorithms can be determined if the loss function is *Lipschitz continuous* with (Lipschitz) constant C_l : the difference $l(\hat{y}, \cdot) - l(\tilde{y}, \cdot)$ is bounded from above by $C_l \cdot |\hat{y} - \tilde{y}|$. The ℓ_1 loss l_1 and the ε -insensitive loss l_{ε} are both Lipschitz continuous with the constant $C_l = 1$.

Given a Lipschitz continuous loss function l and a reproducing kernel Hilbert space \mathcal{H} with kernel $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, the class of regularized risk minimization learning algorithms

$$\mathcal{A}_{\text{RRM}}^{\mathcal{H}, \lambda} := \operatorname{argmin}_{h \in \mathcal{H}} (\hat{R}[h, z] + \lambda \|h\|^2)$$

is $\beta(\cdot)$ -stable with $\beta(m) \leq C_l \sup_{x \in \mathcal{X}} k(x, x)/2\lambda m$. Intuitively, the larger $\lambda > 0$, the smaller the influence of the empirical term $\hat{R}[h, z]$, and hence the more stable the learning algorithm (see also GENERALIZATION AND REGULARIZATION IN NONLINEAR LEARNING SYSTEMS).

In order to exploit the $\beta(\cdot)$ -stability of a learning algorithm, a result from the theory of large deviations of functions of random variables known as *McDiarmid's inequality* is used (Devroye and Lugosi, 2001). This inequality asserts that the probability of a deviation of ε between the value of a function f of m iid variables and the expected value of that function decays as $\exp(-\varepsilon^2/mc^2)$, where c is the maximal deviation of the function's value when exchanging one variable. In this sense, McDiarmid's inequality is a generalization of Equation 6 for nonpointwise loss functions. Considering the deviation between the expected risk and the empirical risk of the function learned by \mathcal{A} as a function of m iid random variables leads to the following result: *For any $\beta(\cdot)$ -stable learning algorithm \mathcal{A} and a bounded loss function $l: \mathcal{Y} \times \mathcal{Y} \rightarrow [0, 1]$, with probability at least $1 - \delta$ over the random draw of the training sample $z \in \mathcal{Z}^m$, $R[\mathcal{A}(z)] \leq \varepsilon_{\text{AS}}(z, \beta, \delta)$, where*

$$\begin{aligned} \varepsilon_{\text{AS}}(z, \beta, \delta) := & \hat{R}[\mathcal{A}(z), z] + 2\beta(m) \\ & + \sqrt{\frac{2(4\beta(m) \cdot m + 1)^2 \ln \left(\frac{1}{\delta} \right)}{m}}. \quad (12) \end{aligned}$$

There are three interesting observations to make:

1. In order for the result to be nontrivial, it is required that $\beta(m)$ decay faster than $1/m$. This readily tells us the range of λ values to consider for $\mathcal{A}_{\text{RRM}}^{\mathcal{H}, \lambda}$.
2. The result as stated in Equation 12 is not directly applicable to the zero-one loss l_{0-1} , as the difference in the latter cannot decay at a rate of $1/m$ but is fixed to the values $\{0, 1\}$. Noticing that in practice we often use thresholded real-valued functions $h(\cdot) = \text{sign}(f(\cdot))$ for classification, it is possible to overcome this limitation by bounding the zero-one loss function from above. In particular, if $\mathcal{Y} = \{-1, +1\}$ then

$$l_{\text{margin}}(f(x), y) := \min(\max(0, 1 - yf(x)), 1) \geq l_{0-1}(f(x), y) \\ := \mathbb{1}_{yf(x) \leq 0},$$

that is, any upper bound on the expected risk $\mathbf{E}_{XY}[l_{\text{margin}}(f(X), Y)]$ is by definition an upper bound on $R[h]$ for the zero-one loss l_{0-1} and the associated binary classification function h .

3. The result is data independent, as the stability $\beta(m)$ needs to be known before the training sample arrives. Recent developments in this area aim to overcome this problem by the notion of a stability measured on the given training sample.

The algorithmic luckiness framework. Finally, we present a recently developed algorithm-dependent framework (Herbrich and Williamson, 2002) that builds on ideas of the data-dependent structural risk-minimization framework. The key observation is that the basic lemma is true not only when one considers the maximum deviation between the expected and empirical risk, it is also true for the deviation between the expected and empirical risk of the *one* function learned using a fixed learning algorithm \mathcal{A} . As a consequence, for any double sample $zz' \in \mathcal{Z}^{2m}$ (training sample z and ghost sample z'), one need only consider the set $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{Z}}$ of functions that can be learned by a fixed learning algorithm \mathcal{A} from any subsample of size m . If the learning algorithm under consideration is permutation-invariant, then this set cannot be larger than $|\mathcal{H}| \leq 2^{2m}$, regardless of the loss function considered.

The notion of *luckiness* changes in that it now maps a given learning algorithm \mathcal{A} and a given training sample z to a real value, which effectively measures the extent to which the given data align with an encoded prior belief. In accordance with the data-dependent structural risk-minimization framework, it is required that the measured value of the luckiness on a random training sample z can be used to upper bound the number of subsets of a double sample that will lead to an increase in the luckiness value. This rather technical condition is known as ω -smallness and is best compared to the probable smoothness of luckiness functions earlier. Using the union bound together with the refined basic lemma leads to the following generalization error bound for all loss function $l: \mathcal{Y} \times \mathcal{Y} \rightarrow [0, 1]$: *For all algorithmic luckiness functions L which are ω -small, with probability at least $1 - \delta$ over the random draw of the training sample $z \in \mathcal{Z}^m$, $R[\mathcal{A}(z)] \leq \varepsilon_{\text{AL}}(z, \mathcal{A}, \omega, L, \delta)$*

$$\varepsilon_{\text{AL}}(z, \mathcal{A}, \omega, L, \delta) := \hat{R}[\mathcal{A}(z), z] \\ + \sqrt{\frac{8}{m} \left(\underbrace{\log_2 \left(\omega \left(L(\mathcal{A}, z), \frac{\delta}{2m} \right) \right)}_{\text{effective complexity}} + \log_2 \left(\frac{2m}{\delta} \right) \right)}.$$

The main difference from Equation 9 is in the definition of the luckiness function. In contrast to Equation 9, we can now exploit properties of the learning algorithm in the definition of the ω -smallness. As an easy example, consider the luckiness function $L_0(\mathcal{A}, z) := -|\mathcal{C}(z)|$ for algorithms of the form given by Equation 10. Then, given a value $d = -L_0(\mathcal{A}, z)$ of the luckiness function on any training sample, there cannot be more than $\binom{2m}{d}$ distinct subsets of the training sample and ghost sample, which shows that $\omega(L_0, m, \delta) = \binom{2m}{d}$ is a valid ω function. Note that this example removes the factor $m/(m - d)$ in front of the empirical term in Equation 11 at the cost $2m$ rather than m in the complexity term $d \ln(2em/d)$.

Discussion

Our presentation of the theory of learning and generalization is nonstandard, since we sought to present many, seemingly different approaches. For standard presentations with more details, see Devroye, Györfi, and Lugosi (1996), Vapnik (1998), Anthony and

Bartlett (1999), Herbrich (2002), and Schölkopf and Smola (2002). A fairly comprehensive overview is given in Kulkarni, Lugosi, and Venkatesh (1998). In this article, we have assumed that the genuine interest is in bounds on the generalization error (see Equation 2). It is worth mentioning that another way to quantify generalization behavior of learning algorithms is in terms of bounds on the leave-one-out error (for further details, see Devroye et al., 1996).

Although we would like to use theoretical bounds directly for model selection and model validation, it currently seems that the potential value of these results is to provide insight into the design of learning algorithms. For example, the question of consistency says that covering numbers are the “right” quantities to look at for ERM algorithms.

For other algorithms the situation is less clear, although there are now several variants on classical VC analysis methods that use the same formal learning problem setup. The various bounds we presented ($\varepsilon_{\text{VC}}(z, d_{\mathcal{H}}, \delta)$, $\varepsilon_{\text{DSRM}}(z, h, \omega, L, \delta)$, $\varepsilon_{\text{PB}}(z, h, \mathbf{P}_{\mathcal{H}}, \delta)$, $\varepsilon_{\text{C}}(z, |\mathcal{C}(z)|, \delta)$, $\varepsilon_{\text{AS}}(z, \beta, \delta)$, $\varepsilon_{\text{AL}}(z, \mathcal{A}, \omega, L, \delta)$) were in terms of a range of parameters; we still do not really know what the “right” ones are. Recent work (Mendelson, 2001) has shown the power of alternative geometric approaches to develop certain classes of generalization bounds. We expect that these and other approaches will lead to deeper understanding of the generalization ability of learning machines.

Road Maps: Computability and Complexity; Learning in Artificial Networks

Related Reading: PAC Learning and Neural Networks; Vapnik-Chervonenkis Dimension of Neural Networks

References

- Anthony, M., and Bartlett, P., 1999, *Neural Network Learning: Theoretical Foundations*, Cambridge, Engl.: Cambridge University Press. ♦
- Bousquet, O., and Elisseeff, A., 2001, Algorithmic stability and generalization performance, in *Advances in Neural Information Processing Systems* (T. K. Leen, T. G. Dietterich, and V. Tresp, Eds.), Cambridge, MA: MIT Press, vol. 13, pp. 196–202.
- Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., and Warmuth, M. K., 1997, How to use expert advice, *J. ACM*, 44:427–485.
- Devroye, L., Györfi, L., and Lugosi, G., 1996, *A Probabilistic Theory of Pattern Recognition*, No. 31 in *Applications of Mathematics*, New York: Springer-Verlag, 1996. ♦
- Devroye, L., and Lugosi, G., 2001, *Combinatorial Methods in Density Estimation*, New York: Springer-Verlag.
- Floyd, S., and Warmuth, M., 1995, Sample compression, learnability, and the Vapnik Chervonenkis dimension, *Machine Learn.*, 27:1–36.
- Herbrich, R., 2002, *Learning Kernel Classifiers: Theory and Algorithms*, Cambridge, MA: MIT Press. ♦
- Herbrich, R., and Williamson, R. C., 2002, Algorithmic luckiness, *Machine Learn.*, 3:175–212.
- Kulkarni, S., Lugosi, G., and Venkatesh, S., 1998, Learning pattern classification: A survey, *IEEE Trans. Inform. Theory*, 44:2178–2206. ♦
- Littlestone, N., 1988, Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm, *Machine Learn.*, 2:285–318.
- McAllester, D. A., 1998, Some PAC Bayesian theorems, in *Proceedings of the Annual Conference on Computational Learning Theory*, Madison, WI: ACM Press, pp. 230–234.
- Mendelson, S., 2001, Geometric methods in the analysis of Glivenko-Cantelli classes, in *Proceedings of the 14th Annual Conference on Computational Learning Theory COLT* (D. Helmbold and B. Williamson, Eds.), pp. 256–272.
- Schölkopf, B., and Smola, A. J., 2002, *Learning with Kernels*, Cambridge, MA: MIT Press. ♦
- Shawe-Taylor, J., Bartlett, P. L., Williamson, R. C., and Anthony, M., 1998, Structural risk minimization over data-dependent hierarchies, *IEEE Trans. Inform. Theory*, 44:1926–1940.
- Vapnik, V., 1998, *Statistical Learning Theory*, New York: Wiley. ♦

Learning and Statistical Inference

Shun-ichi Amari

Introduction

Neural networks, real or artificial, have an ability to learn from examples. Learning takes place under stochastic fluctuations, because learning from examples in neural networks is of a stochastic nature in the sense that examples are randomly generated and a network's behavior is intrinsically fluctuating owing to noise. Statistical estimation identifies the mechanism underlying stochastic phenomena and has a long tradition and history of research. Promising approaches to the study of learning problems from the statistical point of view include such concepts as the Fisher information measure, Bayesian loss, and sequential estimation. Information geometry (Amari and Nagaoka, 2000) affords a more advanced approach connecting statistics and neural networks (see NEUROMANIFOLDS AND INFORMATION GEOMETRY).

Nonlinear neurodynamics, learning, and self-organization, among others, are key concepts leading to new developments in statistical science. Consequently, many statisticians have recently become interested in neural network technology (see, e.g., Bishop, 1995; Ripley, 1996). The present article reviews various aspects of neural learning from the statistical point of view.

Neural Networks and Statistical Models

We first describe stochastic behaviors of single neurons, and then those of networks. A mathematical neuron receives a number of input signals $\{x_1, \dots, x_n\}$, summarized in an input vector $\mathbf{x} = (x_0, x_1, \dots, x_n)$, and emits an output z . Here, $x_0 = 1$ is added to set the bias or the threshold term. The neuron calculates the weighted sum of inputs

$$u = \mathbf{w} \cdot \mathbf{x} = \sum w_i x_i \quad (1)$$

where w_i are the synaptic efficacies or connection weights. The output z is determined stochastically, depending on u . When z takes analogue values, representing the firing rate of a neuron, it is given by a sigmoidal nonlinear function $f(u)$ of u , disturbed by additive noise,

$$z = f(u) + n \quad (2)$$

Since the noise n is random, let $r(n)$ be its probability density function. The expectation of n is assumed to be zero. When input \mathbf{x} is applied, z is determined stochastically, and its conditional probability density is given by

$$p(z|\mathbf{x}) = r\{z - f(\mathbf{w} \cdot \mathbf{x})\}$$

The expectation of z is

$$E[z|\mathbf{x}] = \int zp(z|\mathbf{x})dz = f(\mathbf{w} \cdot \mathbf{x}) \quad (3)$$

where $E[z|\mathbf{x}]$ denotes the conditional expectation of z under the condition that the input is \mathbf{x} (White, 1989). That is, the output z is fluctuating around $f(\mathbf{w} \cdot \mathbf{x})$.

In the binary neuron model, z takes on the binary values 0 and 1, representing nonfiring and firing of a neuron, respectively. A widely used stochastic neuron model is specified by the following probability:

$$p(z|\mathbf{x}) = \frac{\exp\{\beta z \mathbf{w} \cdot \mathbf{x}\}}{1 + \exp\{\beta \mathbf{w} \cdot \mathbf{x}\}}, \quad z = 0, 1 \quad (4)$$

Here, $\beta > 0$ is a constant, and, when β is large, the probability of firing ($z = 1$) is very large for $\mathbf{w} \cdot \mathbf{x} > 0$. When β is small, the probability of firing is almost fifty-fifty, that is, randomness dominates. For this reason, β is called the "inverse temperature," in analogy with statistical physics. In this case,

$$E[z|\mathbf{x}] = f(\mathbf{w} \cdot \mathbf{x})$$

with the sigmoidal function

$$f(u) = \frac{\exp\{\beta u\}}{1 + \exp\{\beta u\}}$$

A multilayer perceptron is a neural network with feedforward connections (see PERCEPTRONS, ADALINES, AND BACKPROPAGATION). It consists of input neurons, hidden neurons, and output neurons. It receives an input \mathbf{x} from the input neurons and emits an output vector signal $\mathbf{z} = (z_1, \dots, z_m)$ from the m output neurons. Even when the hidden neurons behave deterministically, noise is added to the output neurons. Hence, the behavior of a network is stochastic and represented by the conditional probability distribution $p(\mathbf{z}|\mathbf{x})$. Since the network includes a large number of modifiable parameters (synaptic efficacies and thresholds of component neurons), we summarize all of them in a vector $\mathbf{w} = (w_1, \dots, w_k)$. The conditional probability is expressed in the form $p(\mathbf{z}|\mathbf{x}; \mathbf{w})$, showing that it depends on the parameters \mathbf{w} . The conditional expectation of \mathbf{z} is represented by $\mathbf{f}(\mathbf{x}, \mathbf{w}) = E[\mathbf{z}|\mathbf{x}; \mathbf{w}]$.

Let $(\mathbf{x}_t, \mathbf{z}_t)$, $t = 1, \dots, T$, be observed examples of input-output pairs. Learning as well as statistical estimation is carried out based on the training set

$$D_T = \{(\mathbf{x}_1, \mathbf{z}_1), \dots, (\mathbf{x}_T, \mathbf{z}_T)\} \quad (5)$$

to identify the true \mathbf{w} from which the data are generated. In many cases, \mathbf{x}_t are generated independently, subject to an unknown probability distribution $q(\mathbf{x})$, and \mathbf{z}_t is the desired output provided from the "teacher," which has the true parameter \mathbf{w} . In the case of self-organization or unsupervised learning, the desired outputs \mathbf{z}_t are missing, so that D_T consists of only \mathbf{x}_t s. Statistical estimation uses all the data D_T to estimate the true parameter, whereas on-line learning assumes that data come in one by one. The current candidate for $\hat{\mathbf{w}}$ is modified by a new input-output pair when it arrives. The old data are discarded and are not used again. Hence, on-line learning is a procedure to modify the network parameters \mathbf{w} sequentially, based on the series of training data D_T , such that the trained network performs sufficiently well to simulate the true network from which the training data are obtained.

Different from on-line learning, batch learning uses all the data D_T , modifies the current estimate $\hat{\mathbf{w}}$, and repeats the procedure until it converges. Hence, batch learning is an interactive estimation procedure.

Information Measures

Data $(\mathbf{x}_t, \mathbf{z}_t)$ include information to identify the probability distribution $p(\mathbf{z}|\mathbf{x}; \mathbf{w})$ or its parameters \mathbf{w} . How can one measure it? R. A. Fisher argued this problem, and proposed the Fisher information

measure, which is defined soon. This is different from the Shannon information measure, which measures uncertainty by using the entropy of random variables. However, there are certain connections between them. We briefly summarize these information measures and their relations.

Let $p(x, \mathbf{w})$ be the probability density function of a random variable x parameterized by \mathbf{w} . In the case of neural networks, x represents a pair (x, \mathbf{z}) . The family $M = \{p(x, \mathbf{w})\}$ is called a statistical model. Let x_1, \dots, x_T be T independent observations, and let $\hat{\mathbf{w}}$ be an estimator of the true \mathbf{w} . How much information is included in the training data concerning \mathbf{w} ? The Fisher information matrix $G = (g_{ij})$ represents such information. It is defined in component form by

$$g_{ij}(\mathbf{w}) = E \left[\frac{\partial}{\partial w_i} \log p(x, \mathbf{w}) \frac{\partial}{\partial w_j} \log p(x, \mathbf{w}) \right] \quad (6)$$

where E denotes the expectation with respect to $p(x, \mathbf{w})$. In vector matrix notation, this can be rewritten as

$$G(\mathbf{w}) = E[\nabla \log p(x, \mathbf{w}) \nabla \log p(x, \mathbf{w})^T] \quad (7)$$

where $\nabla f(\mathbf{w})$ is the gradient (column) vector of f

$$\nabla f = \left[\frac{\partial f}{\partial w_1}, \dots, \frac{\partial f}{\partial w_n} \right]^T \quad (8)$$

T denoting transposition.

When we measure the accuracy of an estimator $\hat{\mathbf{w}}$ by its error covariance matrix $V = (v_{ij})$,

$$v_{ij} = E[(\hat{w}_i - w_i)(\hat{w}_j - w_j)] \quad (9)$$

the Crámer-Rao theorem shows that

$$V \geq \frac{1}{T} G^{-1} \quad (10)$$

implying that the error covariance matrix is at best as small as the inverse of the Fisher information measure divided by the number T of observations. One of the most popular estimates is the maximum likelihood estimator $\hat{\mathbf{w}}_{mle}$, which maximizes the likelihood $p(x_1, \mathbf{w}) \dots p(x_T, \mathbf{w})$, that is, the probability of obtaining the observed data x_1, \dots, x_T when the underlying distribution is $p(x, \mathbf{w})$. This estimate attains the bound asymptotically, that is, $V \approx G^{-1}/T$ for large T . When G is large, the error $V = G^{-1}/T$ is small, so it is natural to regard G as the measure of information. This topic is covered in standard textbooks (e.g., Cox and Hinkley, 1974).

Another important quantity is the divergence measure between two probability distributions $p(x)$ and $q(x)$. How different are they? The Kullback-Leibler divergence, or the relative entropy, defined by

$$KL(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \quad (11)$$

is a frequently used measure in statistics and information theory (Cover and Thomas, 1991). It is related to both Shannon and Fisher information measures, as follows. Let $p_{XY}(x, y)$ be the joint probability of X and Y , and let $p_X(x)$ and $p_Y(y)$ be their marginal distributions. Then the mutual information between X and Y , defined by Shannon, is

$$I(X : Y) = KL\{p_{XY}(x, y)||p_X(x)p_Y(y)\} \quad (12)$$

When \mathbf{w} and $\mathbf{w} + d\mathbf{w}$ are infinitesimally close, the divergence between $p(x, \mathbf{w})$ and $p(x, \mathbf{w} + d\mathbf{w})$ is given by

$$KL(p(x, \mathbf{w})||p(x, \mathbf{w} + d\mathbf{w})) = \frac{1}{2} \sum g_{ij}(\mathbf{w}) dw_i dw_j \quad (13)$$

so that the KL divergence is measured locally by the quadratic form of the Fisher information measure.

The behavior of a parameterized network is given by the conditional probability $p(\mathbf{z}|\mathbf{x}; \mathbf{w})$. When \mathbf{x} is generated subject to $q(\mathbf{x})$, the input-output joint distribution is $p(\mathbf{x}, \mathbf{z}; \mathbf{w}) = q(\mathbf{x})p(\mathbf{z}|\mathbf{x}; \mathbf{w})$. Consider the set $M = \{p(\mathbf{x}, \mathbf{z}; \mathbf{w})\}$ of all such probability distributions related to neural networks specified by parameter \mathbf{w} . This is identified with the space of parameters \mathbf{w} .

Given data D_T , the maximum likelihood estimator $\hat{\mathbf{w}}_{mle}$ is the one that maximizes the likelihood of the data, $P(D_T; \mathbf{w}) = \prod p(\mathbf{x}_i, \mathbf{z}_i; \mathbf{w})$ or its logarithm

$$\log P(D_T; \mathbf{w}) = \sum \log q(\mathbf{x}_i) + \sum \log p(\mathbf{z}_i|\mathbf{x}_i; \mathbf{w})$$

The $\hat{\mathbf{w}}_{mle}$ is a consistent estimator in the sense that $\hat{\mathbf{w}}_{mle}$ converges to the true parameter \mathbf{w}_0 from which the training data are derived, or its optimal approximation in M . More precisely, $\hat{\mathbf{w}}_{mle}$ is asymptotically normally distributed with mean \mathbf{w}_0 and covariance matrix G^{-1}/T (Cox and Hinkley, 1974), where G is the Fisher information matrix, defined in this case by

$$G = E[\nabla \log p(\mathbf{z}|\mathbf{x}, \mathbf{w}) \{\nabla \log p(\mathbf{z}|\mathbf{x}, \mathbf{w})\}^T] \quad (14)$$

As T becomes large, the error term G^{-1}/T converges to 0.

Since the Fisher information matrix plays a fundamental role in the accuracy of estimation and learning, some examples are shown here. In the first case, the output \mathbf{z} is a noise-contaminated version of $\mathbf{f}(\mathbf{x}; \mathbf{w})$, which is the output of a multilayer perceptron with analogue activation function:

$$\mathbf{z} = \mathbf{f}(\mathbf{x}; \mathbf{w}) + \mathbf{n}$$

where \mathbf{n} is Gaussian noise with mean 0 and covariance matrix $\sigma^2 I$, I being the identity matrix. From the statistical viewpoint, this is the nonlinear regression problem of observed data D_T to the nonlinear model $\mathbf{f}(\mathbf{x}, \mathbf{w})$. By simple calculations, the Fisher information matrix is given by

$$G = \frac{1}{\sigma^2} E[\nabla \mathbf{f}(\mathbf{x}; \mathbf{w}) \{\nabla \mathbf{f}(\mathbf{x}; \mathbf{w})\}^T] \quad (15)$$

where the expectation is taken over the distribution $q(\mathbf{x})$. This shows that the Fisher information tends to infinity, and the estimation error tends to 0, as the noise term σ^2 becomes 0.

In the case of binary neurons given by Equation 4, the Fisher information is calculated as

$$G = \frac{2\beta e^{\beta f}}{1 + e^{\beta f}} \nabla \nabla f + \frac{\beta^2 e^{\beta f}}{(1 + e^{\beta f})^2} \nabla f (\nabla f)^T \quad (16)$$

It should also be noted that G tends to ∞ as the temperature term β^{-1} tends to 0.

Stochastic Descent On-Line Learning and the Bayesian Standpoint

The aim of learning is not to estimate \mathbf{w} but to obtain a network with good information-processing performance. The Bayesian standpoint (Berger, 1985) assumes that we have prior knowledge concerning the probability of \mathbf{w} to be estimated. After observing data, it is modified to the posterior distribution of \mathbf{w} . It also suggests using the notion of a risk or loss function that is to be minimized through learning.

Let $l(\mathbf{x}, \mathbf{z}, \mathbf{w})$ be a loss when an input signal \mathbf{x} is processed by a network of parameter \mathbf{w} , where \mathbf{z} is the desired output given by the teacher. A simplest example is the squared error,

$$l(\mathbf{x}, \mathbf{z}, \mathbf{w}) = \frac{1}{2} \|\mathbf{z} - \mathbf{z}(\mathbf{x}, \mathbf{w})\|^2 \quad (17)$$

where $\mathbf{z}(\mathbf{x}, \mathbf{w})$ is the output from the network specified by \mathbf{w} . In the case when $\mathbf{z}(\mathbf{x}, \mathbf{w})$ is given by

$$\mathbf{z}(\mathbf{x}, \mathbf{w}) = \mathbf{f}(\mathbf{x}, \mathbf{w}) + \mathbf{n} \quad (18)$$

where \mathbf{n} is a Gaussian noise subject to $N(0, I)$, this $l(\mathbf{x}, \mathbf{z}; \mathbf{w})$ coincides with the negative of the log likelihood $l(\mathbf{x}, \mathbf{z}, \mathbf{w}) = -\log p(\mathbf{x}, \mathbf{z}; \mathbf{w})$ except for a constant term $-\log q(\mathbf{x})$ not depending on \mathbf{w} . Hence, minimizing the loss is equivalent to maximizing the likelihood in this case.

Sometimes, a function $F(\mathbf{w})$ of \mathbf{w} is added to the loss in order to penalize a complex network. The function $F(\mathbf{w})$, called the regularization term, takes a large value when the network with parameter \mathbf{w} is complex (Poggio, Torre, and Koch, 1985). One typical example is

$$F(\mathbf{w}) = \sum w_i^2$$

which penalizes large w_i . Another one is

$$F(\mathbf{w}) = \sum \frac{w_i^2}{1 + w_i^2} \quad (19)$$

which penalizes the number of nonzero w_i s when w_i are large.

The risk function $R(\mathbf{w})$ is the expected loss

$$R(\mathbf{w}) = E[l(\mathbf{x}, \mathbf{z}; \mathbf{w})] \quad (20)$$

where expectation is taken with respect to the distribution given by the teacher. When the teacher generates \mathbf{z} by using the network with parameter \mathbf{w}_0 , the distribution of (\mathbf{x}, \mathbf{z}) is $q(\mathbf{x})p(\mathbf{z}|\mathbf{x}; \mathbf{w}_0)$. The function $R(\mathbf{w})$ is called the generalization error, since the behavior of the network specified by \mathbf{w} is evaluated by the expectation with respect to a new example (\mathbf{x}, \mathbf{z}) subject to the same distribution. The best network is supposed to be the one that minimizes $R(\mathbf{w})$.

However, we do not know the risk function $R(\mathbf{w})$, since the true \mathbf{w}_0 which is used in Equation 20 for taking expectation is unknown. Instead, we have a training set D_T generated from the true distribution. This gives us the empirical risk function

$$R_{\text{train}}(\mathbf{w}) = \frac{1}{T} \sum_{i=1}^T l(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) \quad (21)$$

which is an estimate of $R(\mathbf{w})$ by using the training data themselves. It is called the training error, and it converges to $R(\mathbf{w})$ as T tends to infinity.

The penalty term may be derived from the Bayesian standpoint. Assume that we have prior knowledge of \mathbf{w} such that \mathbf{w} is subject to a prior distribution $p_{pr}(\mathbf{w})$. Then, the joint probability of $(\mathbf{x}, \mathbf{z}, \mathbf{w})$ is given by

$$p_{pr}(\mathbf{w})q(\mathbf{x})p(\mathbf{z}|\mathbf{x}, \mathbf{w}) \quad (22)$$

Therefore, the joint probability of data D_T and the parameter \mathbf{w} is

$$p_{pr}(\mathbf{w}) \prod p(\mathbf{x}_i, \mathbf{z}_i; \mathbf{w}) \quad (23)$$

When data D_T are obtained, the posterior probability of \mathbf{w} is

$$p_{\text{post}}(\mathbf{w}|D_T) = \frac{p_{pr}(\mathbf{w})P(D_T|\mathbf{w})}{P(D_T)} \quad (24)$$

A good candidate for \mathbf{w} suggested by data D_T is the maximum posterior estimate that maximizes $p_{\text{post}}(\mathbf{w}|D_T)$. This reduces to the maximum likelihood estimator when $p_{pr}(\mathbf{w})$ is uniform. Minimizing $-\log p_{\text{post}}(\mathbf{w}|D_T)$ is equivalent to minimizing

$$R_{\text{train}} = \frac{1}{T} \sum r(\mathbf{x}_i, \mathbf{z}_i; \mathbf{w}) \quad (25)$$

where

$$l(\mathbf{x}, \mathbf{z}; \mathbf{w}) = -\log p(\mathbf{x}, \mathbf{z}, \mathbf{w}) - \frac{1}{T} \log p_{pr}(\mathbf{w}) \quad (26)$$

In this case, the penalty term is given by $-(1/T) \log p_{pr}(\mathbf{w})$.

Learning is a procedure to estimate the optimal \mathbf{w} based on a given data set D_T . Batch learning uses all the stored data D_T repeatedly. The batch gradient learning method modifies the current estimate \mathbf{w}_t at time t into $\mathbf{w}_{t+1} = \mathbf{w}_t + \Delta \mathbf{w}_t$ by

$$\Delta \mathbf{w}_t = -\eta_t \nabla R_{\text{train}}(\mathbf{w}_t) \quad (27)$$

Here, ∇R_{train} is the gradient vector of R_{train} whose components are $(\partial/\partial w_i)R_{\text{train}}(\mathbf{w})$, and η_t is a learning rate. This is called the *gradient descent method*, because \mathbf{w}_t is modified in the direction of the gradient of R_{train} . For an adequate sequence η_t , \mathbf{w}_t converges to the minimizer of $R_{\text{train}}(\mathbf{w})$. Hence, when l is the negative log likelihood, \mathbf{w}_t converges to the maximum likelihood estimator.

On-line learning, on the other hand, cannot use all the data D_T at once. Instead, one input-output training pair $(\mathbf{x}_t, \mathbf{z}_t)$ becomes available at time t . The current estimator \mathbf{w}_t is updated to \mathbf{w}_{t+1} by using this. The data $(\mathbf{x}_t, \mathbf{z}_t)$ are then discarded so that they cannot be used again. On-line learning provides a simple learning rule, and its behavior is flexible even when the behavior of the teacher is changing.

The stochastic descent on-line learning procedure updates \mathbf{w}_t by

$$\Delta \mathbf{w}_t = -\eta_t \nabla l(\mathbf{x}_t, \mathbf{z}_t; \mathbf{w}_t) \quad (28)$$

This simple stochastic descent learning was proposed by Amari (1967) and was later called the *generalized delta rule* (Rumelhart, Hinton, and Williams, 1986). When it is applied to the multilayer perceptron, the calculation of ∇l is performed through error propagation in the backward direction. This is a nice interpretation (Rumelhart et al., 1986), and the algorithm is called the *error back-propagation method* (see BACKPROPAGATION: GENERAL PRINCIPLES).

For a deterministic neural network without stochastic fluctuations, the squared error (Equation 17) is not related to the negative log probability. This corresponds to the case with $\sigma^2 \rightarrow 0$. However, the gradient descent learning algorithm does not directly include σ^2 . This method was proposed in the deterministic case, and its relation to statistics became clear only later (White, 1989).

The backpropagation learning method has been widely used and is one of the standard engineering tools. However, it is not free from a number of flaws. It is not Fisher efficient; that is, the estimation error of \mathbf{w}_t does not satisfy the Cramér-Rao bound, which the maximum likelihood estimator does. There exist a large number of local minima in R_{train} so that \mathbf{w}_t converges to one of the local minima, which might be different from the global minima. Further, its convergence has been found very slow because of “plateaus.”

It is known that the error decreases quickly in the beginning of learning, but its rate of decrease becomes extremely slow. After surprisingly many steps, the error again decreases rapidly. This is understood as showing that \mathbf{w}_t is trapped in a plateau. A *plateau* is a critical point of $R_{\text{train}}(\mathbf{w})$, but it is not a local minimum. However, it takes a long time for learning to escape from it. The statistical physical method makes clear that plateaus exist because of the “symmetry” existing in the hidden units in the multilayer perceptron (Saad and Solla, 1995).

Various acceleration methods for the backpropagation learning rule have been proposed, but they cannot eliminate plateaus. The natural gradient method (Amari, 1998), based on the Riemannian structure of a neuromanifold, not only eliminates plateaus but is Fisher efficient, as will be shown in the next section.

We should mention that this type of learning is applicable to any parameterized family $p(\mathbf{z}|\mathbf{x}, \mathbf{w})$ of stochastic behaviors as well as to deterministic behaviors. The idea is also used in the self-organization scheme, where there are no desired outputs \mathbf{z}_{true} but the network modifies its structure depending only on the input data $D_T = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$. For example, if the loss function is put equal to

$$l(\mathbf{x}, \mathbf{w}) = a|\mathbf{w}|^2 - \frac{1}{2} \mathbf{w} \cdot \mathbf{x}^2 \quad (29)$$

for a single neuron, the connection weight vector \mathbf{w} of the neuron converges to the principal eigenvector of the correlation matrix

$$V = \int q(\mathbf{x}) \mathbf{x} \mathbf{x}^T d\mathbf{x} \quad (30)$$

of the input signals, if $|\mathbf{w}|$ is normalized. This fact was pointed out in more general perspectives on neural learning by Amari (1977) and studied in detail by Oja (1982) (see PRINCIPAL COMPONENT ANALYSIS).

Another example of self-organization is Kohonen's learning vector quantizer (see LEARNING VECTOR QUANTIZATION). Let us consider a set of k neurons whose connection weights are $\mathbf{w}_1, \dots, \mathbf{w}_k$. The neurons receive a common input \mathbf{x} and calculate the distance $|\mathbf{w}_i - \mathbf{x}|$. The neuron whose weight \mathbf{w}_i is closest to \mathbf{x} is called the winner, and its output is assigned $z_k = 1$, while all the other $z_j = 0$. This is called the *winner-take-all rule*. Let us put

$$l(\mathbf{x}, \mathbf{w}) = \frac{1}{2} \min_i |\mathbf{w}_i - \mathbf{x}|^2$$

where $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_k)$. Then the risk is given by

$$R(\mathbf{w}) = \frac{1}{2} \int \min_i |\mathbf{w}_i - \mathbf{x}|^2 q(\mathbf{x}) d\mathbf{x} \quad (31)$$

The learning rule (Equation 28) in this cases leads to the Kohonen learning vector quantizer.

Natural Gradient Learning

The natural (Riemannian) gradient learning method based on the Fisher information matrix was proposed by Amari (1998) to eliminate plateau phenomena and to accelerate convergence. The gradient $-\nabla l(\mathbf{w})$ is usually believed to be the steepest descent direction of a scalar function $l(\mathbf{w})$. However, in a parameter space M of multilayer perceptrons, it is natural to define

$$\tilde{\nabla} l(\mathbf{w}) = G^{-1}(\mathbf{w}) \nabla l(\mathbf{w}) \quad (32)$$

Here, G^{-1} is the inverse of the Fisher information matrix $G = (g_{ij})$. This is called the natural gradient.

The natural gradient learning algorithm was proposed as the steepest descent method in a Riemannian space where G is the metric,

$$\Delta \mathbf{w}_t = -\eta_t G^{-1}(\mathbf{w}_t) \nabla l(\mathbf{x}_t, \mathbf{z}_t; \mathbf{w}_t) \quad (33)$$

It has two remarkable properties. First, it avoids plateaus, so that it has an optimal dynamic rate of convergence. Second, the accuracy of estimator \mathbf{w}_t by the natural gradient is Fisher efficient when $\eta_t = 1/t$, that is, it has the same asymptotic property as the best batch estimator. The natural gradient method is also successfully applied to INDEPENDENT COMPONENT ANALYSIS (q.v.). The natural gradient method does not eliminate local minima. There are a number of techniques to overcome this problem (see NEUROMANIFOLDS AND INFORMATION GEOMETRY).

Learning Curves and Generalization Errors

A learning curve shows how quickly a learning network improves its behavior evaluated by the generalization error. This is related to the dynamical behavior of neural learning and the complexity of neural networks. We analyze this important problem in this section.

The stochastic approximation guarantees that $\hat{\mathbf{w}}_t$ converges to the optimal parameter \mathbf{w}_0 with probability 1, when the learning

constant η_t tends to 0 in an adequate speed. However, when η_t is too small, learning becomes ineffective. What learning is for, in many cases, is to adjust the network parameters in a changing environment. In this case, η_t should be kept at least to a small constant ε . When η_t is put equal to a small constant ε , the dynamical behavior of $\hat{\mathbf{w}}_t$ is studied in an old paper (Amari, 1967), where the stochastic descent learning rule was proposed for the multilayer perceptron from the Bayesian standpoint.

Let us analyze the accuracy of estimator \mathbf{w}_t obtained by Equation 28. We assume the case that the initial value \mathbf{w}_1 is in a neighborhood of the (local) optimal value \mathbf{w}_0 such that it converges to \mathbf{w}_0 . Let us define two matrices, the Hessian and the covariance of the gradient of l ,

$$A = E[\nabla \nabla l(\mathbf{w}_0)] \quad (34)$$

$$B = E[\nabla l(\nabla l)^T] \quad (35)$$

The expected value of $\hat{\mathbf{w}}_t$ converges to \mathbf{w}_0 exponentially, and the covariance of the error $\hat{\mathbf{w}}_t - \mathbf{w}_0$ also converges exponentially to εV , where V is a matrix obtained from A and B (Amari, 1967). The dynamical behavior of \mathbf{w}_t was also studied when the environment, that is, the optimal \mathbf{w}_0 , is periodically changing over time slowly.

Since the behavior of the net is evaluated by $R(\hat{\mathbf{w}}_t)$, but not directly by $\hat{\mathbf{w}}_t$, it is important to know how fast $R(\hat{\mathbf{w}}_t)$ approaches its optimum value. Here, $R(\hat{\mathbf{w}}_t)$ is the expectation of the loss $l(\mathbf{x}, \mathbf{z}; \hat{\mathbf{w}}_t)$ with respect to a new example pair (\mathbf{x}, \mathbf{z}) . This is the generalization error, which evaluates the behavior of the net by a new example (\mathbf{x}, \mathbf{z}) that is not included in the training set D_T . This is different from the training error in Equation 25, which is an evaluation of $\hat{\mathbf{w}}_t$ based on the training data D_T . We can calculate the latter, but it is difficult to know the generalization error $R(\hat{\mathbf{w}}_t)$ because we do not know the function $R(\mathbf{w})$ itself. If we know the relation between R_{train} and R , we can then evaluate R through R_{train} .

A standard technique of asymptotic statistical inference (Cox and Hinkley, 1974) can be applied to this problem. Let us fix the training data D_t , and let $\hat{\mathbf{w}}_t$ now be the best estimator obtained therefrom, where t is assumed to be a large number. It maximizes R_{train} , so that it satisfies

$$0 = \nabla R_{\text{train}}(\hat{\mathbf{w}}_t)$$

On the other hand, the optimal \mathbf{w}_0 satisfies $\nabla R(\mathbf{w}_0) = 0$. From this we have, by mathematical analysis,

$$E[R(\hat{\mathbf{w}}_t)] = R(\mathbf{w}_0) + \frac{1}{2t} \text{tr}(B^{-1}A) \quad (36)$$

Mathematical analysis also gives

$$E[R_{\text{train}}(\hat{\mathbf{w}}_t)] = R(\mathbf{w}_0) - \frac{1}{2t} \text{tr}(B^{-1}A) \quad (37)$$

The relation between the training error and generalization error $E[R(\hat{\mathbf{w}}_t)]$ is given from Equations 36 and 37 by

$$E[R(\hat{\mathbf{w}}_t)] = E[R_{\text{train}}(\hat{\mathbf{w}}_t)] + \frac{1}{t} \text{tr}(B^{-1}A)$$

The term $\text{tr}(B^{-1}A)$ shows the difference between the training error and the generalization error. When this term is large, R_{train} is much smaller than R because of overfitting. That is, the estimated network fits too much to the observed examples, but it fails to capture the mechanism generating the examples. If a complex network is used, the observed data D_t are easily overfitted. Therefore, this term represents the penalty due to the use of a complex network. This is a generalization of the Akaike information criterion, which is widely used in statistical inference. When the loss is the negative log likelihood and the model includes the true one, both A and B are equal to the Fisher information matrix, so that we have an evaluation of

complexity by the number of modifiable parameters, $\text{tr}(B^{-1}A) = \text{tr}(I) =$ the number of modifiable parameters. This is the result first obtained by Akaike for selecting a reasonable model. This is universal in the sense that the complexity (overfitting factor) depends only on the number of parameters, independently of the architecture of the network. Universal properties of this type are more or less known in various situations concerning learning machines (Amari and Murata, 1993). One more remarkable fact is the t^{-1} convergence of the learning error in learning curves. As the number t of training examples increases, the generalization decreases in proportion to $1/t$. This was first remarked on by T. Cover in his doctoral thesis in 1964. This result is proved in Amari and Murata in a general situation. Learning attracts researchers from algorithmic, information-theoretic, and physics points of view because of its general and flexible ability for information processing related to the human ability.

Discussion

We have focused on neural network learning from the viewpoint of statistical inference. Learning is a sequential estimation from the statistical point of view. The accuracy of learning was shown in terms of the Fisher information measure. We are interested in the dynamical behaviors of a learning network under a general loss criterion. The natural gradient learning algorithm was introduced in this respect. The behavior of learning curves was shown, and the complexity of a neural network was introduced to elucidate the discrepancy between the training and generalization errors.

Road Map: Learning in Artificial Networks

Related Reading: Bayesian Methods and Neural Networks; Data Clustering and Learning; Generalization and Regularization in Nonlinear Learning Systems; Graphical Models: Probabilistic Inference; Independent

Component Analysis; Perceptrons, Adalines, and Backpropagation; Stochastic Approximation and Efficient Learning

References

- Amari, S., 1967, Theory of adaptive pattern classifiers, *IEEE Trans. Electron. Comput.*, 16:299–307.
- Amari, S., 1977, Neural theory of association and concept-formation, *Biol. Cybern.*, 26:175–185.
- Amari, S., 1998, Natural gradient works efficiently in learning, *Neural Group*, 10:251–276.
- Amari, S., and Murata, N., 1993, Statistical theory of learning curves under entropic loss, *Neural Comp.*, 5:140–153.
- Amari, S., and Nagaoka, H., 2000, *Introduction to Information Geometry*, London: AMS and Oxford University Press. ♦
- Berger, J. O., 1985, *Statistical Decision Theory and Bayesian Analysis*, 2nd ed., New York: Springer-Verlag. ♦
- Bishop, C. M., 1995, *Neural Networks for Pattern Recognition*, Oxford: Clarendon Press.
- Cover, T. M., and Thomas, J. A., 1991, *Elements of Information Theory*, New York: Wiley.
- Cox, D. R., and Hinkley, D. V., 1974, *Theoretical Statistics*, London: Chapman and Hall. ♦
- Oja, E., 1982, A simplified neuron model as a principal component analyzer, *J. Math. Biol.*, 15:267–273.
- Poggio, T., Torre, V., and Koch, C., 1985, Computational vision and regularization theory, *Nature*, 317:314–319.
- Ripley, B. D., 1996, *Pattern Recognition and Neural Networks*, Cambridge, Engl.: Cambridge University Press. ♦
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, Learning internal representation by error backpropagation, in *Parallel Distributed Processing* (D. E. Rumelhart, J. L. McClelland, and PDP Group, Eds.), Cambridge, MA: MIT Press.
- Saad, D., and Solla, S. A., 1995, On-line learning in soft committee machines, *Phys. Rev. E*, 52:4225–4243.
- White, H., 1989, Learning in artificial neural networks: A statistical perspective, *Neural Comp.*, 1:425–464. ♦

Learning Network Topology

Chuck P. Lam and David G. Stork

Introduction

To create a neural network, a designer typically fixes a network topology and uses training data to tune its parameters such as connection weights. The designer, however, often does not have enough knowledge to specify the ideal topology. It is thus desirable to learn the topology from training data as well.

Traditional learning can be viewed as a search in the space of parameters. This article looks at topology learning as a search in the space of topologies. In particular, we define in general terms a measure that quantifies the “goodness” of a topology and some search strategies over the space of topologies to find the best one. This framework is applied to learning the topologies of the two forms of networks that have found use in pattern recognition, feed-forward neural networks, and BAYESIAN NETWORKS (q.v.).

Objectives of Learning

Learning is the automated process of modifying a system to improve its performance on its given task. When learning is restricted to modifying a network’s weights, performance is generally measured as just the network’s ability to model a set of input samples. However, this performance measure alone is problematic when one can choose arbitrary topologies, since a large enough network can

model a given data set arbitrarily well, leading to overfitting. Therefore, for topology learning, a bias is added to prefer smaller models. It is often found that this bias produces a network that has better generalization and is more interpretable.

A principled approach to topology selection is to find the most probable topology T given training data D (Cheeseman, 1990). Applying Bayes’s rule, one is thus searching for the topology that maximizes $p(T|D) \propto p(T)p(D|T)$. Here $p(T)$ is a *prior* probability that expresses one’s belief about the probabilities of various topologies to be a “good” topology. One can express one’s *bias* for smaller topologies through $p(T)$. Viewed another way, $p(T)$ allows one to impose a *penalty* on larger topologies. Next, $p(D|T)$ is interpreted as how well a topology models the data, and this measure generally has a higher maximum as T gets bigger and more complex. Most topology learning algorithms follow this framework at least in spirit, somehow incorporating the trade-off between a topology’s fit to data, measured as the output error, and the desirability of that topology, measured as some penalty term. This framework expresses a simple, explicit trade-off between how well a model predicts the data and our belief in the goodness of that model. As we shall see subsequently, learning the topology of Bayes nets often follow this framework explicitly (Buntine, 1996).

Using a measure that takes into account a topology’s fit to data and one’s preference for smaller networks, one can search for the

topology with the highest value. Except for certain unique cases, a greedy search is used over this topology space. For feedforward neural networks there are two dominant approaches to this searching, growing, and pruning (Figure 1). In *growing*, a small network is created initially, and nodes are added through learning. The search space is constrained by the particular method and its way of adding nodes. In some cases the search space is limited to the set of neural networks with a single hidden layer. Conversely, *pruning* starts with a sufficiently large network specified by the designer and proceeds to eliminate “unimportant” weights or nodes. The search space is thus limited to pruned versions of the initial network. In Bayesian networks, all the nodes of the network are given and set, and one searches for a topology by adding or deleting links.

Growing

In network growing, nodes are added and the network is retrained iteratively until it reaches satisfactory performance. Adding nodes to a network can always improve the network’s fit to the training data. The penalty/bias to prevent overly large networks can be expressed as a stopping criterion. One stops adding nodes once the network has satisfactory performance or when adding more nodes does not improve performance further.

Cascade-Correlation Learning Architecture

Fahlman and Lebiere (1990) developed a growing algorithm that searches in the topology space of *cascade architectures*. A cascade architecture is a special type of feedforward neural network in which each hidden layer has only one unit and each layer gets input from all the nodes in all the input and hidden layers before it. Thus each output node is connected to all the input nodes and all the hidden nodes of the network (see the top left of Figure 1).

In this algorithm, the network is frozen before the addition of each node. A pool of candidate nodes is trained to predict the current residual error. The best of the candidates is then connected to the network’s output nodes, thereby reducing the residual error. The rest of the candidates are discarded. This is repeated until the network’s error rate is satisfactory. Conceptually, this is like a Tay-

lor approximation, in which early nodes express lower-order terms while later nodes are like higher-order terms that add to making a finer approximation.

Dynamic Node Creation

Ash (1989) introduced the dynamic node creation (DNC) method for adding hidden nodes to a three-layer feedforward neural network. The algorithm starts with a network with only one hidden node. The given network is then trained normally to reduce its average squared error E . A new hidden node is added whenever learning has slowed down. Specifically, the algorithm adds a node when the reduction in error in the previous T epochs is less than some threshold Δ set by the designer, given that no nodes were added during those T epochs. The algorithm stops when another user-defined criterion (e.g., average or maximum output error) is met.

A major difference between cascade-correlation learning and DNC is that cascade-correlation learning freezes the network before the addition of each node. Ash (1989) argued in favor of DNC, claiming that since all weights are continuously adjusted as each node is added, a more complete search is done in the lower-dimensional weight space. For this reason, DNC can often find smaller acceptable networks than can cascade correlation. On the other hand, Fahlman and Lebiere (1990) believe that a significant amount of time in weight learning is due to the weights trying to adjust to each other. Freezing the network before each node addition enables the weights to learn in a stable context and focuses the learning effort on reducing the output error. Therefore, overall training should be faster.

Pruning

Pruning entails removing weights or nodes from a trained network. When growing a network, one knows that adding nodes will always decrease a network’s training error. Conversely, pruning nearly always increases a network’s training error. A good pruning algorithm should thus start with a fairly large network with low training error, and should prune the weights or nodes whose removal will

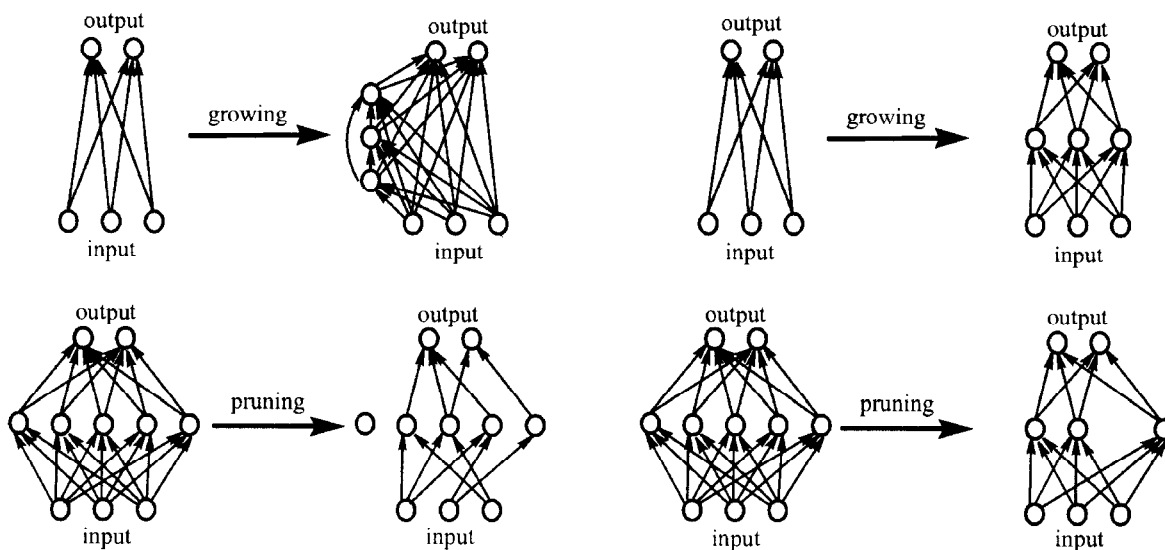


Figure 1. The figure shows network growing and pruning. At the top left, hidden nodes are added in a cascade topology. At the top right, hidden nodes are added in a three-layer topology. At the bottom right, nodes are

pruned from a network. At the bottom left, individual weights are pruned. Note that if all weights are eliminated from a node, then that node has been effectively pruned from the network.

increase a network's training error the least (Reed, 1993). After such removal one may elect to retrain the network for better performance. Some pruning methods were designed to remove individual weights, while others remove entire nodes. Pruning all of a node's input weights or all of its output weights is equivalent to pruning the node itself.

Various researchers have developed approximations to a weight or a node's *saliency*, which is defined as the increase in training error from that weight/node's removal. Those algorithms then proceed to prune weights/nodes with low saliency. Other researchers have worked under the magnitude-based pruning paradigm. In this approach, weight-learning algorithms (generally backpropagation) are modified such that the magnitude of weights (or some other parameters) are trained to become good indicators of saliency. After such training, low-magnitude weights are then pruned.

Magnitude-based Pruning

A weight with zero magnitude can be pruned without any effect on the network. This leads to the intuitive notion that weights with small magnitude can be pruned away without affecting training error too much. This simple version of *magnitude-based pruning* often does not work, however. Consider a neuron working in the linear region of its input-output function. The same function can be expressed either as a neuron with small input weights and large output weights, or as a neuron with medium input weights and medium output weights. The magnitude of each individual weight alone does not necessarily tell one about its saliency.

However, many researchers have added bias terms to the network error in weight-learning algorithms to penalize "complex" networks (Chauvin, 1989; Hanson and Pratt, 1989; Ji, Snapp, and Saltis, 1990). These biases are designed to favor networks in which magnitude-based pruning makes sense. The topological bias is thus embedded in the weight-learning algorithm.

Ji et al. (1990) proposed a penalty term to reduce the number of hidden units and the magnitude of weights. Their effect was to impose a smoothness constraint on the network's input-output function. Chauvin (1989) proposed a penalty term to reduce the sum of "energy" from all the hidden nodes, in which "energy" is some positive monotonic function of the square of a node's output. Examples of energy function include o^2 , $\log(1 + o^2)$, and $o^2/(1 + o^2)$, where o is the output of a node. Hanson and Pratt (1989) proposed a penalty term that is the sum of biases from all units. They specifically examined the hyperbolic bias, $w_i/(1 + \lambda w_i)$, and the exponential bias, $1 - e^{-\lambda w_i}$ (where λ is an adjustable parameter and $w_i = \sum_j \|w_{ij}\|$). Kruschke and Movellan (1991) proposed a penalty term that eliminates hidden nodes that are functionally redundant, as when the correlation between their input weights is high. Some authors (Hanson and Pratt, 1989; Ji et al., 1990) have noted that the addition of a penalty term significantly slows learning.

Skeletonization

Mozer and Smolensky (1989) developed skeletonization, a technique that prunes one node at a time. They introduce an *attentional strength* α_j for each node j . The output of node i is then

$$o_i = f\left(\sum_j \alpha_j w_{ij} o_j\right)$$

If $\alpha_j = 1$, then node j is just a conventional node. If $\alpha_j = 0$, then node j is considered pruned. The saliency of a node is then just the difference between the network's output error when α_j is 1 and when α_j is 0. They note that this saliency is an approximation to the derivative of the output error function with respect to α_j ,

$$E_{\alpha_j=0} - E_{\alpha_j=1} \approx -\left.\frac{\partial E}{\partial \alpha_j}\right|_{\alpha_j=1}$$

Conversely, they use the derivative to approximate saliency for pruning. The approximation can be computed with a backpropagation-like algorithm. In practice, $\partial E/\partial \alpha$ fluctuates strongly in time, and the saliency estimate is smoothed by exponentially decay time averaging. Mozer and Smolensky also use a linear error function in assessing saliency while using the traditional squared error for training.

When tested over a set of logic problems, skeletonization showed that it could correctly identify the salient input and hidden nodes. The skeleton network can also learn with comparable or fewer number of training epochs than a standard network (Mozer and Smolensky, 1989).

Optimal Brain Damage and Optimal Brain Surgeon

The effect of a weight change on output error, which we have defined before as saliency, can be approximated by a Taylor-series expansion,

$$\delta E = \left(\frac{\partial E}{\partial \mathbf{w}}\right)^T \cdot \delta \mathbf{w} + \frac{1}{2} \delta \mathbf{w}^T \cdot \mathbf{H} \cdot \delta \mathbf{w} + O(\|\delta \mathbf{w}\|^3)$$

where $\mathbf{H} \equiv \partial^2 E/\partial \mathbf{w}^2$ is the Hessian matrix, $\delta \mathbf{w}$ is a small candidate weight change, and the superscript T denotes vector transpose. When the network is trained to a local optimum on the error surface, the first term on the right-hand side becomes zero. In the quadratic approximation, cubic and higher-order terms are ignored. One is left with just the second term and can then solve for $\delta \mathbf{w}$ such that (at least) one of the weights become zero in $\mathbf{w} + \delta \mathbf{w}$ yet the predicted increase in error, $\frac{1}{2} \delta \mathbf{w}^T \cdot \mathbf{H} \cdot \delta \mathbf{w}$, is minimum. In addition to finding the weight whose removal increases error the least, this approach also finds an adjustment for the remaining weights to compensate for the weight removal.

The full algorithm follows:

1. Train a "reasonably large" network to minimum error.
2. Compute \mathbf{H}^{-1} . Optimal Brain Damage (OBD) (Le Cun, Denker, and Solla, 1990) assumes \mathbf{H} to be diagonal, which has a simple inverse. Optimal Brain Surgeon (OBS) (Hassibi and Stork, 1993) gives an efficient recursive algorithm to calculate the inverse of the full Hessian.
3. Find the candidate weight (indexed by j) that gives the smallest saliency

$$L_j = \frac{w_j^2}{2[\mathbf{H}^{-1}]_{jj}}$$

If this candidate error increase is greater than a user-specified threshold, go to step 5.

4. Delete the candidate weight. Update the remaining weights by

$$\delta \mathbf{w} = -\frac{w_j}{[\mathbf{H}^{-1}]_{jj}} \mathbf{H}^{-1} \cdot \mathbf{u}_j$$

where \mathbf{u}_j is the unit vector corresponding to j , the index of the weight that was deleted. Go back to step 2.

5. No more weights can be deleted without a large increase in network error. At this point one may want to retrain the network.

OBS is more effective than OBD at finding the correct weights to prune and is suitable for small and medium-size networks. For large networks, however, the computational and storage requirement for deriving the full \mathbf{H}^{-1} may be infeasible, and OBD should be used instead.

Topology Learning in Bayes Nets

Topology learning in Bayes nets has become a significant research area (Buntine, 1996). Since the usage of Bayes nets can be quite different from that of feedforward neural networks, topology learning for Bayes nets operates under a different set of assumptions.

Typically, each node in a Bayes net stands for a random variable that has some specific meaning in the real world. Adding or deleting a node in a Bayes net changes the semantics of the underlying domain, and neither growing nor pruning nodes is therefore done in Bayes nets. Furthermore, the direction of the links in a feedforward neural network denotes information flow and is predetermined (i.e., going from input layer to the hidden layers to the output layer), whereas the direction of links in a Bayes net establishes parent-child relationships to imply certain conditional independencies.

As with neural nets, topology learning in Bayes nets relies on a trade-off between network complexity and accuracy in modeling training data. In Bayes net topology learning, this trade-off is often expressed explicitly in a single metric. One of the more popular metrics is minimum description length (MDL) (Lam and Bacchus, 1994). Heckerman, Geiger, and Chickering (1995) derived another metric, BDe, from methods of Bayesian statistics. The MDL metric is simply the sum of a network's complexity and its accuracy in modeling data, while the BDe is a measure of prior belief plus data. These two metrics converge asymptotically as more data are given (under certain technical assumptions).

As expected, searching over the topology space to optimize a metric is computationally expensive. One usually has to resort to greedy searches. Fortunately, both metrics mentioned above are decomposable into a sum of scores for each node. That is, for nodes labeled X_1, \dots, X_m , the metric for the network can be written as $\sum_i \text{Score}(X_i, \text{Parents}(X_i))$. When an arc is added or deleted, i.e., when a parent is added or deleted, the action affects only one term in the summation. The overall metric can thus be quickly recalculated for each step of the search.

Discussion

This article has highlighted two main issues in topology learning. One is the trade-off between network complexity and fit to data, the other is the selection of a search strategy over the topology space. Growing is a search strategy that adds nodes, while pruning deletes parameters or nodes. In learning Bayesian networks, any efficient greedy search strategy can be used. In practice, when a "good" topology is highly problem dependent and the practitioner has little guideline on how to select a "good" topology, it makes sense to learn it from available data.

Road Map: Learning in Artificial Networks

Related Reading: Bayesian Networks; Graphical Models: Structure Learning

References

- Ash, T., 1989, Dynamic node creation in backpropagation networks, *Connect. Sci.*, 1:365–375.
- Buntine, W., 1996, A guide to the literature on learning probabilistic networks from data, *IEEE Trans. Knowledge Data Eng.*, 8:195–210. ♦
- Chauvin, Y., 1989, A back-propagation algorithm with optimal use of hidden units, in *Advances in Neural Information Processing Systems*, vol. 1 (D. S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 519–526.
- Cheeseman, P., 1990, On finding the most probable model, in *Computational Models of Scientific Discovery and Theory Formation* (J. Shragar and P. Langley, Eds.), San Mateo, CA: Morgan Kaufmann, pp. 73–95. ♦
- Fahlman, S. E., and Lebiere, C., 1990, The Cascade-Correlation learning architecture, in *Advances in Neural Information Processing Systems*, vol. 2 (D. S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 524–532.
- Hanson, S. J., and Pratt, L. Y., 1989, Comparing biases for minimal network construction with backpropagation, in *Advances in Neural Information Processing Systems*, vol. 1 (D. S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 177–185.
- Hassibi, B., and Stork, D. G., 1993, Second order derivatives for network pruning: Optimal Brain Surgeon, in *Advances in Neural Information Processing Systems*, vol. 5 (S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds.), San Mateo, CA: Morgan Kaufmann, pp. 164–171.
- Heckerman, D., Geiger, D., and Chickering, D., 1995, Learning Bayesian networks: The combination of knowledge and statistical data, *Machine Learn.*, 20:197–243. ♦
- Ji, C., Snapp, R. R., and Psaltis, D., 1990, Generalizing smoothness constraints from discrete samples, *Neural Computat.*, 2:188–197.
- Karnin, E. D., 1990, A simple procedure for pruning back-propagation trained neural networks, *IEEE Trans. Neural Netw.*, 1:239–242.
- Kruschke, J. K., and Movellan, J. R., 1991, Benefits of gain: Speeded learning and minimal hidden layers in back-propagation networks, *IEEE Trans. Syst. Man Cybern.*, 21:273–280.
- Lam, W., and Bacchus, F., 1994, Learning Bayesian belief networks: An approach based on the MDL principle, *Computat. Intell.*, 10:269–293.
- Le Cun, Y., Denker, J. S., and Solla, S. A., 1990, Optimal brain damage, in *Advances in Neural Information Processing Systems*, vol. 2 (D. S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 598–605.
- Mozier, M. C., and Smolensky, P., 1989, Skeletonization: A technique for trimming the fat from a network via relevance assessment, in *Advances in Neural Information Processing Systems*, vol. 1 (D. S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 107–115.
- Reed, R., 1993, Pruning algorithms—a survey, *IEEE Trans. Neural Netw.*, 4:740–747. ♦

Learning Vector Quantization

Teuvo K. Kohonen

Introduction

Neural network models are often applied to statistical pattern recognition problems, in which the class distributions of pattern vectors usually overlap and one must pay attention to the optimal location of the decision borders. The *learning vector quantization* (LVQ) algorithms discussed in this article define very good approximations for the optimal decision borders. These algorithms are computationally very light.

In the basic competitive learning neural networks to which the LVQ belongs, all cells may be thought to form an input layer, while there also exist mutual feedbacks or other types of lateral interaction between the cells. All cells receive the same external input, and by means of comparisons made in the lateral direction of the layer, an active response is switched on at the cell with the highest activation (the "winner"), while the responses of all the other cells are suppressed; this is called the *winner-take-all* (WTA) function (cf. Didday, 1970, 1976; Grossberg, 1976; Amari and Arbib, 1977).

Although many competitive learning neural networks have been based on nonlinear dynamic neural models, the decision or classification functions that ensue from their collective behavior are simply described by a formalism that was originally developed for signal analysis, namely, *vector quantization* (VQ). (For a general review of VQ, see Makhoul, Roucos, and Gish, 1985.) Thus, VQ and the neural network models of competitive learning are not alternative methods: the former is an idealized description of the latter on the signal-space level.

Vector Quantization

As in simple neural network models, we assume a signal vector $x = (\xi_1, \xi_2, \dots, \xi_n) \in \mathbb{R}^n$ and a set of units or cells, each provided with a parametric vector (called *codebook vector*) $m_i = (\mu_{i1}, \mu_{i2}, \dots, \mu_{in})^T \in \mathbb{R}^n$. The winner in the category of VQ problems is usually defined as the unit c whose codebook vector has the smallest Euclidean distance from x :

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \quad (1)$$

If x is a natural, stochastic, continuous-valued vectorial variable, we need not consider multiple minima: the probability for $\|x - m_i\| = \|x - m_j\|$ for $i \neq j$ is then zero.

The VQ methods were originally developed to compress information. The m_i had to be placed into the input signal space such that the average expected quantization error E was minimized:

$$E = \int \|x - m_{c(x, m_1, \dots, m_k)}\|^2 p(x) dx = \min! \quad (2)$$

where $p(x)$ is the probability density function of x , and dx is a hypervolume differential in the signal space. Notice that c , the index of the winner, depends on x and all the m_i . It has been shown by Zador (1982), for example, that the point density of the m_i values that minimize E is proportional to $[p(x)]^{n/(n+2)}$. Since in practical problems usually $n \gg 2$, it can then be said that the distribution of the m_i approximates $p(x)$.

Optimal Decision

Assume that all samples of x are derived from a finite set of classes $\{S_k\}$, the distributions of which are allowed to overlap. The problem of optimal decision or statistical pattern recognition is usually discussed within the framework of the Bayes theory of probability (for a textbook account, see, e.g., Kohonen, 1989, chap. 7.2). Let

$P(S_k)$ be the a priori probability of class S_k , and $p(x|x \in S_k)$ be the conditional probability density function of x on S_k , respectively. In this method the so-called *discriminant functions* are defined as

$$\delta_k(x) = p(x|x \in S_k)P(S_k) \quad (3)$$

It can be shown that the average rate of misclassifications is minimized if the sample x is determined to belong to class S_c according to

$$\delta_c(x) = \max_k \{\delta_k(x)\} \quad (4)$$

Learning Vector Quantization Algorithms

The LVQ1

Consider now Figure 1. In the LVQ approaches we assign a *subset of codebook vectors to each class S_k* and then search for the codebook vector m_i that has the smallest Euclidean distance from x . This assignment can be made in such a way that codebook vectors belonging to different classes are not intermingled, although the class distributions overlap. The sample x is then thought to belong to the same class as the closest m_i . As only codebook vectors closest to the class borders define the decision borders, a good approximation of $p(x|x \in S_k)$ is not necessary everywhere. We must place the m_i into the signal space in such a way that the nearest-neighbor rule (Equation 1) minimizes the average expected misclassification probability. Notice that in considering the average expected classification accuracy, the quantization errors can be large in regions where $p(x)$ has small values.

Let

$$c = \arg \min_i \{\|x - m_i\|\} \quad (5)$$

define the index of the nearest m_i to x , denoted by m_c ; x is then determined to belong to the same class to which the nearest m_i belongs.

Let $x = x(t)$ now be a time-series sample of input, and let the $m_i(t)$ represent sequential values of the m_i in the discrete-time domain, $t = 0, 1, 2, \dots$, obtained in the following process. Starting with properly defined initial values $m_i(0)$, Equation 6 shall define the basic learning vector quantization process (Kohonen, 1988; Kohonen, Barna, and Chrisley, 1988); this particular algorithm is called LVQ1.

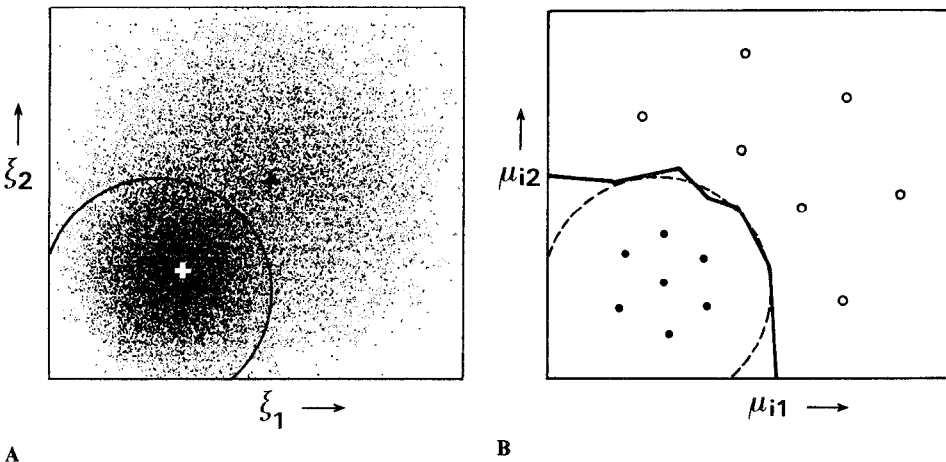


Figure 1. A, The probability density function of $x = [\xi_1, \xi_2]^T$ is represented here by its samples, the small dots. The superposition of two symmetric Gaussian density functions corresponding to two different classes S_1 and S_2 , with their centroids shown by the white and dark cross, respectively, is shown. Solid curve denotes the theoretical optimal Bayes decision surface. B, Large black dots denote codebook vectors of class S_1 ; open circles denote codebook vectors of class S_2 ; the solid curve indicates a decision surface obtained by LVQ1; the broken curve indicates a Bayes decision surface.

$$\begin{aligned}
m_c(t+1) &= m_c(t) + \alpha(t)[x(t) - m_c(t)] \\
&\quad \text{if } x \text{ and } m_c \text{ belong to the same class} \\
m_c(t+1) &= m_c(t) - \alpha(t)[x(t) - m_c(t)] \\
&\quad \text{if } x \text{ and } m_c \text{ belong to different classes} \\
m_i(t+1) &= m_i(t) \text{ for } i \neq c
\end{aligned} \tag{6}$$

It will be shown that the asymptotic values of m_i obtained in this process define a vector quantization for which the rate of misclassifications is approximately minimized. Here $0 < \alpha(t) < 1$, and $\alpha(t)$ (learning-rate factor) is usually made to decrease monotonically with time. It is recommended that α should initially be rather small, say, smaller than 0.1. If only a restricted time is available for learning, the exact law $\alpha = \alpha(t)$ is not crucial, especially also if only a restricted set of training samples is available; they may be applied cyclically, and $\alpha(t)$ may even be made to decrease linearly to zero.

It is in general difficult to show what the exact convergence limits of Equation 6 are. The following discussion is based on the idea that VQ tends to approximate density functions such as $p(x)$. Instead of $p(x)$, we may also consider any non-negative function $f(x)$ in Equation 2.

The Bayes decision borders defined by Equations 3 and 4 divide the signal space into class regions B_k such that the rate of misclassifications is minimized. All such borders together are defined by the condition $f(x) = 0$, where, for $x \in B_k$ and $h \neq k$,

$$f(x) = p(x|x \in S_k)P(S_k) - \max_k \{p(x|x \in S_h)P(S_h)\} \tag{7}$$

Notice that $f(x)$ is piecewise continuous and non-negative. For each $x \in B_k$, $f(x)$ has a positive hump, and these humps are separated by the Bayes borders at which $f(x) = 0$.

If we approximate $f(x)$ by the point density of codebook vectors defined by classical VQ, this point density must then also tend to zero at all borders.

In order to find the minimum of E in Equation 2 and the optimal values for the m_i in VQ by gradient descent, we need an expression for the gradient of E . From Kohonen (1991, Equations A1 through A14) we obtain the result

$$\nabla_{m_i} E = -2 \int \delta_{ci} \cdot (x - m_i) p(x) dx \tag{8}$$

where δ_{ic} is the Kronecker delta, and c is the index of the m_i that is closest to x (i.e., the winner). The gradient step of vector m_i is

$$m_i(t+1) = m_i(t) - \lambda \cdot \nabla_{m_i(t)} E \tag{9}$$

where λ defines the step size, and

$$\nabla_{m_i(t)} E = -2\delta_{ci}[x(t) - m_i(t)] \tag{10}$$

If $p(x)$ in E is now replaced by $f(x)$, we get by substitution:

$$\begin{aligned}
\nabla_{m_i} E &= -2 \int \delta_{ci}(x - m_i) f(x) dx \\
&= -2 \int \delta_{ci}(x - m_i) [p(x|x \in S_k)P(S_k) \\
&\quad - \max_k \{p(x|x \in S_h)P(S_h)\}] dx
\end{aligned} \tag{11}$$

The gradient steps must be computed separately in the event that the sample $x(t)$ belongs to S_k , and in the event that $x(t) \in S_h$. In the event that $x(t) \in S_k$, we obtain

$$\nabla_{m_i(t)} E = -2\delta_{ci}[x(t) - m_i(t)] \tag{12}$$

with the a priori probability $P(S_k)$.

The class with $\max_h \{p(x|x \in S_h)P(S_h)\}$ is the runner-up class signified by index r . In the event that $x(t) \in S_r$, the following expression for $\nabla_{m_i} E$ is obtained with the a priori probability $P(S_r)$:

$$\nabla_{m_i(t)} E = +2\delta_{ci}[x(t) - m_i(t)] \tag{13}$$

The different cases are collected into the following set of equations, rewritten with $\alpha(t) = 2\lambda$:

$$\begin{aligned}
m_c(t+1) &= m_c(t) + \alpha(t)[x(t) - m_c(t)] \\
&\quad \text{for } x(t) \in B_k \text{ and } x(t) \in S_k \\
m_c(t+1) &= m_c(t) - \alpha(t)[x(t) - m_c(t)] \\
&\quad \text{for } x(t) \in B_k \text{ and } x(t) \in S_r \\
m_c(t+1) &= m_c(t) \text{ for } x(t) \in B_k \text{ and } x(t) \in S_h, h \neq r \\
m_i(t+1) &= m_i(t) \text{ for } i \neq c
\end{aligned} \tag{14}$$

If the m_i of class S_k are already within B_k , the VQ will further attract them to the hump corresponding to B_k , at least if the learning steps are small. With a sufficiently large number of codebook vectors in each class region B_k , the closest codebook vectors in adjacent regions B_k will be arbitrarily close to the Bayes border. Thus, VQ and Equation 7 have been shown to define the Bayes borders with arbitrarily good accuracy.

Near equilibrium, close to the borders at least, Equations 6 and 14 can be seen to define almost similar corrections; notice that in Equation 6, *the classification of x was approximated by the nearest-neighbor rule*, and this approximation will be improved during learning. However, notice too that in Equation 6 the minus sign corrections were made every time when x was classified incorrectly, whereas Equation 14 only makes the corresponding correction if x is exactly in the runner-up class. The error thereby made is often insignificant. As a matter of fact, the algorithms called LVQ2 and LVQ3 (Kohonen, 1990) are even closer to Equation 14 in this respect.

The Optimized-Learning-Rate LVQ1 (OLVQ1)

If an individual learning rate $\alpha_i(t)$ is assigned to each m_i , we obtain the following modified learning process (Kohonen, 1992). Let c be defined by Equation 5. Then

$$\begin{aligned}
m_c(t+1) &= m_c(t) + \alpha_c(t)[x(t) - m_c(t)] \\
&\quad \text{if } x \text{ is classified correctly} \\
m_c(t+1) &= m_c(t) - \alpha_c(t)[x(t) - m_c(t)] \\
&\quad \text{if } x \text{ is classified incorrectly} \\
m_i(t+1) &= m_i(t) \text{ for } i \neq c
\end{aligned} \tag{15}$$

We may try to determine the $\alpha_i(t)$ for fastest convergence of Equations 15. Let us express Equations 15 in the shorter form

$$m_c(t+1) = [1 - s(t)\alpha_c(t)]m_c(t) + s(t)\alpha_c(t)x(t) \tag{16}$$

where $s(t) = +1$ if the classification is correct, and $s(t) = -1$ if the classification is wrong. It may be obvious that the *statistical accuracy* of the learned codebook vector values is approximately optimal if all samples have been used with equal weight, i.e., if the effects of the corrections made at different times, when referring to the end of the learning period, are of approximately equal magnitude. Notice that $m_c(t+1)$ contains a trace of $x(t)$ through the last term in Equation 16, and traces of the earlier $x(t')$, $t' = 1, 2, \dots, t-1$, through $m_c(t)$. In a learning step, the magnitude of the last trace of $x(t)$ is scaled down by the factor $\alpha_c(t)$, and, for instance, during the same step the trace of $x(t-1)$ becomes scaled down by $[1 - s(t)\alpha_c(t)] \cdot \alpha_c(t-1)$. Now we first stipulate that these two scalings must be identical:

$$\alpha_c(t) = [1 - s(t)\alpha_c(t)]\alpha_c(t-1) \tag{17}$$

If this condition is made to hold for all t , by induction it can be shown that the traces collected up to time t of all the earlier $x(t')$ will be scaled down by an equal amount at the end, and thus the "optimal" values of $\alpha_i(t)$ are determined by the recursion

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \tag{18}$$

For fast learning, the OLVQ1 algorithm can be started with the $\alpha_i(0)$ in the range of 0.3 to 0.5.

General Considerations

Initialization of the Codebook Vectors

A rather good strategy is to start with the same number of codebook vectors in each class. An upper limit to the total number of codebook vectors is set by the restricted recognition time and computing power. An identical number of codebook vectors per class is justifiable, since for optimal approximation of the borders the average distances between the adjacent codebook vectors (which depend on their numbers per class) ought to be the same in each class. Because the final placement of the codebook vectors and thus their distances are not known until the end of the learning process, equalization of the distances in the various classes ought to be made iteratively.

Once the numbers of codebook vectors have been fixed, one may use first samples of the real training data for their initial values. Referring to the derivation of Equation 9, however, the codebook vectors should always remain inside the respective class regions. For the initial values one can then accept only samples that are not misclassified. In other words, a sample is first tentatively classified against all the other samples in the training set, for instance by the traditional k -nearest-neighbor (kNN) method, and accepted for a possible initial value only if this tentative classification is the same as the class identifier of the sample. (In the learning algorithm itself, however, no samples must be excluded; they are applied independently of whether they fall on the correct side of the class border or not.)

Overall Learning Strategy

One may start the learning with the OLVQ1 algorithm, which has fast convergence; its final recognition accuracy will approximately be achieved after a number of learning steps that is about 30 to 50 times the total number of codebook vectors. In an attempt to ultimately improve recognition accuracy, one may try to continue with the basic LVQ1, or with the other LVQ versions, using a low initial value of learning rate, which is then the same for all classes.

The neural network algorithms often “overlearn”; i.e., when the learning and test phases are alternated, the recognition accuracy is first improved until an optimum is reached. After that point, the accuracy often starts to decrease slowly. A possible explanation in the present case is that when the codebook vectors become very specifically tuned to the training data, the ability of the algorithm to generalize with respect to new data suffers. It is therefore necessary to stop the additional learning process after some optimal number of steps, say, 50 to 200 times the total number of the codebook vectors. Such a stopping rule can only be found by experience.

Comparison with Other Methods

LVQ and SOM

Another related algorithm, the *self-organizing map* (SOM) (Kohonen, 1989, 1990, 1993; see also SELF-ORGANIZING FEATURE MAPS) should be mentioned in this context. It is an unsupervised learning method, whereas supervised training is used in the LVQ. In LVQ, only one or two winner cells are updated during each adaptation step, whereas in the SOM, a block of neighboring cells around the winner *relating to the physical network* is updated simultaneously. The main application areas of these algorithms are also different: LVQ is used for the classification of stochastic data, whereas the SOM is more useful for the visualization of high-dimensional data on a two-dimensional display.

Table 1. Error Percentages in Phonemic Classification

Parametric Bayes	kNN	LVQ1
12.1	12.0	10.2

Relative Performance of LVQ as a Classifier

The number of different applications of LVQ (and SOM) is for the present at least on the order of many hundreds, and it is very difficult to make any comparative survey. Just to give a feeling of the relative performance, Table 1 (Kohonen, 1990) compares the classification accuracies achievable by a couple of classical methods as well as by LVQ1. The data, 15-channel acoustic spectra representing 19 different phonemic classes, were collected from Finnish speech. A total of 1,550 samples were used for training, and another set of 1,550 independent samples for testing, respectively. There were in total 117 codebook vectors.

The term *parametric Bayes* in Table 1 means a Bayesian classification method in which the discriminant functions are approximated by multivariate normal distributions, and kNN is the classical k -nearest-neighbor method, i.e., comparison of each test sample against all the training samples and voting over k closest ones (here $k = 5$). It is generally known that the kNN algorithm gives a very good approximation of the theoretical Bayes limit; but LVQ1 is still better, and here more than ten times faster computationally.

[Reprinted from the First Edition]

Road Map: Learning in Artificial Networks

Related Reading: Competitive Learning; Coulomb Potential Learning; Data Clustering and Learning; Self-Organizing Feature Maps

References

A list of over 5,000 literature references to analyses and applications of LVQ and SOM algorithms is available on the Internet, at the address <http://www.icsi.berkeley.edu/~jagota/NCS>. Extensive software packages of LVQ and SOM algorithms, diagnostic programs, and exemplary data can be found at <http://www.cis.hut.fi/> research.

The following works have been referred to in this article:

- Amari, S., and Arbib, M. A., 1977, Competition and cooperation in neural nets, in *Systems in Neuroscience* (J. Metzler, Ed.), New York: Academic Press, pp. 119–165. ♦
- Didday, R. L., 1970, The simulation and modelling of distributed information processing in the frog visual system, PhD diss., Stanford University.
- Didday, R. L., 1976, A model of visuomotor mechanisms in the frog optic tectum, *Math. Biosci.*, 30:169–180.
- Grossberg, S., 1976, Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors, *Biol. Cybern.*, 23:121–134; II. Feedback, expectation, olfaction, illusions, *Biol. Cybern.*, 23:187–202.
- Kohonen, T., 1988, An introduction to neural networks, *Neural Netw.*, 1:3–16. ♦
- Kohonen, T., 1989, *Self-Organization and Associative Memory*, 3rd ed., Berlin: Springer-Verlag.
- Kohonen, T., 1990, The self-organizing map, *Proc. IEEE*, 78:1464–1480. ♦
- Kohonen, T., 1991, Self-organizing maps: Optimization approaches, in *Artificial Neural Networks* (T. Kohonen, K. Makisara, O. Simula, and J. Kangas, Eds.), Amsterdam: Elsevier, vol. 2, pp. 1677–1680.

- Kohonen, T., 1992, New developments of learning vector quantization and the self-organizing map, in *Symposium on Neural Networks: Alliances and Perspectives in Senri 1992 (SYNAPSE'92)*, Osaka, Japan.
- Kohonen, T., 1993, Physiological interpretation of the self-organizing map algorithm, *Neural Netw.*, 6:895–905.
- Kohonen, T., Barna, G., and Chrisley, R., 1988, Statistical pattern recognition with neural networks: Benchmarking studies, in *Proceedings of*

- the IEEE International Conference on Neural Networks*, vol. 1, New York: IEEE, pp. 61–68.
- Makhoul, J., Roucos, S., and Gish, H., 1985, Vector quantization in speech coding, *Proc. IEEE*, 73:1551–1588. ♦
- Zador, P. L., 1982, Asymptotic quantization error of continuous signals and the quantization dimensions, *IEEE Trans. Inform. Theory*, IT-28:139–149.

Lesioned Networks as Models of Neuropsychological Deficits

John A. Bullinaria

Introduction

Cognitive neuropsychology uses the patterns of performance observed in brain-damaged patients to constrain our models of normal cognitive function. Historically, this methodology was rooted in simple “box-and-arrow” models, with particular cognitive deficits being taken to indicate selective breakdown of corresponding “boxes” or “arrows.” Studying patients with complementary patterns of deficit allows us, in principle, to piece together a complete model of mental structure (Shallice, 1988). Of particular importance in this process has been the concept of *double dissociation*, which has been taken to imply modularity within many systems. If one patient can perform task 1 better than task 2 and another can perform task 2 better than task 1, then a natural explanation is in terms of separate modules for the two tasks.

In recent years, connectionist techniques have been employed to model the operation and interaction of these “modules” in increasing detail (Farah, 1994). Networks of simplified processing units loosely based on real neurons are set up with general architectures based on known physiology, trained to perform appropriately simplified versions of the human tasks, and iteratively refined by checking their performance against humans’. Such network models can clearly be wired together in the manner of the old box-and-arrow models, with all the old explanations of patient data carrying through. The obvious advantage now is that we can look at the details of the degradation of the various components and, by removing neurons or connections in our models, construct natural analogues of real brain damage. Moreover, in addition to elaborating previous models, we can also question the validity of the old assumptions of neuropsychological inference and explore the possibility that processing is actually more distributed and interactive than the older models implied.

This article reviews the general issues involved in lesioning neural network models to simulate neuropsychological deficits. I shall point out potential sources of misleading results, clarify apparent contradictions in the literature, and discuss some representative models.

Lesioning Simple Feedforward Networks

Many neural network models of human performance are based on simple feedforward networks that map between conveniently simplified input and output representations via a single hidden layer, or that have such systems as identifiable subcomponents. An important feature of these models is that they *learn* to perform the relevant tasks by iteratively adjusting their connection weights (e.g., by some form of gradient descent algorithm) to minimize the output errors for an appropriate training set of input-output pairs. Generally, we simply assume that the quick and convenient learning algorithms we choose will generate results similar to those

produced by more biologically plausible procedures. Comparisons between backpropagation and contrastive Hebbian learning by Plaut and Shallice (1993) provide some justification for this assumption. We can then compare the development of the networks’ performance during training and their final performance (e.g., their output errors, generalization ability, reaction times, priming effects, speed-accuracy trade-offs, robustness to damage, etc.) with the performance of human subjects to narrow down the correct architecture, representations, and so on, to generate increasingly accurate models.

An obvious feature of network learning is that performance on one pattern will be affected by training on other patterns. It follows straightforwardly from adding up the network weight change contributions resulting from individual training patterns that:

1. Regular items will be learned more quickly than irregular items, because consistent weight changes combine and inconsistent weight changes cancel.
2. High-frequency items will be learned more quickly than low-frequency items, because the appropriate weight changes get applied more often.
3. Ceiling effects will arise as sigmoids saturate and weight changes tend to zero.

These fundamental properties of neural network learning not only result in human-like *age of acquisition* effects but indirectly account for realistic patterns of reaction times, speed-accuracy trade-off effects, and so on (Bullinaria, 1997). After training our networks and confirming that they are performing in a sufficiently human-like manner, we can then set about inflicting simulated brain damage on them. Small (1991) considered the various ways in which connectionist networks might be lesioned, and discussed their neurobiological and clinical neurological relevance. He identified two broad classes of lesion: *diffuse*, such as those created by globally scaling or adding noise to all the weights, and *focal*, such as those created by removing adjacent subsets of connections and/or hidden units. Which class we choose will depend on the type of patient we are modeling. Focal lesions would be appropriate for stroke patients, whereas diffuse lesions would be required for diseases such as Alzheimer’s. Generally, for our simplified models, it is appropriate to examine all these possibilities. Finally, we should be aware that relearning after damage may affect the observed pattern of deficits, and so we must check this also (Plaut, 1996).

The relevant issues have been explored in an abstract setting by Bullinaria (1999), who trained a simple feedforward network (with 10 inputs, 100 hidden units, and 10 outputs, with binary inputs and output targets) on two sets of 100 regular items (different permuted identity mappings) and two sets of 10 irregular items (random mappings). One regular set and one irregular set appeared during train-

ing 20 times more frequently than the others. Figure 1 shows that both regularity and frequency do indeed affect the speed of learning in the expected manner.

Bullinaria and Chater (1995) explored the effects of damage on fully distributed, homogeneous connectionist systems and investigated the possibility that double dissociation between regular and irregular items could arise without modularity. They found that lesioning trained networks by removing random hidden units, removing random connections, globally scaling the weights, or adding random noise to the weights all led to very similar patterns of deficits. They concluded that, assuming one successfully avoids small-scale artifacts and controls for all other factors, only single dissociations were possible. Moreover, these single dissociations were seen to be a natural consequence of the ease with which the mappings were originally learned. Plotting the patterns of activation feeding into the output units revealed why this should be the case. Each form of damage results in these activations either drifting in a random direction or falling to zero. For every output unit there will be some correct response threshold, and the items that are learned first during training will end up furthest past the thresholds when the training is stopped. They will consequently tend to be the last to cross over again and hence be the last to result in output errors as more damage is incurred. Thus we get clear dissociations, with the regulars more robust than frequency-matched irregulars and high-frequency items more robust than regularity-matched low-frequency items. Figure 2 shows this pattern explicitly for the network of Figure 1.

These basic effects extend easily to more realistic models, for example, surface dyslexia in the reading model of Bullinaria (1997). Here we successfully simulate not only the relative error proportions for the various word categories (i.e., regular/irregular, high/low frequency) but also the types of errors that are produced. The closest threshold to an irregularly pronounced letter will be that of regular pronunciation, and hence the errors will be predominantly regularizations of the lowest-frequency irregular items, exactly as is observed in acquired human surface dyslexia.

Figures 1 and 2 also reveal what is behind a potential source of confusion. Bullinaria and Chater (1995) argued that network lesions would always result in single dissociations with the regular items more robust. Marchman (1993), however, studied models of past tense production and seemingly found dissociations, with the irregular items more robust than the regulars. It is easy to see from the figures that sufficiently high-frequency irregulars can be more robust than regulars. The English language has evolved to leave

the irregulars with much higher frequencies than the regulars; otherwise, they would have been lost from the language. Marchman built this into her models, with the expected consequences. This illustrates how important it is to control for all confounding factors when describing dissociations and drawing conclusions from them. Lavric et al. (2001) provide a review of the issues involved in understanding the dissociations of verb morphology.

It is also evident from Figure 2 that, if the frequencies and regularities are carefully matched, performance on the high-frequency irregulars can cross performance on the lower-frequency regulars. Initially there is a dissociation with better performance on the irregulars, and later the opposite dissociation. Such a “double dissociation” is a form of *resource artifact* that is well known not to imply underlying modularity (Shallice, 1988, p. 234). Patterns of deficits of this type are actually rather easily obtainable in neural network models. Devlin et al. (1998) present an interesting example involving a connectionist account of category-specific semantic deficits. The finer grain of detail that connectionist modeling affords here allows explicit accounts of human deficits that older box-and-arrow models could accommodate only with difficulty.

The general point one can make about single, fully distributed subsystems is that some items are naturally learned more quickly and more accurately than others, and the effects of subsequent network damage follow automatically from these patterns of learning. There are actually many factors, in addition to regularity and frequency, that can cause differing learning and damage rates. We can explore them all in a similar manner and use them in models of neuropsychological data in the same way. Consistency and neighborhood density are the factors most closely related to regularity and are commonly found in models of language tasks such as reading and spelling (e.g., Plaut et al., 1996; Bullinaria, 1997). Representation sparseness or pattern strength are often used to distinguish between concrete and abstract semantics, as in models of deep dyslexia (e.g., Plaut and Shallice, 1993). Correlation, redundancy, and dimensionality are commonly used in models to distinguish the semantics of natural things versus artifacts, as in models of category-specific semantic deficits (e.g., Devlin et al., 1998). At some level of description, all of these factors act in a similar manner as frequency and regularity, and their effects can easily be confounded. Which we use will depend on exactly what we are attempting to model. If we want to make claims about neuropsychological deficits involving one of them, however, we need to be careful to control for all the others.

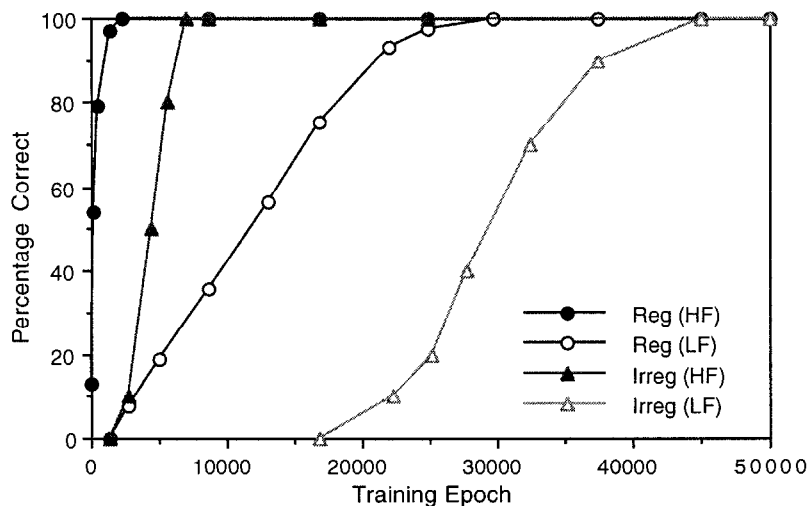
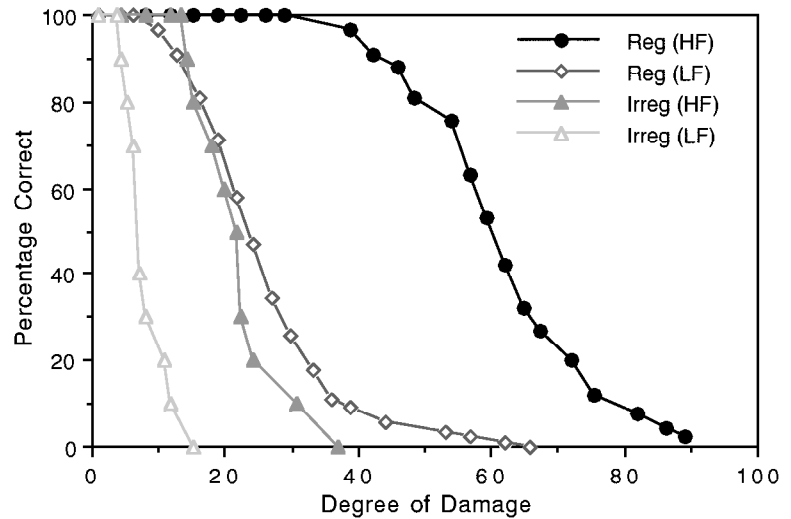


Figure 1. Regularity and frequency effects during the course of learning. HF, high frequency; LF, low frequency.

Figure 2. Regularity and frequency effects with increasing degrees of network damage. HF, high frequency, LF low frequency.



Following brain damage, patients often show a rapid improvement in performance. This is important to connectionist modelers for two reasons. First, if relearning occurs automatically and quickly in patients, then we need to be sure that the same effects are observed in our models, and that we are comparing patient and model data at equivalent stages of the relearning process. Second, our models may be of assistance in formulating appropriate remedial strategies for brain-damaged patients (Plaut, 1996). Since learning and damage have the same underlying regularity and frequency effects, relearning from the original training data is unlikely to reverse this pattern; indeed, it is likely to enhance it (Bullinaria and Chater, 1995). However, if some rehabilitation regime is employed that involves a very different set of training examples from that of the original learning process, it is possible for different results to arise (Plaut, 1996). Here the models can be used to predict or refine appropriate relearning strategies, and the patients' responses can be used to validate our models.

Small-Scale Artifacts

One should never forget that modeling massively parallel brain processes by simulating neural networks on serial computers is rendered feasible only by abstracting the essential details and scaling down the size of the networks. It is clearly important not to take the abstraction and scaling process so far that we miss important fundamental properties of the systems we are modeling, or introduce features that are nothing but small-scale artifacts. The damage curves of Figure 2 are relatively smooth because our network has many more hidden units and connections than are actually required to perform the given mappings, and individual connections or hidden units make only small contributions to the network's outputs. For smaller networks, however, the effect of individual damage contributions can be large enough to produce wildly fluctuating performance on individual items, and this can result in dissociations in arbitrary directions. Often these small-scale artifacts are sufficient to produce convincing-looking double dissociations (Shallice, 1988, p. 254). Bullinaria and Chater (1995) showed that as we scale up to larger networks, the processing becomes more distributed, and apparent double dissociations dissolve into single dissociations.

Our modeling endeavors would be much easier if some independent procedure could determine when networks were sufficiently distributed to obtain reliable results. In effect, we need to make sure that individual processing units are not acting as "mod-

ules" in their own right, and the obvious way to do this is by checking that all the individual contributions feeding into each output unit are small compared to the total. In this case, many such lost contributions must conspire to result in an output change large enough to be deemed an error. This is the brain-like resilience to damage often known as *graceful degradation*. Fortunately, this distribution of information processing tends to occur automatically if the network is supplied with a sufficiently large number of hidden units. However, in general, it seems that we do need many hidden units to avoid small-scale artifacts—many times the minimal number required to learn the given task (Bullinaria, 1999). So, what can be done if limited computational resources make this impossible? Obviously, after removing a random subset of the hidden units or connections, the number of contributions will be reduced by some factor, but in large, fully distributed networks, the mean contribution will not change much, and so the total contribution after damage is simply reduced by the same factor. We can achieve the same result simply by globally scaling all the weights by the same factor. In smaller networks, this equivalence breaks down because the means tend to suffer relatively large random fluctuations during damage. However, since global weight scaling does not suffer from such random fluctuations, it can be used to simulate a smoothed form of lesioning and give a reasonable approximation in small networks to what will happen in more realistic networks. Alternatively, if one wants to claim that each hidden unit corresponds to a number of real neurons, then the weight scaling can be regarded as removing a fraction of those neurons.

Lesioning Attractor Networks

Many successful models of human performance and their associated neuropsychological deficits have been based on attractor networks (see COMPUTING WITH ATTRACTORS) rather than simple feedforward networks. These are recurrent networks that develop *attractors* to appropriate patterns of activity; i.e., they have points in the *state space* of output activations to which the network settles. Lesions of this type of network can alter the settling behavior by distorting or shifting the *basins of attraction*. Here the errors correspond to the network settling into the wrong attractor, rather than an output unit activation failing to reach a particular threshold. Nevertheless, the resilience to damage still follows directly from how the particular items were originally learned.

One of the earliest applications of attractor networks to neuropsychology was the Mozer and Behrmann (1990) model of *neglect*

dyslexia. But perhaps the most successful models of this type are the Plaut and Shallice (1993) models of *deep dyslexia*, which were extensions of earlier work by Hinton and Shallice (1991) showing how both visual and semantic errors could arise from a single lesion. These attractor networks mapped from orthography to semantics via a layer of hidden units, and then from semantics to phonology via another set of hidden units, with layers of *clean-up units* at the semantics and phonology levels. One particular model was trained on 40 words, using backpropagation through time, until it settled into the correct semantics and phonology when presented with each orthography. Lesions at two different locations in the trained network were then found to produce a double dissociation between concrete and abstract word reading, where concreteness was coded as the proportion of activated semantic microfeatures. Specifically, removal of orthographic to hidden layer connections resulted in preferential loss of abstract word reading, whereas removal of connections to the semantic clean-up units primarily impaired performance on the concrete words. Although the two damage locations do not constitute modules in the conventional sense, it is not difficult to understand how they contribute to the processing of the two word types to different degrees, and give opposite dissociations when damaged. It is simply a consequence of the sparser representations of the abstract words making less use of the semantic clean-up mechanism, and depending more on the direct connections, than the richer representations of the concrete words (Plaut and Shallice, 1993). This does not conflict with the claim of Bullinaria and Chater (1995) that only single dissociations are possible. The robustness of each location in the attractor network is fully consistent with the general discussion above, and the only disagreement concerns the appropriateness of using the word “module” to describe the two damage locations. As Plaut himself points out (Plaut, 1995), one of the problems when discussing modularity is that different authors define the term differently. This is fine, but to avoid confusion one should be careful to quote the definitions along with the conclusions.

Discussion

This article has covered the basic issues and complications involved in lesioning neural network models to provide accounts of neuropsychological deficits, and has provided pointers to a range of representative case studies. It seems clear that, despite all the abstractions and simplifications involved, connectionist modeling has a lot to offer in fleshing out the details of, or even replacing, earlier box-and-arrow models to provide a more complete picture of cognitive processing. The resulting enhanced models and the

new field of connectionist neuropsychology not only are producing good accounts of existing empirical data, but are also beginning to suggest more appropriate experimental investigations for further fine-tuning of these models, and an ethical approach for exploring potential remedial actions for neuropsychological patients.

Road Map: Cognitive Neuroscience

Related Reading: Neurological and Psychiatric Disorders; Neuropsychological Impairments

References

- Bullinaria, J. A., 1997, Modelling reading, spelling and past tense learning with artificial neural networks, *Brain Lang.*, 59:236–266.
- Bullinaria, J. A., 1999, Connectionist dissociations, confounding factors and modularity, in *Connectionist Models in Cognitive Neuroscience* (D. Heinke, G. W. Humphreys, and A. Olsen, Eds.), London: Springer-Verlag, pp. 52–63.
- Bullinaria, J. A., and Chater, N., 1995, Connectionist modelling: Implications for cognitive neuropsychology, *Lang. Cognit. Proc.*, 10:227–264. ♦
- Devlin, J. T., Gonnerman, L. M., Andersen, E. S., and Seidenberg, M. S., 1998, Category-specific semantic deficits in focal and widespread brain damage: A computational account, *J. Cognit. Neurosci.*, 10:77–94.
- Farah, M. J., 1994, Neuropsychological inference with an interactive brain: A critique of the locality assumption, *Behav. Brain Sci.*, 17:43–104.
- Hinton, G. E., and Shallice, T., 1991, Lesioning an attractor network: Investigations of acquired dyslexia, *Psychol. Rev.*, 98:74–95.
- Lavric, A., Pizzagalli, D., Forstmeier, S., and Rippon, G., 2001, Mapping dissociations in verb morphology, *Trends Cognit. Sci.*, 5:301–308.
- Marchman, V. A., 1993, Constraints on plasticity in a connectionist model of the English past tense, *J. Cognit. Neurosci.*, 5:215–234.
- Mozer, M. C., and Behrmann, M., 1990, On the interaction of selective attention and lexical knowledge: A connectionist account of neglect dyslexia, *J. Cognit. Neurosci.*, 2:96–123.
- Plaut, D. C., 1995, Double dissociation without modularity: Evidence from connectionist neuropsychology, *J. Clin. Exp. Neuropsychol.*, 17:291–321. ♦
- Plaut, D. C., 1996, Relearning after damage in connectionist networks: Towards a theory of rehabilitation, *Brain Lang.*, 52:25–82.
- Plaut, D. C., McClelland, J. L., Seidenberg, M. S., and Patterson, K. E., 1996, Understanding normal and impaired word reading: Computational principles in quasi-regular domains, *Psychol. Rev.*, 103:56–115.
- Plaut, D. C., and Shallice, T., 1993, Deep dyslexia: A case study of connectionist neuropsychology, *Cognit. Neuropsychol.*, 10:377–500. ♦
- Shallice, T., 1988, *From Neuropsychology to Mental Structure*, Cambridge, Engl.: Cambridge University Press. ♦
- Small, S. L., 1991, Focal and diffuse lesions in cognitive models, in *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum, pp. 85–90.

Limb Geometry, Neural Control

Francesco Lacquaniti, Mauro Carrozzo, Yuri P. Ivanenko, and Myrka Zago

Introduction

Sensorimotor transformations. Motor control involves the problem of transforming sensory inputs into motor outputs. The outputs are the motor commands that act on muscles and the inputs are derived from sensory feedback and efference copy of the motor commands. The simplest solution would be to encode similar parameters at both input and output levels, such as the kinetic parameters of muscle forces and joint torques. The problem of motor

control would then simplify to one of specifying patterns of time-varying muscle activity, and causality would dictate the ensuing limb motion. Some sensory inputs reflect kinetics (e.g., Golgi tendon organs), but many others reflect kinematics. For instance, muscle spindles encode changes in muscle length, and retinal or auditory signals encode target motion. Moreover, motor planning requires the ability to predict the limb motion resulting from the application of a given torque. Prediction is straightforward only for a limited class of movements, those restricted to a single joint; in

such cases, angular acceleration is simply proportional to joint torque. However, this simple relation does not hold for movements involving multijointed coordination.

Multijointed control. Consider the case of arm movements involving shoulder and elbow rotations. The activation of an elbow flexor will always contribute a flexor torque at the elbow, but the resulting elbow movement can be flexion, extension, or no motion at all, depending on the actively produced torque at the shoulder. In general, the relation between joint torque and angular acceleration of each limb segment depends in a complex manner on limb inertia, angular velocity, and limb geometrical configuration (Soechting and Flanders, 2002; see GEOMETRICAL PRINCIPLES IN MOTOR CONTROL). Limb geometry, in turn, affects limb inertia and determines the gravitational loads that need to be opposed. Because limb inertia, geometry, and velocity may all vary during a movement, the central nervous system (CNS) should be able to estimate each of these parameters by means of feedback or feedforward, to predict the motion resulting from the application of a given torque. However, the time delays in position and velocity feedback and the uncertainties in the estimate of the starting position of the limb may prevent accurate estimates.

Kinematic and kinetic variables. Although in principle, a coordinated motor action could be planned muscle by muscle, a more parsimonious solution is to plan more global goals at higher levels of organization and let the lower-level controllers specify the implementation details. Global goals often encompass both kinematics and kinetics. Thus, reaching or walking require the specification of the spatial location and contact force of the respective end point (hand or foot). According to hierarchical control, kinematic and kinetic plans could be fed as inputs to limb controllers; the latter would transform these plans into the appropriate motor commands. A kinematic plan could involve the mere specification of target position, and therefore of the limb end point that matches the target. In addition, however, the plan could include a specification of the path and law of motion of the end point from start to target. Kinematic transformations (inverse kinematics) convert the desired end-point trajectory into the angular motion of each limb segment. Dynamic transformations (inverse dynamics) convert limb motion into the appropriate commands to generate muscle forces and torques. A kinetic plan could involve the specification of the force field at the end point (see MOTOR PRIMITIVES), or it could specify directly the arm and muscle dynamics (Nakano et al., 1999).

Implementation problems. Despite the apparent simplicity of this cascade of events, their actual implementation can be very demanding from a computational standpoint and can result in substantial errors. In a redundant limb, such as the arm or leg, that involves many more degrees of freedom at the level of muscles and joints than at the end point, inverse transformations generally are ill defined. Unique solutions can be imposed by introducing kinematic and/or kinetic constraints, or by using optimization principles. In the following sections we briefly review some issues related to the kinematic aspects of limb geometry control for arm movements and for posture and gait.

Arm Movements

Sensorimotor Transformations for Reaching to a Target

End-point specification. To reach a visual target requires transforming information about target location into commands that specify the patterns of muscle activity that bring the hand to the target. Psychophysical evidence suggests that the specification of the final position of the hand depends on a cascade of sensorimotor transformations that remaps target location from the initial two-

dimensional (2D) retinal frame of reference to a three-dimensional (3D) binocular viewer-centered frame to arm- and hand-centered frames (Flanders, Helms Tillery, and Soechting, 1992; Gordon, Ghilardi, and Ghez, 1994; McIntyre et al., 2000; see also EYE-HAND COORDINATION IN REACHING MOVEMENTS). Remapping depends on the combination of retinal and extraretinal signals with somatic signals to update target and end-point spatial representations as the eyes, head, trunk, or limb move. Exactly how this remapping occurs is a matter of controversy, but an emerging view is that the frame of reference used to specify the end point may be task and context dependent. Moreover, different spatial dimensions of the end point (e.g., direction and distance) are not treated in a unitary manner but are processed in parallel and largely independently of each other, according to principles of modular organization (see Georgopoulos, 1991; Gordon et al., 1994).

Egocentric frames. The spatial patterns of hand errors have been studied extensively in pointing to actual or remembered targets that are presented visually or proprioceptively, the movements being performed with or without visual guidance (Baud-Bovy and Viviani, 1997; Flanders et al., 1992; Gordon et al., 1994; McIntyre et al., 2000). Constant and variable errors may reveal the presence of bias and random noise, respectively, in internal representations of the end point, whereas local distortions may reveal transformations between different frames of reference (McIntyre et al., 2000). Thus, in pointing to a continuously visible virtual target, the presence of radial noise and contraction of distance along the sightline indicates the representation of the target and limb end point in a 3D binocular viewer-centered frame, the anisotropy resulting from fusion of eye position signals with retinal disparity signals and from a coupling of uncompensated eye movements. When vision of the hand is prevented, head- and shoulder-centered distortions characterize end-point distribution and denote the transformation of information into the corresponding frames. The increase in head/shoulder-centered local contraction for increasing delays (up to 8 s) for movements performed toward previously visible then memorized targets indicates that memory storage of the intended end point is held within these frames, with separate storage of distance and direction, the distance information decaying faster than the directional information. Additional variability along the movement direction suggests that target information is combined with hand information to form a hand-centered vectorial plan of the intended movement trajectory as an extent and direction relative to the starting hand position (Gordon et al., 1994). Extent and direction also adapt differentially during motor learning (Krakauer et al., 2000). By contrast, when targets are presented kinesthetically instead of visually, subjects underestimate the perceived target distance relative to the shoulder (Baud-Bovy and Viviani, 1997).

Allocentric frames. The evidence reviewed so far indicates that, when targets are immersed in an otherwise neutral space, end-point position is specified in egocentric frames of reference, that is, relative to some body parts (eyes, head, or arm). However, when targets are embedded in a geometrically structured space, the visual or cognitive context can shape pointing errors and reveal the use of allocentric reference frames to represent end-point position. Thus, the final position of a pointing movement toward a remembered target is biased by the position of a surrounding frame (Bridgeman, Peery, and Anand, 1997; see DISSOCIATIONS BETWEEN DIFFERENT VISUAL PROCESSING MODES).

Limb Kinematics

Kinematic regularities. A different issue is whether kinematic plans also include a specification of the path and law of motion of the limb. Moreover, is limb kinematics planned at the level of the

end point or at the level of joint and limb segments? There is no consensus on these issues. A number of lawful relationships have been described for the kinematic trajectories of the hand in external space and of individual limb segments in the angular coordinates of the joints. Thus, the spatial trajectories of both the hand and the joints are essentially unaffected by wide changes in speed and load. In point-to-point movements the velocity profile of the hand tends to be bell-shaped, while the velocity profiles of shoulder and elbow angular motions tend to be temporally correlated. In curved movements (such as those of drawing and handwriting), the instantaneous tangential velocity of the hand is inversely related (by a power law) to the local curvature of the path (Lacquaniti, 1997).

Optimum principles. Although these kinematic regularities are compatible with the existence of a kinematic plan, they do not prove it. Nevertheless, kinematic optimization principles are able to predict the time course of reaching and drawing (see OPTIMIZATION PRINCIPLES IN MOTOR CONTROL). Optimization may involve end-point trajectory or joint angular trajectories. The minimum jerk principle, for instance, constrains hand reaching to follow a maximally smooth time course. Superposition of smooth harmonic oscillations of joint angular motion predicts the power law for drawing.

Kinematic regularities could also result from optimizing kinetic criteria instead of kinematic ones. Nakano et al. (1999) have been able to account for a variety of reaching trajectories by assuming that the commanded change in torque is minimized. In a similar vein, Soechting and Flanders (2002) showed that for reaching to a given target starting from different locations, the final posture of the arm minimizes the amount of work that must be done to transport the arm from start to end.

Planning Dynamic Interactions

In interceptive tasks, such as catching, information about the relative motion between limbs and objects must be preprocessed in order to plan the dynamic interaction in advance of its occurrence. Time, location, and momentum of the impact need be accurately estimated, and limb kinematics and kinetics controlled accordingly. A priori knowledge of the most likely path and law of motion of the object is used in conjunction with visual on-line information. Thus, catching movements are time-locked to time-to-contact computed by combining optic flow information with an internal estimate of the acceleration of gravity (Lacquaniti, 1997). Also, the compliant relation between hand and object appears to be internally modeled by the CNS, resulting in prospective tuning of limb geometry and compliance. This control scheme is adaptive: the response to the dynamic interaction predicted by the internal model is compared with the actual response of the limb (as monitored by kinesthetic and cutaneous signals), and the resulting error is used to calibrate the parameters of the neural controller and to update the internal model.

Processes of trajectory formation have also been shown to undergo adaptive changes in response to novel force fields experienced at the hand; adaptation is based on the progressive changes of the internal models of limb dynamics, initially defined in the intrinsic coordinates of the joints and muscles. Proprioceptive information is essential to maintain internal models (Ghez, Gardon, and Ghilardi, 1995). Indeed, deafferented (as a result of large-fiber neuropathies) subjects are unable to compensate for workspace anisotropies in limb inertia and produce pointing errors that are direction dependent.

Neural correlates

Positional and directional codes. Electrophysiological recordings from single neurons in frontal and parietal cortical areas have been

related to kinematic parameters of reaching in different frames of reference (see MOTOR CORTEX: CODING AND DECODING OF DIRECTIONAL OPERATIONS; REACHING MOVEMENTS: IMPLICATIONS FOR COMPUTATIONAL MODELS). In general, many neurons are broadly tuned to both the target location and the direction of the hand movement (Georgopoulos, 1991). Target location and movement direction can be defined in eye-centered or arm-centered coordinates, depending on the cerebral area and the task (Batista et al., 1999). The combination of retinal, eye-, and hand-related signals occurs at early stages of cortical processing, as revealed by the activity pattern of parieto-occipital neurons (Battaglia-Mayer et al., 2001). Directional signals of many such neurons for hand reaching are spatially congruent with those for eye saccades. These activity patterns are modulated by context (presence or absence of visual feedback of hand movement, memory delay), in agreement with the psychophysical data reported above.

Velocity codes. The time-varying changes in length and direction of the population vectors in M1 parallel the corresponding changes in the vector of tangential velocity in reaching and drawing. In drawing, the virtual changes in movement velocity predicted by the population vectors are related to path curvature by the same power law that applies to actual movement velocity.

Kinematics or kinetics? Neural codes of limb kinematics can occur independently of limb kinetics, or there may be an interaction. Thus, when the same movement is performed in the presence of different loads pulling the arm in different directions, some neurons are very sensitive to the applied load and appear to encode parameters related to movement kinetics, whereas other neurons are relatively insensitive to loads and appear to encode movement kinematics, and still other neurons fall in between, exhibiting both kinematic and kinetic tuning.

Limb geometry configuration. Another possible interaction is between the movement direction and the geometrical configuration of the limb and the participating muscles. Thus, the directional tuning of wrist muscles for flexion/extension and abduction/adduction depends on forearm pronation/supination. Directional tuning of some motor cortical neurons changes in parallel with the changes in directional tuning of the muscles (compatible with a kinetic code), whereas the tuning of other neurons changes in parallel with changes in posture (compatible with a kinematic code of limb geometrical configuration), and the tuning of still other neurons is not affected (compatible with a kinematic code of abstract movement direction independent of limb configuration and muscle activity; Kakei, Hoffman, and Strick, 1999).

Posture

Control of Limb Geometry in Posture

Postural control is often equated with stabilization of the body against gravity, i.e., a kinetic control problem. In fact there is now ample evidence that limb geometry can be controlled largely independent of the ground contact forces. Thus, when cats standing on a platform are pitched by variable angles, the resulting distribution of ground contact forces and joint torques is idiosyncratic to each animal and condition, whereas the geometry of both forelimbs and hindlimbs is much more stereotyped (Lacquaniti, 1997). The length and the angle of orientation of the limb axis relative to the vertical change little despite wide changes in platform tilt. Limb geometry is also preserved unmodified after the application of external loads, at the expense of marked changes in the distribution of weight and effort between forelimbs and hindlimbs. The CNS controls postural geometry directly rather than balance (distribution

of contact forces), presumably because it has learned that the preferred posture is stable under normal operating conditions. On the other hand, not only has posture largely evolved to oppose gravity for the maintenance of balance, it is also organized in a reference frame that is anchored to the direction of gravity. As was noted earlier, limb orientation is controlled relative to the vertical.

Modular Organization.

In contrast to limb orientation, limb length is not significantly affected by tilt of the visual surround or by application of abnormal somesthetic stimuli. This and the differential dynamic behavior of the changes in limb length and orientation in response to dynamic pitch suggest that these two geometrical variables might be controlled independently of each other. This modular organization is reminiscent of that reported above for the control of direction and distance in arm reaching. Note further that limb length and orientation are encoded independently in the responses of dorsal spinocerebellar neurons to applied changes in lower-limb posture in the anesthetized cat spinal cord (Poppele, Bosco, and Rankin, 2002).

Coordinate Transformations for the Control of Posture

Length and orientation specify the position of the foot relative to the hip in a global manner, leaving the detailed geometrical configuration undetermined. There is an additional processing stage that transforms limb length and orientation into the angular coordinates of the joints (Lacquaniti, 1997). The changes of these angles under both static and dynamic conditions are not independent, but covary close to one plane. The orientation of this plane is essentially the same in all animals (despite wide differences in their biomechanical parameters), and is also the same at the forelimbs and at the hindlimbs. The latter invariance is especially remarkable, considering that the forelimbs differ considerably from the hindlimbs in terms of the length and orientation of the individual corresponding segments. A related planar covariation has also been found in the case of whole-body motion in man (postural responses to external perturbations and anticipatory responses prior to voluntary trunk axial bending). Moreover, these kinematic strategies remain unchanged under microgravity, that is, in the absence of equilibrium constraints.

Neural Network Implementation of Coordinate Transformation

How might a nervous system learn to map limb length and orientation into joint angles? A forward model of the controlled system could be learned by monitoring both the input and the output of the system. The forward model would map the expected relationship between the set of joint angles and the resulting end-point position. This forward model would not need to be learned in its most general and exact manner by the postural system. In fact, although the forward mapping from joint angles to limb length and orientation is generally nonlinear, we find that the latter two parameters can be estimated simply and accurately using linear compounds of the joint angles, at least within the range of postures normally adopted. After the forward model has been learned, the desired movement trajectory (sequence of end-point positions) could be fed to the inverse model to derive the feedforward motor command (sequence of joint angles). The resulting error in end-point position is propagated through the forward model to derive the corresponding error in the motor command space (joint angle space). The latter error represents the signal to train the inverse model. Parameterized constraints, such as the planar constraint on the joint angles for the postural control, could be incorporated into

the learning procedure, and could thereby bias the choice of a particular inverse function.

Locomotion

Kinematic Coordination

A law of planar covariation also applies to the changes of elevation angles of lower limb segments during locomotion (Lacquaniti, Grasso, and Zago, 1999). The fact that similar laws of intersegmental coordination apply to the control of posture and locomotion is functionally significant, inasmuch as locomotion must ensure a forward progression compatible with dynamic equilibrium, adapting to potentially destabilizing factors (e.g., changes in body posture or load, uneven terrain, obstacles) in an anticipatory fashion by means of coordinated synergies of the whole body (see LOCOMOTION, VERTEBRATE).

In walking, the patterns of limb segment angular motion are remarkably simple and consistent. Each segment of the lower limbs oscillates forward and backward, with a waveform that mainly differs in timing and amplitude among different segments. The temporal changes of the elevation angles of lower limb segments do not evolve independently of each other, but they are tightly coupled. When the elevation angles are plotted one versus the others, they describe regular loops constrained close to a plane, common to both stance and swing phases. The specific orientation of the plane of angular covariation reflects the phase relationships between the elevation angles of the lower limb segments, and therefore the timing of the intersegmental coordination. Because the degrees of freedom of limb angular motion in the sagittal plane are reduced to two by the planar constraint, they match the corresponding degrees of freedom of linear motion of the center of body mass (CM).

Relation to Energy Expenditure

Saving the mechanical energy of the body during walking depends to a large extent on the exchange between the forward kinetic energy and the gravitational potential energy of CM. The selection of the elevation angles of each limb segment with respect to the direction of gravity and that of forward progression as the controlled variables may help predict the energetic consequences of the desired kinematics. Moreover, the planar covariation of the elevation angles is instrumental in reducing the degrees of freedom of limb motion to those of CM, where most mechanical energy is expended in walking. There is an additional, important mechanism embedded in the law of kinematic coordination that contributes to the control of mechanical energy expenditure. The net mechanical power tends to increase rapidly with speed, because the changes in potential energy are roughly independent of speed, whereas the changes in kinetic energy increase with speed, and therefore less and less energy is conserved by means of the energy exchange. However, there is a compensatory mechanism that reduces the oscillations of CM. The phase coupling between the instantaneous changes of the elevation angles of the limb segments shifts systematically with increasing speed both in humans and in cats. In humans, it has been shown that the phase shift translates into a reduction in the increment of the net mechanical power with increasing speed. This mechanism is not equally developed in all human subjects, however. Trained subjects generally exhibit a more pronounced phase shift with increasing speed than untrained subjects. Accordingly, the mechanical power output at intermediate and high speeds is significantly lower in the former than in the latter subjects.

Interaction Between Posture and Locomotion

Human erect locomotion is unique among living primates. Evolution selected specific biomechanical features that make human locomotion mechanically efficient. These features are matched by the motor patterns generated in the CNS. But what happens when humans walk stooped (as it happens in a low tunnel)? Are normal motor patterns of erect locomotion maintained, or are locomotor patterns completely reorganized? Walking has been compared in bent postures, either knee-flexed or knee- and trunk-flexed. These postures imply large differences in the position of the CM compared with its position in the standard erect posture: the CM is displaced downward and forward (outside the body), and its oscillations are reduced because the legs cannot fully extend. Thus, the exchange of kinetic and potential energy is much more limited than usual.

In bent posture, ground reaction forces differ prominently from those of erect posture, displaying characteristics intermediate between those typical of walking and those of running. Amplitudes and waveforms of the muscle activities also are deeply affected by the adopted posture. By contrast, the waveforms of the elevation angles along the gait cycle remain essentially unchanged, irrespective of the adopted postures. Thigh, shank, and foot angles covary close to a plane in all conditions, but the plane orientation is systematically different in bent versus erect locomotion. This is explained by the changes in the temporal coupling (phase shift) among the three segments.

An integrated control of gait and posture is made possible because these two motor functions share some common principles of spatial organization. Thus, the kinematic reference frame seems to be anchored to the vertical for both postural responses and locomotion. Also, the planar law of intersegmental kinematic coordination applies to both tasks.

Adaptation to Changes in Body Load

Body weight unloading is compatible with accurate control of limb kinematics in human locomotion. Changing the amount of body weight support (BWS) between 0% and 95% while subjects walk results in drastic changes in kinetic parameters but in limited changes in kinematic coordination. In particular, the peak vertical contact forces decrease proportionally to BWS; at 95% BWS they are 20-fold smaller than at 0% and are applied at the forefoot only. Also, there are considerable changes in the amplitude of EMG activity of most lower limb muscles and a complex reorganization of the pattern of activity of limb muscles. By contrast, the corresponding variation in the parameters that describe shape and variability of the foot path is very limited, always less than 30% of the corresponding values at 0% BWS. Moreover, the planar covariation of the elevation angles is obeyed at all speed and BWS values. At 100% BWS, subjects step in the air, their feet oscillating back and forth just above the treadmill but never contacting it. In this case, step-to-step variability of foot path is much greater than at all other BWS levels, but it is restored to lower values when minimal surrogate contact forces are provided during the "stance" phase. Thus, the detection of minimal contact forces is sufficient for accurate limb trajectory control.

Reversal of Walking Direction

Reversal of walking direction from forward to backward is a key test for studying locomotor patterns. According to the influential scheme put forth by Grillner, backward walking could be produced by switching the sign of the phase coupling among unit oscillators (see MOTOR PATTERN GENERATION). As a result, the controlled output patterns would simply be the time-reversed copy of those

of forward gait. If we consider the patterns of muscle activities, there is no way we can superimpose those of backward gait onto those of forward gait, irrespective of whether the waveforms are plotted in normal forward time or reversed in time (Lacquaniti et al., 1999). This is perhaps not so surprising given that the mechanical requirements of backward walking are very different from those of forward walking. Stance is characterized by an inverted plantigrade-digitigrade sequence in the two movement directions. Forward stance begins with heel contact and ends with toe-off. Backward stance begins with toe contact and ends with heel-off. The anatomical and functional asymmetry of foot and leg muscles along the anteroposterior axis also imposes different biomechanical constraints on forward and backward gait. Forward thrust is mainly provided by ankle plantar flexors, whereas the backward thrust is provided by hip and knee extensors.

However, the time-reversed waveforms of backward gait are almost perfectly superimposable onto those of forward gait (Lacquaniti et al., 1999). Accordingly, the planar covariation is the same; the loop is simply traversed in the opposite direction, owing to a switching of the thigh-shank phase. Thus, kinematic patterns follow Grillner's predictions of a phase switch among unit oscillators. It appears as though the same kinematic templates can be output by CPGs in either direct or time-reversed form (like a motor tape), depending on the direction (forward or backward) of gait.

Mechanisms for Kinematic Coordination in Locomotion

The planar law of intersegmental coordination may emerge from the coupling of neural oscillators between each other and with limb mechanical oscillators. Muscle contraction intervenes at variable times to reexcite the intrinsic oscillations of the system when energy is lost. The hypothesis that a law of coordinative control results from a minimal active tuning of the passive inertial and viscoelastic coupling among limb segments is congruent with the idea that movement has evolved according to minimum energy criteria.

It is known that multisegment motion of mammals locomotion is controlled by a network of coupled oscillators (CPGs; see HALF-CENTER OSCILLATORS UNDERLYING RHYTHMIC MOVEMENTS). Flexible combinations of unit oscillators give rise to different forms of locomotion. Interoscillator coupling can be modified by changing the synaptic strength (or polarity) of the relative spinal connections. As a result, unit oscillators can be coupled in phase, out of phase, or with a variable phase, giving rise to different behaviors, such as speed increments or reversal of gait direction (from forward to backward). Supraspinal centers may drive or modulate functional sets of coordinating interneurons to generate different walking modes (or gaits).

Although it is often assumed that CPGs control patterns of muscle activity, an equally plausible hypothesis is that they control patterns of limb segment motion instead. According to this kinematic view, each unit oscillator would directly control a limb segment, alternately generating forward and backward oscillations of the segment. Intersegmental coordination would be achieved by coupling unit oscillators with a variable phase. Intersegmental kinematic phase plays the role of global control variable previously postulated for the network of central oscillators. In fact, intersegmental phase shifts systematically with increasing speed both in humans and in cats. Because this phase shift is correlated with the net mechanical power output over a gait cycle, phase control could be used for limiting the overall energy expenditure with increasing speed. Adaptation to different walking conditions, such as changes in body posture, body weight unloading, and backward walking, also involves intersegmental phase tuning, as does the maturation of limb kinematics in toddlers.

Conclusion

There is ample evidence that movement trajectories are controlled for tasks as diverse as arm reaching, drawing, catching, and walking. Also, limb kinematics can be controlled independently of kinetics. Ill-defined inverse transformations from end point to joint coordinates can be solved by introducing kinematic constraints, such as the law of planar intersegmental coordination, or by means of optimization principles (see OPTIMIZATION PRINCIPLES IN MOTOR CONTROL). To simplify, control, and reduce errors, hybrid feedback/feedforward control schemes are presumably used whenever possible (see MOTOR CONTROL, BIOLOGICAL AND THEORETICAL). In addition internal models that map motor commands onto their sensory consequences and vice versa are used to improve estimates and to learn new tasks (see SENSORIMOTOR LEARNING). However, we still do not know how the kinematic control is obtained, and in particular, we ignore whether it arises from explicit trajectory planning or is derived implicitly from implementation of intrinsic neural dynamics.

Road Map: Mammalian Motor Control

Related Reading: Arm and Hand Movement Control; Geometrical Principles in Motor Control; Optimization Principles in Motor Control

References

- Batista, A. P., Buneo, C. A., Snyder, L. H., and Andersen, R. A., 1999, Reach plans in eye-centered coordinates, *Science*, 285:257–260.
- Battaglia-Mayer, A., Ferraina, S., Genovesio, A., Marconi, B., Squatrito, S., et al., 2001, Eye-hand coordination during reaching: II. An analysis of visuomanual signals in parietal cortex and of their relationship with parieto-frontal association projections, *Cereb. Cortex*, 11:528–544.
- Baud-Bovy, G., and Viviani, P., 1997, Pointing to kinesthetic targets in space, *J. Neurosci.*, 18:1528–1545.
- Bridgeman, B., Peery, S., and Anand, S., 1997, Interaction of cognitive and sensorimotor maps of visual space, *Percept. Psychophysics*, 59:456–469.
- Flanders, M., Helms Tillery, S. I., and Soechting, J. F., 1992, Early stages in a sensorimotor transformation, *Behav. Brain Sci.*, 15:309–362.
- Georgopoulos, A. P., 1991, Higher order motor control, *Ann. Rev. Neurosci.*, 14:361–377. ♦
- Ghez, C., Gordon, J., and Ghilardi, M. F., 1995, Impairments of reaching movements in patients without proprioception: II. Effects of visual information on accuracy, *J. Neurophysiol.*, 73:361–372.
- Gordon, J., Ghilardi, M. F., and Ghez, C., 1994, Accuracy of planar reaching movements: I. Independence of direction and extent variability, *Exp. Brain Res.*, 99:97–111.
- Kakei, S., Hoffman, D. S., and Strick, P., 1999, Muscle and movement representations in the primary motor cortex, *Science*, 285:2136–2139.
- Krakauer, J. W., Pine, Z. M., Ghilardi, M. F., and Ghez, C., 2000, Learning of visuomotor transformations for vectorial planning or reaching trajectories, *J. Neurosci.*, 20:8916–8924.
- Lacquaniti, F., 1997, Frames of reference in sensorimotor coordination, in *Handbook of Neuropsychology* (F. Boller and J. Grafman, Eds.), vol. 11, Amsterdam: Elsevier, pp. 27–64. ♦
- Lacquaniti, F., Grasso, R., and Zago, M., 1999, Motor patterns for walking, *News Physiol. Sci.*, 14:168–174.
- McIntyre, J., Stratta, F., Droulez, J., and Lacquaniti, F., 2000, Analysis of pointing errors reveals properties of data representations and coordinate transformations within the central nervous system, *Neural Computat.*, 12:2823–2855.
- Nakano, E., Imamizu, H., Osu, R., Uno, Y., Gomi, H., Yoshioka, T., and Kawato, M., 1999, Quantitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model, *J. Neurophysiol.*, 81:2140–2155.
- Poppele, R. E., Bosco, G., and Rankin, A. M., 2002, Independent representations of limb axis length and orientation in spinocerebellar response components, *J. Neurophysiol.*, 87:409–422.
- Soechting, J. F., and Flanders, M., 2002, Movement regulation, in *Encyclopedia of the Human Brain* (V. S. Ramachandran, Ed.), San Diego: Academic Press. ♦

Localized Versus Distributed Representations

Simon J. Thorpe

Introduction

What happens in the brain when you recognize a familiar stimulus such as your grandmother's face? Most researchers accept that recognition involves activating some sort of internal representation, but there is little agreement about how such representations are physically implemented in neuronal hardware. According to the localist coding view, recognizing an object involves activating neurons tuned to that particular object—an idea often described as “grandmother cell” coding. Alternatively, the presence of a particular object might never be made explicit at the single-cell level. Instead, the final representation of one's grandmother might be distributed across large populations of cells, none responding selectively to grandmothers alone.

Suppose we need to represent four different stimuli—green and red bars that can be either horizontal or vertical. Figure 1 illustrates three options: a local coding scheme using separate units to code each stimulus (Figure 1A), a semilocal scheme with color and orientation encoded separately (two active units are needed to represent each stimulus; Figure 1B), and a distributed coding scheme representing all four stimuli with just three units (Figure 1C). In the last case, someone listening to the response of any individual unit would have difficulty making sense of the activity. Such situations arise both in so-called Hopfield networks (see COMPUTING

WITH ATTRACTORS) and in the hidden layer of backpropagation-trained networks that have fewer units than the number of stimuli that need to be represented.

So, what does the brain do? It is likely that the brain uses a range of coding strategies; there are several possibilities. Few neuroscientists have gone so far as to suggest that individual neurons might explicitly represent particular objects, Jerzy Konorski and Horace Barlow being two notable exceptions (Konorski, 1967; Barlow, 1985). Most prefer some form of sparse representation in which a relatively low percentage of active cells represents each object but none is tuned to that particular object (see SPARSE CODING IN THE PRIMATE CORTEX), as in the semilocal scheme illustrated earlier. However, many connectionists have used localist representations to model phenomena that include word and letter perception (see CONNECTIONIST AND SYMBOLIC REPRESENTATIONS) (Grainger and Jacobs, 1998; Page, 2000), and although they generally insist that the units in their models are not real neurons, there is no obvious reason why such models might not map directly to the neural level.

How might we determine the nature of the strategy used by the brain to represent objects? In the first section of this article we examine neurophysiological evidence that both distributed coding and local coding are used in high-order visual areas. In the second section we examine some computational reasons for preferring representations that are more distributed or localist. Finally, we ex-

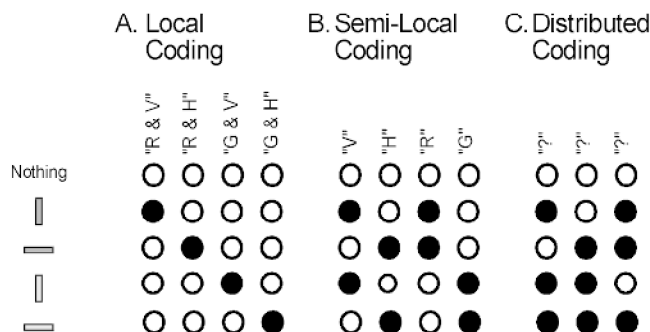


Figure 1. Three ways of representing four stimuli. Filled circles correspond to active units. *A*, Local coding. Each stimulus is explicitly coded by dedicated units. *B*, Semilocal coding. The different features—red (R), green (G), vertical (V), and horizontal (H)—are represented by separate units. *C*, Distributed coding.

amine how work on temporal coding schemes has changed the nature of the local versus distributed debate.

Neurophysiological Evidence

The best way to analyze how neurons represent objects is to record responses to a wide range of stimuli and determine their selectivity. This should be done at the highest levels of the sensory pathways, in areas such as inferotemporal (IT) cortex (see OBJECT RECOGNITION, NEUROPHYSIOLOGY). In the early 1970s, Charles Gross and his colleagues at Princeton described cells in IT with selectivity for complex shapes such as monkey paws. At the time, many scientists read such reports with incredulity, but by the early 1980s, researchers such as Bob Desimone, David Perrett, and Edmund Rolls had started to provide incontrovertible evidence for neurons responding selectively to faces (Perrett, Mistlin, and Chitty, 1987) (see FACE RECOGNITION; NEUROPHYSIOLOGY AND NEURAL TECHNOLOGY). Although most such cells respond to a range of faces, there have been reports of much higher degrees of selectivity. For example, one study described a cell that responded to only one of a set of 27 photographs of Japanese male faces (Young and Yamane, 1992).

Some researchers feel that the high selectivity for faces constitutes a special case, but highly selective responses to a range of visual stimuli have now been found. Nikos Logothetis and co-workers trained monkeys to respond to arbitrary “paper clip” forms from a range of views. Following training, some cells responded only to particular shapes seen from a limited range of viewing angles (Logothetis, Pauls, and Poggio, 1995). Rufin Vogels (1999) studied neuronal responses to photographs of natural objects in monkeys performing a categorization task and found that 17% responded to less than four out of a set of 60 images. And in a study in which 100 different photographs were presented repeatedly to a large number of IT cells, half of the cells responded to seven or less of the stimulus set (Tamura and Tanaka, 2001). Indeed, the most selective cell responded over five times more to the best stimulus (a photograph of a chair) than to the next best one, and showed significant responses to only three of the 100 photographs.

Some of the most intriguing data have come from single-cell recordings in humans undergoing investigation for intractable epilepsy. Such studies have provided tantalizing hints that in humans, too, neurons are remarkably selective to categories of visual stimuli that include faces, natural scenes and houses, famous people, and animals (Kreiman, Koch, and Fried, 2000).

Clearly, at least some IT neurons show strong selectivity. But there is also plenty of evidence for more broadly tuned cells re-

sponding to many different objects, and some authors have argued strongly for distributed coding (Rolls and Deco, 2002). However, the data are also perfectly consistent with a hybrid model using both distributed and local coding. It is important to realize that neuronal selectivity in IT appears strongly experience dependent, the most selective responses being found in monkeys trained with particular sets of stimuli many hours a day for weeks or even months. Furthermore, the response properties of IT cells are plastic: even a few seconds of visual exposure can change selectivity (Tovee, Rolls, and Ramachandran, 1996). It could be that while most cells are relatively broadly tuned, experience might produce small numbers of increasingly specialized cells. Although allowing specialized cells for all possible objects would be impossible, it might not be so unreasonable to devote neural hardware to the relatively small number of objects that are particularly critical. In the next section, we will look at some of the computational reasons why the brain might prefer to opt for more explicit local coding in certain cases.

Computational Pros and Cons

If it is possible to represent objects with a pattern of activity across a large population of cells, why would the brain ever move to more localist representations? After all, distributed codes have certain clear advantages. They allow large numbers of different stimuli to be represented with a relatively small number of units, as in the case of the ASCII code, where seven binary nodes can encode 128 (or 2^7) different characters (a localist version would need 128 nodes). By using relatively broad tuning, distributed coding guarantees some sort of activity pattern even for stimuli never experienced before, and it has been claimed that distributed codes have a greater capacity for generalization and graceful degradation in the event of damage to the system (see Rolls and Deco, 2002).

On the other hand, localist representations also have some computational advantages. For example, efficient learning requires organisms to estimate the relative frequencies of events in the outside world, which in turn requires some sort of counting mechanism. It could be considerably easier to estimate the frequency of events represented with a local code than when the representation is distributed across a large number of neurons (Gardner-Medwin and Barlow, 2001). Imagine trying to estimate the frequency of the character E by examining the frequency of activation of the individual bits of the ASCII code. The problem would be trivial with a neuron that responded every time the letter E was presented.

Localist representations are also largely immune to the classic “binding problem” that results when two or more different objects or events need to be represented simultaneously. All three coding schemes in Figure 1 are fine for representing the four different objects as long as only one object is present at a time, but when both a red vertical bar and a green horizontal bar are present, only the local representation can provide reliable information. For the semilocal code, there is no way to decide which color goes with which orientation, and the fully distributed code would be completely unable to generate a meaningful response.

Interestingly, detecting the presence of red vertical bars in a field composed of green vertical and red horizontal distractors is a difficult and time-consuming task. In contrast, even relatively complex visual forms such as animals can be easy to detect in complex natural scenes (see FAST VISUAL PROCESSING). Why would such high-level objects “pop out” when other, apparently simpler combinations of features pose such problems? One possibility is that only stimuli that are explicitly coded at the single-unit level can be processed in parallel. Our difficulty with certain stimulus conjunctions might stem from the fact that we rarely, if ever, need to use a simple combination of vertical and red to define an object. Perhaps units at the earlier levels of the visual system that would be

useful for explicitly coding such combinations are no longer sufficiently plastic in adults to allow such low-level features to be used. In contrast, richer combinations of features of the type that could be encoded in IT cortex might be easier to group together as a result of experience in adults.

The Impact of Temporal Coding

In this final section, we will examine how the debate over local versus distributed representations has been influenced by the development of ideas concerning temporal coding in neural processing. Many earlier discussions of coding schemes were based on the premise that neurons effectively use a firing rate code. For instance, in Barlow's 1972 formulation, firing rate represents the probability that a particular stimulus configuration is present. But the situation changes radically if we introduce the idea that information can also be contained in the temporal pattern of spikes. There is now considerable evidence that synchronization across populations of cells can help bind features together (see SYNCHRONIZATION, BINDING AND EXPECTANCY and DYNAMIC LINK ARCHITECTURE), and this could provide an alternative solution to the binding problem mentioned earlier. With the semilocal coding scheme in Figure 1B, the simultaneous presence of a red vertical bar and a green horizontal bar would cause problems. But synchronization could be used to link the different features together by making the "red" neuron fire synchronously with the "vertical" neuron, and the "green" neuron fire with the "horizontal" neuron. This is a clear case of a distributed representation, because analyzing the firing of the four single cells in isolation would fail to disambiguate the stimuli.

Another particularly clear example of a distributed code is rank-order coding that uses the order in which a population of neurons fire to encode information (Thorpe, Delorme, and Van Rullen, 2001). It uses the fact that the time taken for a neuron to reach threshold and fire a spike depends on the strength of the input: the stronger the input, the shorter the latency (Figure 2A). Under such conditions, the order of activation across a population of neurons can encode the stimulus (Figure 2B). In a sense, this is the ultimate illustration of a distributed code, because listening to each neuron on its own provides no information whatsoever about the stimulus. It is only when the relative ordering across neurons is taken into account that an observer can derive information about the stimulus.

Interestingly, decoding order-related information can be done quite simply using a biologically plausible circuit involving feed-forward excitatory synapses coupled with shunting inhibition. In one step, it is possible to go from a representation of the input pattern that is fully distributed to one in which the information is made explicit in the firing of a single cell. The power of the coding scheme is illustrated by recent results showing that simple feed-forward architectures based on these principles can produce outputs that are effectively grandmother cells, responding to different views of only one particular face (Delorme and Thorpe, 2001). This demonstrates that switching from a fully distributed to a local representation can be done efficiently and with limited hardware.

Discussion

The suggestion that our brains might contain "grandmother cells" is a very controversial one, but it is a suggestion that should not be rejected without good reason. In recent years, neurophysiological evidence for highly selective neurons has become increasingly difficult to ignore. Furthermore, several computational arguments make local coding attractive in certain situations. But it is also clear that it would be impracticable to use local coding to represent all objects. Common sense would seem to favor hybrid systems with both distributed and local representations, thus allowing the best

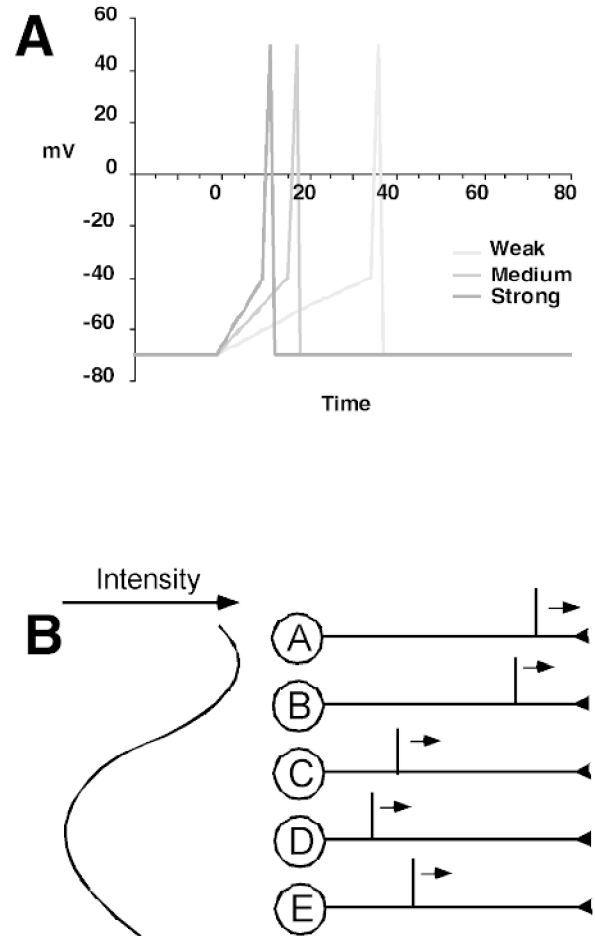


Figure 2. A distributed representation based on rank-order coding. *A*, Spike latency varies with the intensity of the input. *B*, With several units, the order of firing (in this case $A > B > E > C > D$) provides information about the input pattern.

of both worlds. Indeed, even if neurons at the top end of the visual system do explicitly code the presence of particular objects and stimuli, representations of these objects earlier in the system have to be distributed.

The fact that selective neurons are typically found in animals following extensive training suggests that the move toward more explicit, local representations is experience dependent. It might be that high-capacity sparse distributed representations are often sufficient; it may be worth dedicating neurons to the representation of stimuli that are particularly significant. Such local coding could increase reliability and allow such stimuli to be detected in parallel without the need for time-consuming attention-based search.

Road Map: Neural Coding

Related Reading: Connectionist and Symbolic Representations; Feature Analysis; Object Recognition, Neurophysiology; Sparse Coding in the Primate Cortex

References

- Barlow, H. B., 1985, The Twelfth Bartlett Memorial Lecture: The role of single neurons in the psychology of perception, *Q. J. Exp. Psychol. A*, 37:121–145
- Delorme, A., and Thorpe, S. J., 2001, Face identification using one spike

- per neuron: Resistance to image degradations, *Neural Netw.*, 14:795–803.
- Gardner-Medwin, A. R., and Barlow, H. B., 2001, The limits of counting accuracy in distributed neural representations, *Neural Comput.*, 13:477–504.
- Grainger, J., and Jacobs, A. M., 1998, *Localist Connectionist Approaches to Human Cognition*, Mahwah, NJ: Erlbaum.
- Konorski, J., 1967, *Integrative Activity of the Brain: An Interdisciplinary Approach*, Chicago: University of Chicago Press.
- Kreiman, G., Koch, C., and Fried, I., 2000, Category-specific visual responses of single neurons in the human medial temporal lobe, *Nature Neurosci.*, 3:946–953.
- Logothetis, N. K., Pauls, J., and Poggio, T., 1995, Shape representation in the inferior temporal cortex of monkeys, *Curr. Biol.*, 5:552–563.
- Page, M., 2000, Connectionist modelling in psychology: A localist manifesto, *Behav. Brain Sci.*, 23:443.
- Perrett, D. I., Mistlin, A. J., and Chitty, A. J., 1987, Visual neurons responsive to faces, *Trends Neurosci.*, 10:358–364.
- Rolls, E. T., and Deco, G., 2002, *Computational Neuroscience of Vision*, Oxford: Oxford University Press.
- Tamura, H., and Tanaka, K., 2001, Visual response properties of cells in the ventral and dorsal parts of the macaque inferotemporal cortex, *Cereb. Cortex*, 11:384–399.
- Thorpe, S., Delorme, A., and Van Rullen, R., 2001, Spike-based strategies for rapid processing, *Neural Netw.*, 14:715–725.
- Tovee, M. J., Rolls, E. T., and Ramachandran, V. S., 1996, Rapid visual learning in neurones of the primate temporal visual cortex, *Neuroreport*, 7:2757–2760.
- Vogels, R., 1999, Categorization of complex visual images by rhesus monkeys: Part 2. Single-cell study, *Eur. J. Neurosci.*, 11:1239–1255.
- Young, M. P., and Yamane, S., 1992, Sparse population coding of faces in the inferotemporal cortex, *Science*, 256:1327–1331.

Locomotion, Invertebrate

Randall D. Beer and Hillel J. Chiel

Introduction

Locomotion can be defined as an animal's ability to move its body along a desired path, making it fundamental to many other animal behaviors (Dickinson et al., 2000). Given the diversity of ecological niches that animals inhabit, and the variety of body plans that they possess, it is not surprising that their modes of locomotion are equally diverse. Types of locomotion include walking, swimming, flying, crawling, and burrowing.

Despite this diversity, certain common principles can be discerned. All locomotion systems must solve the twin problems of *support* and *progression*. The problem of support arises because in many modes of locomotion (e.g., flight), the gravitational attraction of the earth must be overcome. The problem of progression arises because an animal must generate propulsive forces that overcome not only its body's inertia, but also any drag from the density and viscosity of the medium or the friction of the substrate.

Both support and progression involve the generation of forces. This is accomplished by the contraction of muscles attached to either flexible hydrostatic skeletons or rigid skeletons. In addition, many animals have specialized body structures and appendages that facilitate locomotion, such as fins, wings, and legs. Thus, the detailed design of an animal's body is a crucial component of its locomotion system. As a result of the nature of these specializations, the problems of support and progression are rarely independent. Wings, for example, are used to generate both lift and propulsion in flying animals.

In order to provide support and progression, the movements of these specialized body structures must be coordinated by an animal's nervous system. The diverse modes of locomotion and the variety of body plans lead to equally diverse neural circuitry mediating locomotion. However, once again, certain basic principles can be discerned. Underlying many forms of locomotion are basic oscillatory patterns of movement generated by neural circuits that are referred to as *motor pattern generators* (MOTOR PATTERN GENERATION). Even when these circuits contain dedicated neurons that autonomously produce rhythmic outputs (so-called *central pattern generators*), this central pattern is often strongly shaped by sensory feedback, fundamentally involving the body and environment in the generation of a locomotor pattern. In fact, sensory feedback can play such a fundamental role that it sometimes makes no sense to speak of a distinct central pattern generator.

Researchers have begun to use computer modeling to understand the neural basis of locomotion. In contrast to most work in computational neuroscience, models of animal bodies are playing an important role in understanding locomotion systems. Increasingly, experimental evidence suggests that motor systems cannot be fully understood without considering the biomechanical properties of the bodies in which they are embedded (Chiel and Beer, 1997). Modeling of both an animal's body and the neural circuitry underlying its behavior has been termed *computational neuroethology* (NEUROETHOLOGY, COMPUTATIONAL). This chapter will focus on invertebrate locomotion systems for which quantitative modeling has been done, reviewing computer models of swimming, flying, crawling, and walking.

Swimming

In swimming, support is less of a problem than it is in other modes of locomotion. However, unless an animal is neutrally buoyant, it must still make efforts to keep from either sinking or rising. Progression requires much more effort as a result of the drag from water's density and viscosity. Thus, the bodies of swimming animals are streamlined. Swimming invertebrates utilize one of two mechanisms, either hydraulic propulsion or rhythmic undulations of the body.

Although models of swimming in leeches, mollusks, and nematodes have been constructed (Pearce and Friesen, 1988; Niebur and Erdős, 1991), perhaps the most modeled swimming system is not that of an invertebrate but that of a primitive vertebrate known as the lamprey. Lampreys swim using coordinated contractions of muscles on each side of the body. These contractions produce a traveling wave along the body, with a wavelength of approximately one body length across a wide range of swimming speeds. Although the lamprey possesses much of the basic vertebrate neural architecture, the experimental accessibility of its nervous system has allowed a level of neurophysiological analysis that is more typically applied to invertebrate systems. Earlier work used mathematical analysis and simulation of chains of model oscillators to study intersegmental coordination in the lamprey spinal cord (SPINAL CORD OF LAMPREY: GENERATION OF LOCOMOTOR PATTERNS; CHAINS OF OSCILLATORS IN MOTOR AND SENSORY SYSTEMS). Recent work has focused on more realistic models of the underlying

neuronal circuit and models of the relevant mechanics of the lamprey body and the water through which it swims.

Ekeberg and Grillner (1999) have reviewed much of the recent work in this area. The rhythm-generation circuit consists of populations of motor neurons, excitatory interneurons, and two distinct types of inhibitory interneurons repeated in each segment. Models of this circuitry have demonstrated that oscillations are relatively easy to generate, but details of the pattern (e.g., burst termination) depend on biophysical details of the nerve cells. The generation and propagation of the traveling wave along the segments has been studied by coupling chains of model segmental oscillators. This work revealed that if the rostral segments receive stronger excitation, they become the source of the traveling wave, and variation of this extra excitation allows the spatial wavelength of the swimming pattern to be controlled separately from its temporal frequency.

Mechanical aspects of swimming have been investigated by coupling pattern-generation circuitry to a segmented body model actuated by linear viscoelastic model muscles and embedded in a model of the static drag force produced by the surrounding water. By varying the level and asymmetry of tonic input, this neuromechanical model could produce swims at a range of speeds, turns, and rolls. In addition, two kinds of sensory feedback have been modeled. Incorporating feedback from intraspinal stretch receptors led to improved robustness of the swim pattern against unpredictable changes in water flow. Feedback from vestibular receptors was incorporated in order to model roll and pitch stabilization.

Flying

In many ways, flying is similar to swimming. However, because of the much lower density of air, considerably faster motions are required for powered flight than for swimming. While quasi-steady-state aerodynamic analyses of the sort used to understand aircraft have been successfully applied to larger animals, they have not been very successful for small flying insects. According to steady-state theory, many insects should be unable to generate sufficient lift to hold themselves aloft!

A recent model by Dickinson and colleagues has begun to shed considerable light on insect flight (Dickinson, Lehmann, and Sane, 1999). Because of the delicate size and high speed of insect wings, direct measurement of the forces involved is extremely difficult. For this reason, a robotic model was used to explore unsteady flows during hovering by the fruit fly *Drosophila melanogaster*. The model was submerged in mineral oil and scaled both in space and time so as to reproduce the Reynolds number (ratio of inertial to viscous forces) relevant to small insects flying in air. Dickinson and colleagues found that three major mechanisms contributed to lift generation in the model. First, vortices formed at the leading edge of the wing produce lift during much of the power stroke. Second, additional lift is produced by circulation of air around the wings resulting from rapid rotation at the beginning and end of each stroke. Third, further forces are produced at the start of each upstroke and downstroke as a result of collisions of the wings with the swirling wake produced by the previous stroke, a mechanism termed *wake capture*. Because of the sensitivity of the latter two mechanisms to the timing of wing rotation, the model suggests that the control of small details of wing motion can be used in steering flight.

Crawling

In crawling, locomotion occurs along the bottom surface of an aquatic environment or the surface of the earth via rhythmic contact between the body and the substrate. Invertebrates generate propulsive forces for crawling by changing body shape in one of three

ways: contract-anchor-extend (as in the leech), pedal locomotion (as in molluscs), or peristaltic locomotion (as in earthworms). Crawling invertebrates typically utilize either hydrostatic skeletons or muscular hydrostatic structures to accomplish these movements.

A detailed neuromechanical model of crawling in the leech has been constructed by Kristan et al. (2000). This model assumes that the cross-sectional geometry of each body segment is elliptical, that the volume of body segments remains constant during movement, and that the animal's shape minimizes total potential energy. Kristan et al.'s simulations incorporate relatively realistic models of the circular and longitudinal muscles found in the leech body wall. Driving the model body with activation patterns deduced from the kinematics of intact animals produces crawling movements that are considerably more realistic than those produced by activation patterns derived from reduced preparations. These results suggest that sensory feedback plays a critical role in providing appropriate timing of activation of longitudinal and circular muscles.

Walking

In legged animals, the body is raised above the ground and propelled by a sequence of leg movements. During walking, each leg cycles between a *stance phase*, in which the leg is providing support and propulsion, and a *swing phase*, in which the leg is off the ground and swinging forward. Swing phase duration is often nearly constant, while stance phase duration varies considerably with the speed of progression. Because the legs provide both support and propulsion and must be lifted after each stance, their movements must be coordinated so that the center of mass of the body remains within a polygon of support formed by the stancing legs (static stability). Otherwise, the animal must dynamically stabilize its body. Another coordination problem arises because adjacent legs must not interfere with one another. In many-legged animals, avoiding interference between adjacent legs is the crucial coordination problem, whereas the maintenance of stability is more important for animals with fewer legs.

Insect locomotion is remarkably flexible and robust. Insects can walk over a variety of terrains, as well as vertically and upside-down. In addition, they can also adapt their gait to the loss of up to two legs without severe degradation of performance (Delcomyn, Chapter 2 in Beer, Ritzmann, and McKenna, 1993) and sometimes even utilize dynamically stable gaits (Full, Chapter 1 in Beer et al., 1993). Most modeling has focused on statically stable walking across flat, horizontal surfaces. Even under these conditions, insects exhibit different gaits depending on their speed of locomotion.

Slowly walking insects show distinct *metachronal waves* on each side of the body: each leg begins its swing immediately following the termination of the swing of the leg behind it, with a 180° phase relationship between the pair of legs in each segment. Fast-walking insects utilize a *tripod gait*, in which the front and back legs on each side of the body step in unison with the middle leg on the opposite side. In one of the earliest theoretical models of insect walking, Wilson (1966) suggested that the entire range of observed insect gaits could be explained by assuming that fixed, antiphasic metachronal waves on each side of the body increasingly overlap as walking speed increases.

We developed a neural network model based on work by Pearson and colleagues on the neural organization of the American cockroach's walking system (Beer and Chiel, Chapter 12 in Beer et al., 1993). In this model, each leg controller has a pacemaker neuron whose output rhythmically oscillates due to a voltage-dependent intrinsic current. These pacemakers implement the swing burst-generators that Pearson hypothesized. A pacemaker burst initiates a swing by inhibiting the foot and backward swing motor neurons and exciting the forward swing motor neurons, causing the foot to lift and the leg to swing forward. Between bursts, the foot is down

and tonic excitation from a command neuron moves the leg backward. Feedback from two sensors that signal when a leg is nearing its extreme forward or backward position fine-tunes pacemaker output. Forward angle sensor inhibition encourages burst termination, whereas backward angle sensor excitation encourages burst initiation. The forward angle sensor also makes direct connections to the motor neurons, modeling leg reflex pathways described by Pearson.

In order to generate statically stable gaits, the swings of the individual legs must be coordinated in some way. Following Pearson, we inserted mutually inhibitory connections between the pacemaker neurons of adjacent legs. We also added an entrainment mechanism for generating metachronal waves: slightly increasing the angle ranges of the rear legs lowers the burst frequency of the rear pacemakers, causing the pattern generators on each side of the body to phase-lock into a stable metachronal relationship.

In simulations of this circuit in a kinematic hexapod body model, a continuous range of statically stable gaits similar to those described by Wilson (1966) were observed. This range of gaits was produced simply by varying the tonic level of excitation of the command neuron. Smooth transitions between gaits could be generated by continuously varying this excitation. We found that the ability of this circuit to generate statically stable gaits was quite robust to lesions. For example, removing any single sensor or interpacemaker connection did not generally disrupt locomotion. These studies also demonstrated that sensory feedback was crucial for the maintenance of the slower metachronal gaits, but was relatively unimportant in the tripod gait.

The stick insect *Carausius morosus* has also been a major focus of legged locomotion research. Cruse (1990) reviewed leg coordination influences in both the stick insect and the crayfish *Astacus leptodactylus*. In the stick insect, there are three major influences: (1) a swinging leg inhibits the swing of a more anterior leg; (2) when a leg begins its stance phase, it excites the swing of a more anterior leg; and (3) as a stancing leg nears the end of its stance, it increasingly excites the swing of a more posterior leg. Some of these influences also operate between pairs of legs in the same segment.

Dean (1991) simulated these and other coordination mechanisms. The pattern generator for each leg was modeled as a relaxation oscillator with two states corresponding to stance and swing. The positions of each of the six legs were the state variables for a kinematic model of walking. The coordination mechanisms modified the position at which an affected leg began its swing, with inhibitory influences producing a posterior shift and excitatory influences producing an anterior shift. Dean's simulations demonstrated that these coordination mechanisms were sufficient to generate a continuous range of gaits, including the wave gait at low stepping frequencies and the tripod gait at high stepping frequencies. The model also exhibited distinct asymmetries in stepping pattern observed in the stick insect, in which the phase relationship between legs in the same segment is consistently lower or higher than 180°. A good review of earlier models of stick insect walking can also be found in Dean (1991).

Dean also explored the robustness of these coordinating mechanisms to various perturbations, including variations in starting configurations, perturbations of individual leg velocities, and obstructions to the swing of individual legs. He found that the gaits generated by these mechanisms were quite robust to such perturbations and that, in most cases, the model's responses were similar to those of the insect. Discrepancies between the model and the insect could be traced to the need for dynamic variables in addition to kinematic ones. Dean varied the strength and form of the coordination mechanisms. He found that influence (3) was the most important to maintaining proper coordination due to its graded na-

ture, though the model was quite robust to substantial variations in the strengths of individual mechanisms.

Biorobotics

The remarkable flexibility and robustness of animal locomotion has intrigued roboticists. Biologically inspired locomotion controllers offer a number of advantages over more classical approaches, including their distributed nature, their robustness, and their computational efficiency. Likewise, robots can serve as an important new modeling methodology for testing biological hypotheses. Thus, a number of researchers have begun to explore the interface between biology and robotics (Beer et al., 1998; Webb, 2000). Raibert and Hodgins (Chapter 14, in Beer et al., 1993) have argued for the importance of leg and actuator design in locomotion, designing a series of dynamically stable hopping and running robots based on the biomechanical design of animal limbs. For example, we implemented both the locomotion circuit and the stick insect coordination mechanisms described previously in hexapod robots and found that they could generate a range of gaits similar to those observed in simulations and were equally robust to perturbations (Beer et al., 1997), and more recent work has successfully incorporated significantly more biological realism into the latest robot (Quinn and Ritzmann, 1998). Thus, models of animal locomotion may not only yield insights into the neural control of motor behavior, but may also have significant technological applications.

Discussion

We have touched on several successful examples of quantitative modeling of locomotion. It is notable that the different simulations utilize very different neural models. More fundamentally, it is striking that very different neural architecture can be utilized to generate locomotion. Undoubtedly, this variety is a result of the diverse body plans of animals and the many different ecological niches that they occupy. One consistent theme that does emerge, however, is the complex interplay of sensory input and central circuitry in the generation of locomotion. This complex interplay is responsible for the adaptive flexibility of animal locomotion.

Road Map: Motor Pattern Generators

Related Reading: Biologically Inspired Robotics; Chains of Oscillation in Motor and Sensory Systems; Half-Center Oscillators Underlying Rhythmic Movements; Locomotion, Vertebrate; Locust Flight: Components and Mechanisms in the Motor; Spinal Cord of Lamprey: Generation of Locomotor Patterns

References

- Beer, R. D., Quinn, R. D., Chiel, H. J., and Ritzmann, R. E., 1997, Biologically-inspired approaches to robotics, *Comm. ACM*, 40:31–38.
- Beer, R. D., Chiel, H. J., Quinn, R. D., and Ritzmann, R. E., 1998, Bi-robotic approaches to the study of motor systems, *Curr. Op. Neuro.*, 8:777–782. ♦
- Beer, R. D., Ritzmann, R. E., and McKenna, T., Eds., 1993, Biological neural networks in invertebrate neuroethology and robotics, Academic Press.
- Chiel, H. J., and Beer, R. D., 1997, The brain has a body: Adaptive behavior emerges from interactions of nervous system, body and environment, *Trends Neurosci.*, 20:553–557.
- Cruse, H., 1990, What mechanisms coordinate leg movement in walking arthropods? *Trends Neurosci.*, 13:15–21.
- Dean, J., 1991, A model of leg coordination in the stick insect, *Carausius morosus*. II. Description of the kinematic model and simulation of normal step patterns, *Biol. Cybern.*, 64:393–402.
- Dickinson, M. H., Farley, C. T., Full, R. J., Koehl, M. A. R., Kram, R., and Lehman, S., 2000, How animals move: An integrative view, *Science*, 288:100–106. ♦

- Dickinson, M. H., Lehmann, F.-O., and Sane, S. P., 1999, Wing rotation and the aerodynamic basis of insect flight, *Science*, 284:1954–1960.
- Ekeberg, O., and Grillner, S., 1999, Simulations of neuromuscular control in lamprey swimming, *Phil. Trans. R. Soc. Lond. B*, 354:895–902.
- Kristan, W. B., Jr., Skalak, R., Wilson, R. J. A., Skierczynski, B. A., Murray, J. A., Eisenhart, F. J., and Cacciatore, T. W., 2000, Biomechanics of hydroskeletons: Studies of crawling in the medicinal leech, in *Biomechanics and Neural Control of Posture and Movement* (J. M. Winters and P. E. Crago, Eds.), New York: Springer-Verlag, pp. 206–218.
- Niebur, E., and Erdős, P., 1991, Theory of the locomotion of nematodes: Dynamics of undulatory progression on a surface, *Biophys. J.*, 60:1132–1146.
- Pearce, R. A., and Friesen, W. O., 1988, A model for intersegmental coordination in the leech nerve cord, *Biol. Cybern.*, 58:301–311.
- Quinn, R. D., and Ritzmann, R. E., 1998, Construction of a hexapod robot with cockroach kinematics benefits both robotics and biology, *Conn. Sci.*, 10:239–254.
- Webb, B., 2000, What does robotics offer animal behaviour? *Anim. Behav.*, 60:545–558. ♦
- Wilson, D. M., 1966, Insect walking, *Annu. Rev. Entomol.*, 11:103–122.

Locomotion, Vertebrate

Auke Jan Ijspeert

Introduction

Locomotion is a fundamental skill for animals. It is required for a large variety of actions, such as finding food, encountering a mate, and escaping predators. Among the various forms of vertebrate locomotion are swimming, crawling, walking, flying, and the more idiosyncratic movements such as hopping, brachiation, and burrowing.

Animal locomotion is characterized by rhythmic activity and the use of multiple degrees of freedom (i.e., multiple joints and muscles). In vertebrates, motion is generated by the musculoskeletal system, in which torques are created by antagonistic muscles at the joints of articulated systems composed of rigid bones. All types of vertebrate locomotion rely on some kind of rhythmic activity to move forward: undulations or peristaltic contractions of the body, oscillations of fins, legs, or wings. As the animal rhythmically applies forces to the environment (ground, water, or air), reaction forces are generated that move the body forward.

This type of locomotion is in contrast to the motion of most man-made machines, which usually relies on few degrees of freedom (e.g., a limited number of powered wheels, propellers, or jet engines) and continuous rather than rhythmic actuation. From a technological point of view, animal locomotion is significantly more difficult to control than most wheeled or propelled machines. The oscillations of the multiple degrees of freedom need to be well coordinated to generate efficient locomotion. However, as can be observed from the swimming of a dolphin or the running of a goat over irregular terrain, animal locomotion presents many interesting features, such as energy efficiency (for swimming) and agility. The next sections review the neural and mechanical mechanisms underlying vertebrates' fascinating locomotor abilities.

Neural Control of Locomotion

Despite diversity in types of locomotion, the general organization of the vertebrate locomotor circuit appears to be highly conserved. Locomotion is controlled by the interaction of three components: (1) spinal central pattern generators (CPGs), (2) sensory feedback, and (3) descending supraspinal control. The combination of these three components is sometimes called the motor pattern generator (MPG).

Central Pattern Generators

Central pattern generators are circuits that can generate rhythmic activity without rhythmic input (see HALF-CENTER OSCILLATORS UNDERLYING RHYTHMIC MOVEMENTS and MOTOR PATTERN GEN-

ERATION). The rhythms can often be initiated by simple tonic (i.e., nonoscillating) electrical or pharmacological stimulation. In vertebrates, the CPGs are located in the spinal cord and distributed in different oscillatory centers. In the lamprey, for instance, the swimming CPG is a chain of approximately 100 segmental oscillators distributed from head to tail (see CHAINS OF OSCILLATORS IN MOTOR AND SENSORY SYSTEMS and SPINAL CORD OF LAMPREY: GENERATION OF LOCOMOTOR PATTERNS). In tetrapods, the locomotor CPG appears to be composed of different centers, one for each limb, that are themselves decomposed into different oscillatory subcenters for each joint (Grillner, 1981). Recent evidence from intracellular recordings in the mudpuppy suggests that joint subcenters can be decomposed even further into distinct oscillatory centers for flexor and extensor muscles (Cheng et al., 1998).

Experiments in completely isolated spinal cords and in deafferented animals (i.e., animals without sensory feedback) have shown that the patterns generated by the CPG are very similar to those recorded during intact locomotion. This demonstrates that sensory feedback is not necessary for generating and coordinating the oscillations underlying locomotion during stationary conditions.

Sensory Feedback

Although sensory feedback is not necessary for rhythm generation, it is essential for shaping and coordinating neural activity with actual mechanical movements. The main sensory feedback to the CPGs is provided by sensory receptors in joints and muscles (see MOTOR CONTROL, BIOLOGICAL AND THEORETICAL). Rhythmically moving the tail or a limb of a decerebrate vertebrate is often sufficient to initiate the rhythmic patterns of locomotion. The frequency of oscillations then matches that of the forced movement, illustrating the strong influence of peripheral feedback on pattern generation.

Sensory feedback is especially important in higher vertebrates with upright posture such as mammals (as opposed to vertebrates with sprawling postures, like certain amphibians and reptiles), because the limbs of those vertebrates play an important role in posture control—supporting the body—in addition to locomotion.

A whole set of reflexes exists to coordinate neural activity with mechanical activity. One example is the stretch reflex, which generates the contraction of a muscle when the muscle is lengthened and which therefore helps maintain posture. The reflex pathways often share many of the interneurons that participate in locomotion control, and the action of reflexes is therefore not fixed. During locomotion, the action of reflexes can be modulated by central commands and in some cases even reversed, depending on the timing within the locomotor cycle (see Pearson and Gordon, 2000, and

SENSORIMOTOR INTERACTIONS AND CENTRAL PATTERN GENERATORS for reviews).

Descending Supraspinal Control

Locomotion is initiated and modulated by descending pathways from diencephalic and mesencephalic locomotor centers. (For reviews, see Donkelaar, 2001, and Rossignol in Rowell and Shepherd, 1996, chap. 5). Some of these pathways are direct; an example is the pathway from the vestibular nuclei and the cerebellum to the spinal neurons. Other pathways are relayed by centers in the brainstem, in particular the red nucleus and the reticular nuclei. In all vertebrates, the reticulospinal tract plays a crucial role in generating the drive for the basic propulsive body and limb movements. In the lamprey, for instance, reticulospinal neurons control both the speed and direction of locomotion (Grillner et al., 1995). In mammals, additional direct pathways exist between the motor cortex and the spinal cord—the corticospinal tracts. These tracts are unique to mammals and play an important role in visuomotor coordination, such as accurate foot placement in uneven terrain.

Interestingly, the input signals to the brainstem do not need to be complex to generate locomotion. It has been known since the 1960s that simple electrical stimulation of the brainstem initiates the walking gait in a decerebrate cat, and progressively increasing the amplitude of the stimulation leads to an increase in the oscillation frequency, accompanied by a switch from walking to trotting and eventually to galloping (Shik, Severin, and Orlovsky, 1966). This demonstrates that the brainstem and the spinal cord contain most of the circuitry necessary for locomotion, including complex phenomena such as gait transitions (see GAIT TRANSITIONS).

The Biomechanics of Locomotion

Locomotion is the result of an intricate coupling between neural dynamics and body dynamics, and many fundamental aspects of locomotion control, including gait transition, control of speed, and control of direction, cannot be fully understood by investigating the locomotor circuit in isolation from the body it controls. A body has its own dynamics and intrinsic frequencies with complex non-linear properties, to which the neural signals must be adapted for efficient locomotion control. As observed by roboticist Marc Raibert, the central nervous system (CNS) does not control the body, it can only make suggestions.

The body is a redundant system, with many muscles per joint and several muscles acting on more than one joint. Muscles serve as actuators, brakes, stiffness regulators, and stores of elastic energy. During locomotion, the frequencies, amplitudes, and phases of the signals sent to the multiple muscles must be well orchestrated. In most vertebrates, complex coordination is required not only between different joints and limbs but also between antagonist muscles, which combine periods of co-activation for modulating the stiffness of the joint and periods of alternation for actuating the joint.

In legged locomotion, the dynamics of a leg can be approximated by a pendulum model during walking and by a spring-mass model during running. These models allow one to relate several features, such as resonance frequencies, to the length and stiffness of the legs, and are able to describe the mechanics of legged locomotion surprisingly well in many animals.

The importance of the mechanical properties of the body is illustrated by research on passive walkers. Passive walkers are legged machines (some with knees and arms) that transform potential energy from gravity into kinetic energy when walking down a gentle slope. When correctly designed, these machines do not require any actuation or control for generating a walking gait, which in some cases can be strikingly human-like.

Numerical Simulations of Locomotor Circuits

Although the general organization of the vertebrate locomotor circuit is known, much work remains to be done to elucidate how its different components are implemented and how they interplay to generate the complex patterns underlying locomotion. This is a complex task because (1) these patterns are due to the interaction of the CNS and the body in movement, (2) numerous neurons in the brainstem and the spinal cord are involved, and (3) in most vertebrates, the same circuits appear to be involved in generating very different patterns of activity (e.g., different gaits in tetrapods). For the moment, the best decoded locomotor circuits are probably the swimming circuits in the lamprey and the frog embryo. For other vertebrates, in particular tetrapods, significant parts of the structure and functioning of the locomotion circuitry remain unknown.

Numerical simulations have an important role to play in evaluating whether a potential model of a neural circuit is adequate and sufficient to reproduce the rhythmic patterns observed through intracellular and/or EMG measurements. Several important issues can be investigated in simulation, such as the general stability of the patterns and the effect of modulating the tonic drive on the frequencies and phases of the oscillations. Simulations do not need to be restricted to the CNS. An interesting approach to understanding locomotion control is to couple the simulations of the locomotor circuits to physics-based simulations of the body (or to a robot). Such *neuromechanical* simulations are particularly useful because they embed the neural circuits in a body in interaction with the environment, therefore allowing one to close the sensing-acting loop and to investigate the complete resulting motor patterns (as opposed to only the patterns produced by the isolated CPGs).

Some Models of Vertebrate Locomotor Systems

This section presents some results of modeling of vertebrate locomotion, with a special focus on neuromechanical simulations.

Swimming

Vertebrate swimming has been most studied in the lamprey (see SPINAL CORD OF LAMPREY: GENERATION OF LOCOMOTOR PATTERNS), an eel-like fish using *anguilliform* swimming, in which a traveling wave is propagated along the whole elongated body. Ekeberg developed a neuromechanical simulation composed of a connectionist neural network representing the lamprey's 100-segment spinal locomotor circuit and a simplified model of the body in interaction with water (Ekeberg, 1993). The neural network produces oscillating activity when tonic input is provided to the neurons, with the frequency of oscillation being proportional to the level of excitation. When extra excitation is provided to the most rostral (i.e., closest to the head) segments, a traveling wave is propagated from head to tail. The extra excitation determines the wavelength, independent of the frequency. With these settings, the model therefore replicates the fact that a swimming lamprey can cover a large range of frequencies while maintaining the wavelength constant at approximately one body length.

The mechanical simulation is a two-dimensional articulated rigid body actuated by muscles simulated as spring and dampers. Although the hydrodynamics of the model is simplified, it produces swimming gaits very similar to those of lamprey swimming (Figure 1). The mechanical simulation allowed Ekeberg to investigate the effect of modulating the locomotor pattern on the speed and direction of locomotion, as well as the effect of sensory feedback from spinal stretch-sensitive cells. The model demonstrated that the speed of swimming can be varied by changing the frequency of oscillation through the level of tonic input, whereas the direction

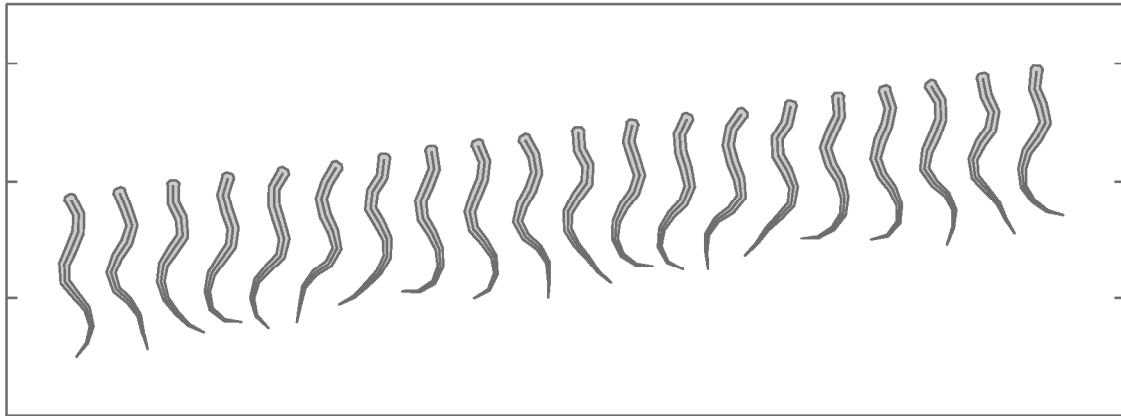


Figure 1. Neuromechanical simulation of lamprey swimming. (Reimplementation by the author of the model presented in Ekeberg, Ö, 1993, A combined neuronal and mechanical model of fish swimming, *Biol. Cybern.*, 69:363–374.)

of swimming can be varied by applying asymmetric tonic drive between left and right sides of the locomotor circuit.

Vertebrate swimming has inspired several underwater vehicles, such as eel-like robots that use anguilliform swimming (REEL, at the University of Pennsylvania) and a lamprey-based undulatory robot (at the Marine Science Center of Northeastern University), and carangiform swimming in the RoboTuna (at the Massachusetts Institute of Technology).

From Swimming to Walking

One of the most important changes during vertebrate evolution has been the transition from aquatic to terrestrial habitats. Our own work investigated the transition from swimming to walking in the salamander, an animal that is believed to be one of the modern animals closest to the first vertebrates that made this transition during evolution.

The salamander swims like a lamprey by propagating an undulation from head to tail. On ground, it switches to a stepping gait, usually with the phase relation of a trot. Although the locomotor circuit of the salamander has not yet been decoded, it has been found to share many similarities with the swimming circuit of the lamprey (Cohen, 1988; Delvolvé, Bem, and Cabelguen, 1997).

Our work sought to demonstrate that a lamprey-like swimming circuit could be extended to produce the swimming and stepping gaits of the salamander, with, in particular, a traveling wave along the body during swimming and a standing wave during stepping. The neural configuration of the model is illustrated in Figure 2. It is composed of a lamprey-like body CPG, extended by forelimb and hindlimb CPGs (Ijspeert, 2001). These limb centers have been identified just rostral to the anterior and posterior girdles, respectively. The mechanical simulation was an extension of Ekeberg's model of the lamprey (see Ijspeert, 2001, for a detailed description).

The model is able to (1) generate stable traveling waves and standing waves, depending on simple tonic input, (2) quickly switch between them, and (3) coordinate body and limb movements so as to produce swimming and walking gaits very similar to those recorded in salamanders. Gait transition is obtained as follows: when only the body CPG receives tonic input, the limb CPGs remain silent (limbs are maintained tonically against the body) and the body CPG produces a traveling wave that propels the salamander forward in water, whereas when tonic input is applied to both the body CPG and the limb CPGs, the body CPG is forced by the limb CPGs to produce a standing wave for stepping. The body then makes a standing S-shaped wave with the nodes at the girdles that

is coordinated with the movements of the limbs so as to increase the reach of the limbs during the swing phase (Figure 3, bottom).

Much as in Ekeberg's model of the lamprey, the speed and direction of locomotion can be modulated by respectively varying the level and the asymmetry (between left and right) of tonic input applied to the CPGs. Experiments involving the tracking of a randomly moving target show that locomotion is stable even when the input signals change rapidly and continuously (Ijspeert and Arbib, 2000). In collaboration with Richard Woesler and Gerhard Roth, we are currently extending this work to investigate visuomotor coordination (see VISUOMOTOR COORDINATION IN SALAMANDER).

Quadruped Locomotion

Quadruped locomotion in vertebrates has evolved from the sprawling posture found in salamanders and lizards to the upright posture found in mammals. During that evolution, the limbs gradually moved under the body, and movements in the body evolved from lateral to mainly sagittal (i.e., ventrodorsal) undulations.

The upright posture means that limbs serve both for locomotion and for maintaining balance. Gaits can either be *statically stable*, in which the center of mass is maintained at all times above the polygon formed by the contact points of the limbs with the ground, or *dynamically stable*, when this rule is not maintained at all times and stability is achieved as a limit cycle that balances moments, gravitational forces, and inertial forces over time. Depending on the phase relation between limbs, a large variety of gaits can be distinguished, such as the walk, the trot, the pace, and the gallop. Mammals can usually switch between these gaits very quickly (see GAIT TRANSITIONS).

The neural mechanisms underlying quadruped locomotion have not yet been decoded, but investigations in the cat have shown that the rhythmic patterns for locomotion are generated by spinal CPGs, while control of posture and accurate placement of feet are under control of the cerebellum and motor cortex. Decerebrate cats, for instance, can produce normal-looking gaits on a treadmill, but need to be supported to do so. The mechanisms underlying intra- and interlimb coordination, however, are still far from understood, especially in relation to gait transition.

Kimura, Akiyama, and Sakurama (1999) present a model of quadruped locomotion that emerges from the coupling of a neural controller with a quadruped robot with 12° of freedom. The neural controller is composed of four coupled oscillators, one for each limb, and several types of reflexes. Kimura and colleagues investigated several schemes of how feedback from load sensors, touch

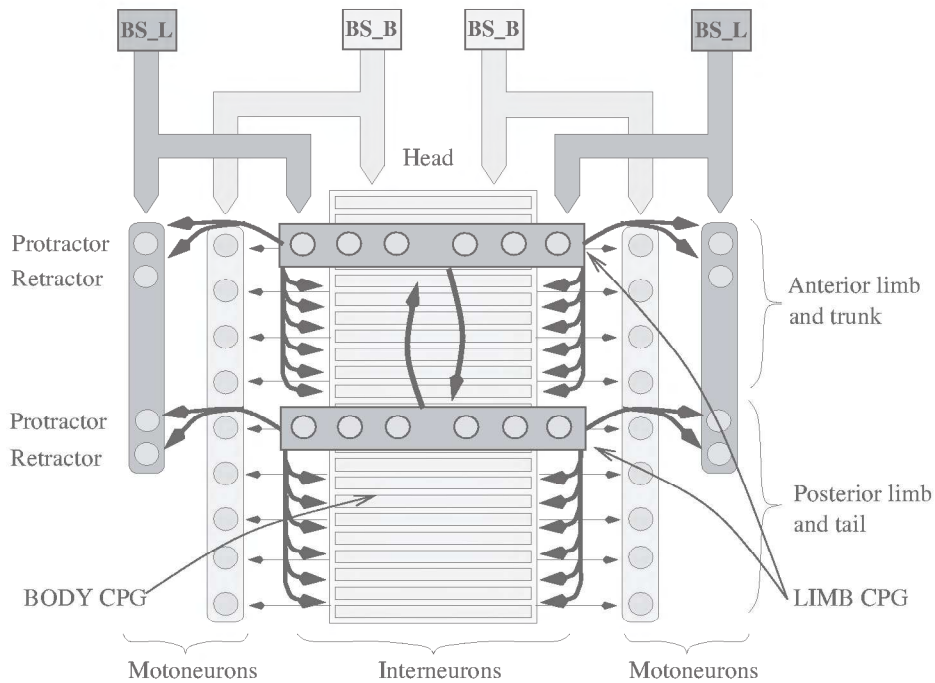


Figure 2. Potential model for the central pattern generator responsible for locomotion in the salamander. (From Ijspeert, A., 2001, A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander, *Biol. Cybern.*, 85:331–348. Reprinted with permission.)

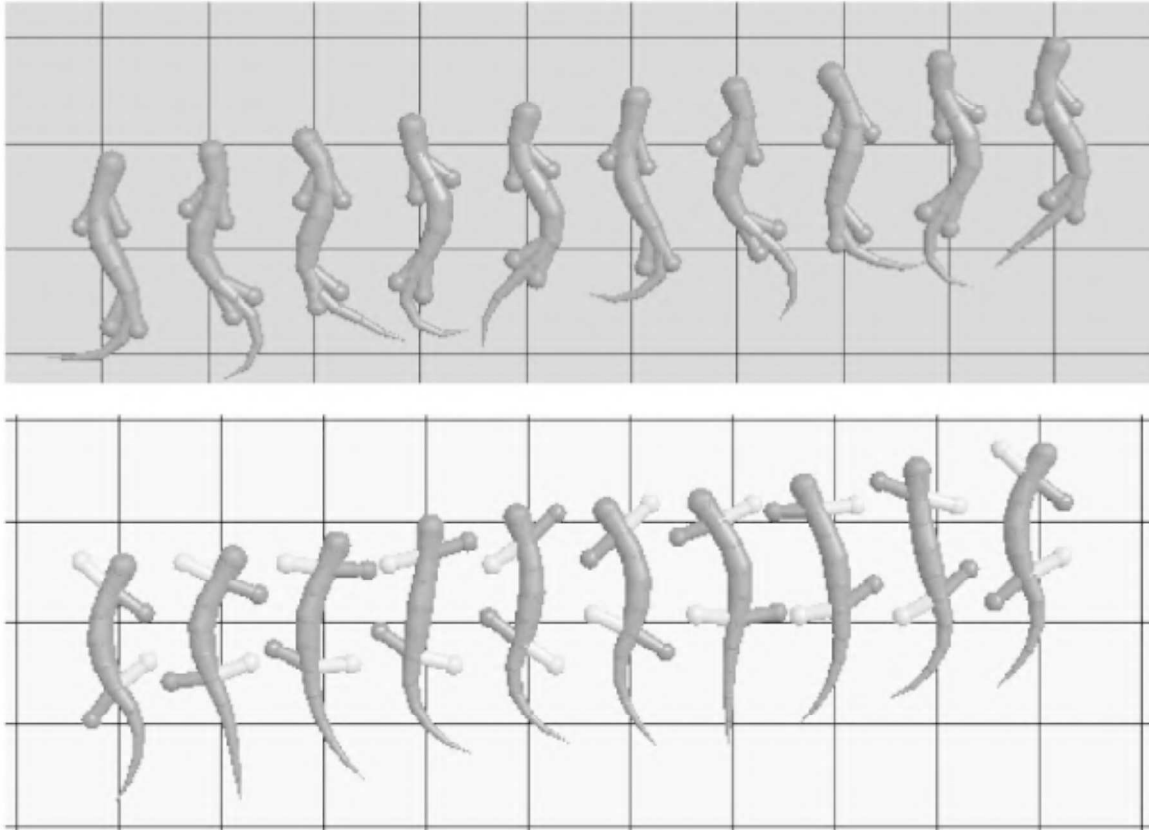


Figure 3. Neuromechanical simulation of salamander locomotion. *Top*, swimming; *bottom*, stepping. (From Ijspeert, A., 2001, A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander, *Biol. Cybern.*, 85:331–348. Reprinted with permission.)

sensors, and a vestibular system (a rate gyro) could be coupled to the CPG. The schemes in which the feedback was fed into and gated by the CPGs (as opposed to being independent of the CPGs) were found to generate significantly more stable gaits on irregular terrain. This strongly resembles the modulation of reflex signals by CPGs found in vertebrates and described earlier under Sensory Feedback. Other examples of impressive running and hopping robots can be found in Raibert and Hodgins (1993), for instance.

Biped Locomotion

Biped locomotion, such as human locomotion, is usually a dynamically stable gait. Humans use mainly two gaits: walking, in which at least one foot is in contact with the ground during the whole locomotor cycle, and running, which has a flight phase without foot contact.

The control of posture is essential in biped locomotion because of the erect posture. In humans, the motor cortex and the cerebellum play a crucial role in locomotion, much more so than in lower vertebrates. As in other vertebrates, there seems to be good evidence that the locomotor pattern can be generated at the spinal level, most likely driven from reticulospinal pathways. Clearly, the postural problem involves an important role of the cerebellum for behaviorally successful locomotion, with the corticospinal pathway playing, in addition, a role in the step-to-step modification (e.g., visually guided) of the locomotor cycle. See Horak and MacPherson in Rowell and Shepherd (1996, chap. 7) for a review.

In a series of papers, Gentaro Taga developed an interesting two-dimensional model of human locomotion (motion in the sagittal plane) in which stable locomotor patterns emerged from the interaction of a set of neural oscillators coupled to a musculoskeletal system composed of eight rigid segments (e.g., Taga, 1998). Taga's work was seminal in showing potential mechanisms of global entrainment between two highly nonlinear systems, the neural oscillators and the body. Balance in the model is maintained by a posture controller that regulates the impedance of the joints in parallel to the oscillators. The patterns are sufficiently stable to generate gaits even in unpredictable environments. In the latest version of the model, the locomotion controller is extended with a discrete movement generator for anticipatory adaptation for stepping over obstacles. The discrete movement generator modifies the stepping by generating a sequence of discrete motor signals, changing the gains of specific muscles. The functional role of the discrete movement generator is therefore comparable to the modulatory effect of the motor cortex observed during obstacle avoidance tasks in cats and humans.

Discussion

Vertebrate locomotion control is organized such that neural networks in the spinal cord generate the basic rhythmic patterns necessary for locomotion, and higher control centers interact with the spinal circuits for posture control and accurate limb movements. This means that, in general, the control signals sent to the spinal cord do not need to specify all the details of when and how much the muscles must contract, but rather specify higher-level commands such as stop and go signals, speed, and heading of motion. This type of distributed control has provided an interesting inspiration for robotics, as it implies (1) a reduction in the amount of information that has to be communicated back and forth, and (2) a reduction in the time delays between sensing, command generation, and acting.

Locomotor circuits are the result of evolution, which means that there exists a chain of changes from the ancestral vertebrate to all vertebrates. An important question that remains open is to determine which modifications have occurred in the locomotor circuits

from the generation of traveling waves for swimming (the most ancestral vertebrates were close to the lamprey) to the generation of standing waves for walking, to the generation of multiple gaits for quadruped locomotion, and finally to the generation of biped locomotion (not to forget all the other forms of vertebrate locomotion mentioned in the Introduction). This is an important issue, since the mechanisms of locomotion in modern vertebrates are strongly shaped by this evolutionary heritage and might not be fully understood without taking evolution into account. In particular, we will need to determine to what extent the three components of locomotion control—CPGs, sensory feedback, and supraspinal descending commands—have changed. It is clear that important morphological changes have significantly modified the patterns of sensory feedback. However, for lower vertebrates, it is likely that most of the changes are due to modifications of the CPGs, since CPGs are able to generate relatively normal gaits without sensory feedback, and comparative studies show that descending pathways are in general strikingly conserved (Donkelaar, 2001). In higher vertebrates such as mammals, changes of the CPGs have been accompanied by important modifications of the descending pathways under the requirements of complex posture control and accurate limb movements, although the extent of the respective changes remains unknown. In addition to neurophysiological experiments and comparative studies, computer models, in particular models that combine neural models with biomechanical models, have an important role to play in answering these fascinating questions.

Road Maps: Motor Pattern Generators; Neuroethology and Evolution

Related Reading: Evolution of Artificial Neural Networks; Spinal Cord of Lamprey: Generation of Locomotor Patterns; Visuomotor Coordination in Salamander

References

- Cheng, J., Stein, R., Jovanovic, K., Yoshida, K., Bennett, D., and Han, Y., 1998, Identification, localization, and modulation of neural networks for walking in the mudpuppy (*Necturus maculatus*) spinal cord, *J. Neurosci.*, 18:4295–4304.
- Cohen, A., 1988, Evolution of the vertebrate central pattern generator for locomotion, in *Neural Control of Rhythmic Movements in Vertebrates* (A. H. Cohen, S. Rossignol, and S. Grillner, Eds.), New York: Wiley.
- Delvolvé, I., Bem, T., and Cabelguen, J.-M., 1997, Epaxial and limb muscle activity during swimming and terrestrial stepping in the adult newt, *Pleurodeles waltl*, *J. Neurophysiol.*, 78:638–650.
- Donkelaar, H. ten, 2001, Evolution of vertebrate motor systems, in *Brain Evolution and Cognition* (G. Roth and M. Wullmann, Eds.), New York: Wiley Spectrum, pp. 77–112.
- Ekeberg, Ö., 1993, A combined neuronal and mechanical model of fish swimming, *Biol. Cybern.*, 69:363–374.
- Grillner, S., 1981, Control of locomotion in bipeds, tetrapods and fish, in *Handbook of Physiology: The Nervous System, 2, Motor Control* (V. Brooks, Ed.), Bethesda, MD: American Physiology Society, pp. 1179–1236. ♦
- Grillner, S., Degliana, T., Ekeberg, Ö., El Marina, A., Lansner, A., Orlovsky, G., and Wallén, P., 1995, Neural networks that co-ordinate locomotion and body orientation in lamprey, *Trends Neurosci.*, 18:270–279.
- Ijspeert, A., 2001, A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander, *Biol. Cybern.*, 85:331–348.
- Ijspeert, A., and Arbib, M., 2000, Visual tracking in simulated salamander locomotion, in *Proceedings of the Sixth International Conference of the Society for Adaptive Behavior (SAB2000)* (J. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S. Wilson, Eds.), Cambridge, MA: MIT Press, pp. 88–97.
- Kimura, H., Akiyama, S., and Sakurama, K., 1999, Realization of dynamic walking and running of the quadruped using neural oscillators, *Auton. Robots*, 7:247–258.
- Pearson, K., and Gordon, J., 2000, Spinal reflexes, in *Principles of Neural*

Science, 4th ed. (E. Kandel, J. Schwartz, and T. Jessel, Eds.), New York: McGraw-Hill. ♦

Raibert, M., and Hodgins, J., 1993, Legged robots, in *Biological Neural Networks in Invertebrate Neuroethology and Robotics* (R. Beer, R. Ritzmann, and T. McKenna, Eds.), San Diego, CA: Academic Press, pp. 319–354.

Rowell, L., and Shepherd, J., Eds., 1996, *Handbook of Physiology*, sect.

12: *Exercise: Regulation and Integration of Multiple Systems, Neural Control of Movement*. New York: Oxford University Press. ♦

Shik, M., Severin, F., and Orlovsky, G., 1966, Control of walking by means of electrical stimulation of the mid-brain, *Biophysics*, 11:756–765.

Taga, G., 1998, A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance, *Biol. Cybern.*, 78:9–17.

Locust Flight: Components and Mechanisms in the Motor

R. Meldrum Robertson

Introduction

The locust flight motor provides an excellent model system for investigations of constraints and mechanisms of MOTOR PATTERN GENERATION at the neuronal level. In locusts the neural elements involved in generating the patterns of flight motor activity are individually identifiable (see Comer and Robertson, 2001, for a review of identified neurons controlling insect behaviors). It is thus possible to describe the operation of networks of identified neurons, connected by identified synapses, and to determine how these networks contribute to the computational task of producing rhythmical motor patterns capable of keeping the locust aloft in an unpredictable environment.

The flight systems of other insects have attracted research interest in their neural control mechanisms. Indeed, the visuomotor control of dipteran flight has received notable attention (VISUAL COURSE CONTROL IN FLIES). Nevertheless, it is only for the locust that enough is known of the circuitry underlying the form and timing of the wingbeat that it can be useful as a model of central nervous system function.

The Motor Output

The locust flight system (Figure 1) creates a spatiotemporal pattern of electrical activity in about 80 flight motoneurons that activate muscles controlling the four wings (a pair of forewings and a pair of hindwings) and cause beating of the wings at around 22 cycles/s. Telemetric techniques now exist to monitor the activity of identified flight muscles during free flight under conditions that require the generation of different combinations of rotational and translational flight forces (Kutsch, 1999). Particular features of the motor pattern can be correlated with specific flight parameters that are modified to effect adaptive flight maneuvers (i.e., natural behaviors). It was originally demonstrated that a version of the motor pattern, albeit slower (around 12 cycles/s), could be generated by a central nervous system deafferented from phasic timing information emanating from wing proprioceptors and other sense organs. This discovery was influential in establishing the central pattern generator concept (MOTOR PATTERN GENERATION). An important question is to what extent the central pattern generator is responsible for controlling the *behavior*, particularly given that afferent input can change the set of active flight interneurons in the locust. There is no doubt that a rhythmic central pattern can be generated, but it is conceivable that this pattern is the output of a network artificially created by the act of deafferentation, i.e., a malformed, degenerate pattern that has no real bearing on the generation of the functional flight motor pattern. There is little evidence for this extreme position, and the extent to which sensory feedback supersedes the role of the central pattern generator in normal intact

flight remains unclear. Nevertheless, it is quite clear that proprioceptive feedback is necessary for appropriate timing of the wingbeat phase transitions. The tegulae are external sense organs stimulated by depression of each wing and they can initiate the subsequent elevator phase by excitation of elevator motoneurons and interneurons. The stretch receptors are internal, at the wing base, and activated by wing elevation. They promote the occurrence of the subsequent depression by opposing the hyperpolarization between the bursts of action potentials in depressor moto-

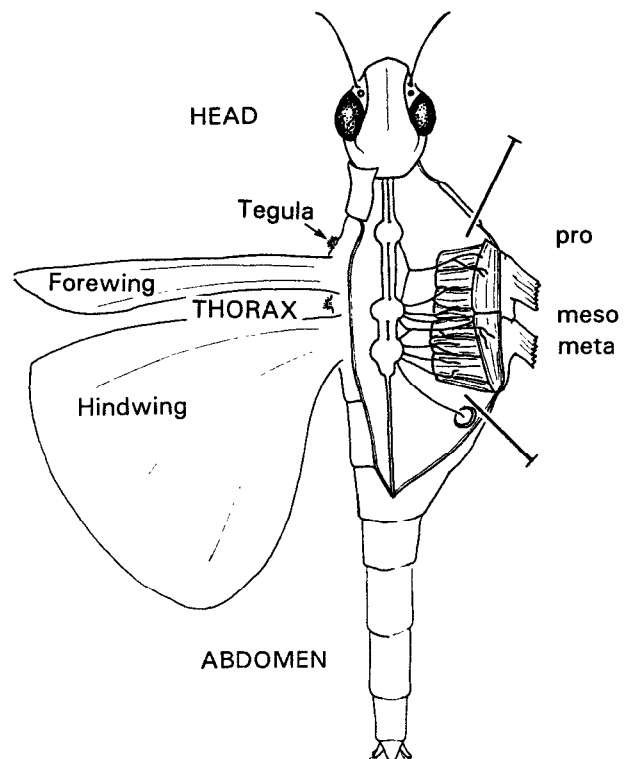


Figure 1. The locust flight system. Diagrammatic representation of a locust showing on the left side the form of the forewing and hindwing and the position of the fore and hind tegulae (only the forewing tegula is labeled). On the right side the thorax has been pinned open to reveal the bank of flight muscles that power the wings and the three thoracic ganglia (pro-, meso-, and metathoracic) that contain the motoneurons and interneurons involved in generating flight motor patterns.

neurons. A simple model describing how the stretch receptors regulate wing beat frequency has been described but would benefit from a quantitative implementation (Figure 2) (Pearson and Ramirez, 1990).

The Neuronal Components and Their Organization

The centrally generated rhythm arises as a result of the cellular properties of interneurons in the three thoracic ganglia (pro-, meso-, and metathoracic) and the interactions between these neurons. Numerous interneurons have been described. They are connected into circuits via standard, short latency, synaptic interactions probably mediated by gamma-aminobutyric acid (inhibitory) and glutamate (excitatory) (Robertson, 1989). This central circuit operates essentially as a unit distributed throughout six serially homologous, segmental neuromeres (Robertson, Pearson, and Reichert, 1982). A

simple conceptual model of the circuitry described to date has at its heart a circuit of delayed excitation and feedback inhibition that would result in an elevator-depressor burst sequence (Figure 2B) (Robertson and Pearson, 1985).

Transection and hemisection experiments have demonstrated a multiplicity of patterning elements that may aid in stabilizing the output pattern, an extremely important role for sensory elements in timing and coordination of the four wings during intact flight, and a preeminent role for the metathoracic ganglion, compared with the role of the mesothoracic ganglion, in central pattern generation. There are many oscillator mechanisms contributing to the generation of the rhythm, such that rhythm generation survives much experimental manipulation (see the section, "Modeling in the Locust Flight System," later in this article). However, there is not yet any strong evidence to support the notion that the central rhythm generator is organized as a coupled oscillator system in the sense that each wing, or pair of wings, is controlled by a separate central oscillator with the relative phasing determined by the nature of the coupling between them. This makes the locust flight system apparently unique among locomotor pattern generators, most of which do seem to be organized as coupled central oscillators.

Circuitry Underlying Steering

An important feature of any motor pattern generator for locomotion is that it must control the direction of movement through space, as well as simple translation (VISUAL COURSE CONTROL IN FLIES and SENSORIMOTOR INTERACTIONS AND CENTRAL PATTERN GENERATORS). This entails both maintaining a course in the face of environmental factors tending to displace the animal, and changing the course to enable movement toward or away from biologically relevant stimuli. Most information has accumulated for course correction behaviors—mechanisms of the "autopilot." It is only for the course correction circuitry that there is a model, at the cellular level, explaining how multimodal sensory input signaling deviation from course can be integrated into the operation of the central circuitry to cause the asymmetries in the motor output that would be necessary to compensate for an unintended change in the direction of flight (e.g., Reichert, Rowell, and Griss, 1985). The basis of the model is that continuous signals from exteroceptors, such as the ocelli (simple light detectors) or wind-sensitive head hairs, excite premotor interneurons that are rhythmically activated by the central circuits at the same time. Thus, the course deviation signal is gated through these premotor interneurons and transmitted to the motor neurons at the appropriate phase for an effective change in the motor pattern. Much of the asymmetry in the form of the wingbeat that generates steering torques occurs during the downstroke (the power stroke) of the wings, while the upstroke remains relatively symmetrical. The deviation signal can be gated so that it affects only those motoneurons involved in controlling the form of the downstroke, and it need not interact directly with the central rhythm generator.

Neuromodulation and Plasticity

The neural networks that control motor patterns are not static entities (Pearson, 2000; also see NEUROMODULATION IN INVERTEBRATE NERVOUS SYSTEMS). The mix of circulating transmitters and neuromodulators controls the particular set of circuit components and characteristics at any one time. Octopamine has a multifaceted role in the control and coordination of locust flight (Orchard, Ramirez, and Lange, 1993). From mobilization of energy resources to the modification of flight muscle properties, octopamine has influences throughout the locust enabling it to fly efficiently and, equally, to respond to the metabolic demands of flight. Indeed octopamine released from specific subgroups of DUM (dor-

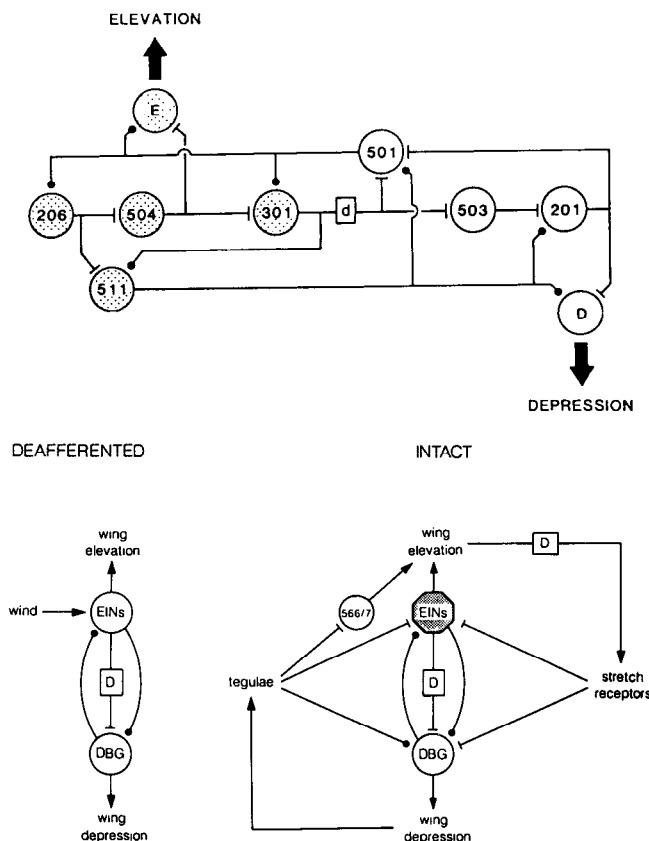


Figure 2. Circuit models of the locust flight system. **A**, Model illustrating the role of proprioceptive feedback in generating the flight motor pattern. The deafferented system (left) consists of a depressor burst generator (DBG) with reciprocal inhibitory interactions with elevator interneurons (EINs). These interneurons also pass excitation from wind input to the DBG through a delay (D) pathway. In intact animals (right), feedback from proprioceptors can recruit interneurons (e.g., 566/7) as well as interacting with the elements of the central rhythm generator. Stippling indicates that the activity pattern of the EINs is altered by the feedback. Filled circles, inhibitory connections; "T"-bars, excitatory connections. Taken from Pearson and Ramirez, 1992. **B**, Model illustrating some of the connections between flight interneurons that may contribute to generating the central flight rhythm. Elevator interneurons are stippled. Interneuron 206 receives excitation from wind input. Note the similarity with the deafferented model in **A**. The heart of the circuit is delayed excitation (301 to 501) and feedback inhibition (501 to 301). Taken from Robertson, 1986.

sal unpaired median) neurons may orchestrate the peripheral reconfigurations required by different motor programs such as walking or flight (Duch and Pflüger, 1999). Local injection of octopamine at specific sites in the thoracic ganglia is sufficient to release flight-like activity from the nervous system. Similarly, topical application of octopamine to exposed nervous systems can generate flight-like motor patterns, even in immature stages of the locust that normally do not generate such patterns. The basis for these observations is likely the fact that octopamine can induce intrinsic bursting properties (plateau potentials; OSCILLATORY AND BURSTING PROPERTIES OF NEURONS) in identified flight interneurons, and there are good reasons for supposing that this experimental manipulation reflects a physiological role for octopamine in the generation of normal flight motor patterns. It also seems likely that, in addition to the described interneuronal circuits, cellular properties and the generation of bistable plateau potentials contribute to the generation of the motor pattern in the absence of octopamine.

A short-term plasticity in the output of the flight system has been described and is ascribed to associations between muscle-specific proprioceptive input and exteroceptive input signaling deviation from course (Möhl, 1993). The interesting concept proposed by this work is that the central circuits provide a motor framework that is subsequently sculpted by the immediately preceding flight experience to provide the output that is most effective in controlling flight (e.g., maintaining a straight and level orientation). Thus, the operating circuit can be tailored to the current condition of the animal and its flight system. The specific synaptic mechanisms underlying this short-term plasticity remain to be determined. In contrast, there is some information on the synaptic mechanisms involved in a longer-term plasticity underlying functional recovery after a specific deafferentation (Wolf and Büschges, 1997). Ablation of the hindwing tegulae impairs the operation of the flight system, and this is reflected in a reduced wingbeat frequency. However, the system recovers during the subsequent two weeks due to the forewing tegulae taking over the function of the ablated hindwing tegulae. The basis for the recovery is the formation of new connections between the afferents and specific flight interneurons, accompanied by sprouting and growth of both the axonal branches of afferent fibers and the dendritic arbors of the interneurons. The synaptic connections are in a dynamic equilibrium that is disrupted by the lesion, and it is particularly interesting that different, though serially homologous, sense organs can replace those ablated.

Finally, locusts do not physiologically regulate body temperature but exist in a harsh ecological niche where ambient temperatures often exceed 45°C. Adaptive mechanisms exist to condition the circuitry by prior exposure to high temperatures and thus to extend the operating range by 5°–7°C. The current model proposes that heat stress-mediated long-term reduction of potassium conductance delays the failure of action potentials as temperature is increased by preventing potassium currents from overwhelming sodium currents at high temperatures (Wu and Robertson, 2001). Given the evolutionary conservation of cellular protective responses (e.g., the heat shock response), the mechanisms underlying thermoprotection of circuit function in this system could have implications for the development of therapeutic strategies to combat thermal failure of mammalian circuits (e.g., the hyperthermic failure of respiratory rhythm generation that has been proposed as an explanation for some cases of SIDS).

Modeling in the Locust Flight System

Insect flight lends itself to modeling at several different levels. The construction of flapping machines and flying robots is well advanced and our understanding of the aerodynamics of flapping insect flight has been greatly improved by biomechanically modeling

the wing kinematics of functionally two-winged fliers (dipteran flies and moths). Extending this approach to functionally four-winged locusts remains a challenge. Robotic and virtual models of the processing and integration of sensory information to control a search strategy (e.g., olfactory stimuli) or avoid collisions (e.g., looming visual stimuli) are well established. However, the modeling of circuit and cellular mechanisms generating flight motor rhythms is unsophisticated compared with that of many other rhythm generating systems.

Initial attempts to model the locust flight circuits used electronic Lewis “neuromimes,” which simulate the behavior of neural membranes and can be connected into networks (Wilson and Waldron, 1968). An arrangement of neuromimes into positively coupled subsets interconnected with reciprocal inhibition successfully mimicked several features of the flight motor pattern. Unfortunately, successive families of detailed models have not followed this early success. What mostly exist in the literature are conceptual circuit models of the common “ball and stick” type (e.g., Figure 2). One notable exception is the computer simulation of the central flight circuit performed with BioSim 3.0 (Grimm and Sauer, 1995). This simulation was rudimentary by current standards and introduced numerous simplifications; nevertheless, it clearly demonstrated the following: that the known circuit could produce acceptable flight-like rhythms; that subloops of the complete circuit could generate comparable rhythms; that circuit operation was relatively resistant to “synaptic strength”; and that the addition of plateau potential generating properties did not greatly affect the output of the circuit or the robustness of the rhythm generating mechanism.

Knowledge of the flight system is currently at a stage at which more detailed models would be beneficial. Could changing the parameters of a model central circuit, according to the known effects of temperature on conduction velocities and synaptic interactions, replicate the known effects of temperature on the motor patterns? Is it possible to generate a model that mimics the coordination of the four wings using a single depressor burst generator located primarily in the metathoracic ganglion? Can the effect of specific ablations and recoveries be accurately modeled? The list here, as for any other nontrivial system, is endless.

Discussion

Locust flight motor patterns are generated by an interactive mixture of the intrinsic properties of flight neurons, the operation of complex circuits, and phase-specific proprioceptive input. These mechanisms are subject to the concentrations of circulating neuromodulators and are also modulated according to the demands of a constantly changing sensory environment to produce adaptive behaviors. The system is flexible and plastic in the short term and in the long term, able to operate in spite of severe ablations and subsequently to recover from these lesions, and able to cope with extreme environmental conditions.

Without a doubt, the neural processes involved in higher brain functions will not be first described in the locust. However, the basis for these higher functions is likely due both to the generation of patterns of electrical activity in time and space and to the modulation of these patterns by the extracellular environment, by the periphery, and by experience—the control of such spatiotemporal patterning can profitably be investigated in the locust flight system.

Read Maps: Motor Pattern Generators; Neuroethology and Evolution
Related Reading: Half-Center Oscillators Underlying Rhythmic Movements; Locomotion, Invertebrate; Motor Pattern Generation; Respiratory Rhythm Generation

References

- Comer, C. M., and Robertson, R. M., 2001, Identified nerve cells and insect behavior, *Prog. Neurobiol.*, 63:409–439. ♦

- Duch, C., and Pflüger, H.-J., 1999, DUM neurons in locust flight: A model system for amine-mediated peripheral adjustments to the requirements of a central motor program, *J. Comp. Physiol.*, 184:489–499.
- Grimm, K., and Sauer, A. E., 1995, The high number of neurons contributes to the robustness of the locust flight-CPG against parameter variation, *Biol. Cybern.*, 72:329–335.
- Kutsch, W., 1999, Telemetry in insects: The “intact animal approach,” *Theory Biosci.*, 118:29–53. ♦
- Möhl, B., 1993, The role of proprioception for motor learning in locust flight, *J. Comp. Physiol.*, 172:325–332.
- Orchard, I., Ramirez, J.-M., and Lange, A. B., 1993, A multifunctional role for octopamine in locust flight, *Annu. Rev. Entomol.*, 38:227–249.
- Pearson, K. G., 2000, Neural adaptation in the generation of rhythmic behavior, *Annu. Rev. Neurosci.*, 62:723–753. ♦
- Pearson, K. G., and Ramirez, J.-M., 1990, Influence of input from the forewing stretch receptors on motoneurons in flying locusts, *J. Exp. Biol.*, 151:317–340.
- Pearson, K. G. and Ramirez, J.-M., 1992, Parallels with other invertebrate and vertebrate motor systems, in *Dynamic Biological Networks* (R. M. Harris-Warrick, E. Marder, A. I. Selverston, and M. Moulins, Eds.), Cambridge, MA: MIT Press, pp. 263–281.
- Reichert, H., Rowell, C. H. F., and Griss, C., 1985, Course correction circuitry translates feature detection into behavioural action in locusts, *Nature (Lond.)*, 315:142–144.
- Robertson, R. M., 1986, Neuronal circuits controlling flight in the locust: Central generation of the rhythm, *Trends Neurosci.*, 9:278–280.
- Robertson, R. M., 1989, Idiosyncratic computational units generating innate motor patterns: Neurons and circuits in the locust flight system, in *The Computing Neuron* (R. Durbin, R. C. Miall, and G. Mitchison, Eds.), London: Addison-Wesley, pp. 262–277. ♦
- Robertson, R. M., and Pearson, K. G., 1985, Neural circuits in the flight system of the locust, *J. Neurophysiol.*, 53:110–128.
- Robertson, R. M., Pearson, K. G., and Reichert, H., 1982, Flight interneurons in the locust and the origin of insect wings, *Science*, 217:177–179.
- Wilson, D. M., and Waldron, I., 1968, Models for the generation of the motor output pattern in flying locusts, *Proc. IEEE*, 56:1058–1064.
- Wolf, H., and Büschges, A., 1997, Plasticity of synaptic connections in sensory-motor pathways of the adult locust flight system, *J. Neurophysiol.*, 78:1276–1284.
- Wu, B. S., and Robertson, R. M., 2001, Heat shock-induced thermoprotection of action potentials in the locust flight system, *J. Neurobiol.*, 49:188–199.

Markov Random Field Models in Image Processing

Anand Rangarajan and Rama Chellappa

Introduction

Markov random field (MRF) models have become useful in several areas of image processing. The success of MRFs can be attributed to the fact that they give rise to good, flexible, stochastic image models. The goal of image modeling is to find an adequate representation of the intensity distribution of a given image. What is adequate often depends on the task at hand, and MRF image models have been versatile enough to be applied in the areas of image and texture synthesis (Zhu, Wu, and Mumford, 1997), image restoration (Geman and Geman, 1984), tomographic reconstruction (Lee, Rangarajan, and Gindi, 1995), image and texture segmentation (Krishnamachari and Chellappa, 1997), flow field segmentation (Konrad and Dubois, 1992), surface reconstruction (Geiger and Girosi, 1991), and object recognition (Gold and Rangarajan, 1996). Our aim in this article is to highlight the central ideas of this field using illustrative examples and to provide pointers to the many applications.

A guiding insight underlying most of the work on MRFs in image processing is that the information contained in the local spatiotemporal structure of images or image sequences is sufficient to obtain a good, global representation. This notion is captured by means of a local, *conditional* probability distribution. Here, the image intensity at a particular location depends only on a *neighborhood* of pixels. The conditional distribution is called an MRF. For example, a typical MRF model assumes that the image is locally smooth except for relatively few intensity gradient discontinuities corresponding to region boundaries or edges. The MRF image models are defined on the image intensities and on a further set of *hidden* attributes (edges, texture, and region labels). The observed quantities are usually noisy, blurred images, feature vectors, or projection data (in the case of emission tomography). The intensity image underlying the observations is needed in applications such as restoration and tomographic reconstruction, whereas region, boundary, and texture labels are sought in applications such as texture segmentation.

Once the local, conditional probability distribution of the MRF is specified, there are five remaining steps involved. First, the joint

distribution of the MRF is obtained. In this way, the image is represented in one global, joint probability distribution. Next, the process by which the observations are generated from the image is captured in a *degradation* probability distribution. In image restoration, for example, the degradation corresponds to a (typically uniform) blur. Then, Bayes's theorem is invoked to obtain the posterior probability distribution of the image given the observations. The posterior distribution gives us the probability that an image (with smooth regions and sharp region boundaries, for example) could have been degraded to obtain the particular observed noisy, blurred image. Once the posterior probability distribution is obtained, we can associate a cost with each configuration in the posterior. For example, if only the true underlying image will do, the cost penalizes all other images equally. The cost is formulated, keeping in mind the task at hand. A measure of the cost is minimized with respect to the image intensities (in image recovery tasks) or image attributes (in labeling tasks). Finally, since MRFs are specified with model parameters, these are estimated from a training set (if one exists) or adaptively, along with the cost minimization phase alluded to earlier. The overall MRF framework fits well within a Bayesian estimation/inference paradigm. In the next section, we step through all five phases of MRF modeling.

A Framework for Estimation and Inference

MRF image models represent knowledge in terms of local probability distributions. Specifically, the kinds of probability distributions generated by MRFs have a local neighborhood structure. Neighborhood systems commonly used by MRFs are depicted in Figure 1A.

Let us associate an image with a random process X whose element is X_s , where $s \in S$ refers to a site in the image. The local conditional distribution can be written as follows:

$$\Pr(X_s = x_s | X_t = x_t, t \neq s, t \in S) = \Pr(X_s = x_s | X_t = x_t, t \in G_s) \quad (1)$$

where X and x denote the random field and a particular realization, respectively, and G_s is the local neighborhood at site s . Note that

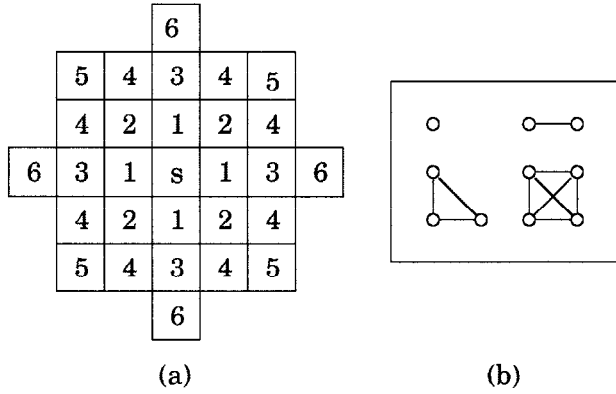


Figure 1. A, Neighborhood systems for MRFs. B, Cliques in MRFs.

in general, G_s can be large or small, but it is usually a local neighborhood, in keeping with the spirit of MRF modeling.

Let s be the site (i, j) and let the local neighborhood be a first-order neighborhood ($G(s)$ is the collection $(i, j + 1)$, $(i, j - 1)$, $(i + 1, j)$, $(i - 1, j)$). Then, let the conditional density take the form

$$p(X_s = x_s | X_t, t \in G_s) = \frac{1}{\sqrt{2\pi}} \times \exp \left[-\frac{1}{2} \left(x_{ij} - \frac{1}{4} [x_{i,j+1} + x_{i,j-1} + x_{i+1,j} + x_{i-1,j}] \right)^2 \right] \quad (2)$$

This is a very simple special case of the first-order Gauss-Markov model (Besag, 1974). The Gauss-Markov model has been widely used in image processing tasks (Dubes and Jain, 1989).

The MRF model consists of a set of *cliques*. A clique is a collection of sites such that any two sites are neighbors. Different orders of cliques are shown in Figure 1B. The order of a clique refers to the number of distinct sites that appear multiplicatively. We now calculate the clique energies involving the site x_{ij} by expanding the conditional probability density and collecting the terms. There are cliques of order one and two. They are

$$\frac{x_{ij}^2}{2}, \quad -\frac{x_{ij}x_{i,j+1}}{4}, \quad \text{and} \quad -\frac{x_{ij}x_{i+1,j}}{4} \quad (3)$$

The first term in Equation 3 is of order one and the latter two terms are of order two.

MRF-Gibbs equivalence

We now ask the following question: Given the conditional probability structure $\Pr(X_s = x_s | X_t = x_t, t \in G_s)$, what is the joint probability distribution $\Pr(X = x)$? This is of utmost importance, since it is the joint probability distribution and not the conditional distribution that contains the complete image representation.

Before relating the conditional and joint distributions, we introduce the concept of a Gibbs distribution, which will turn out to be crucial in specifying the relationship. A Gibbs distribution is specified by an *energy function* $E(x)$ and can be written as

$$\Pr(X = x) = \frac{1}{Z} \exp(-E(x)) \quad (4)$$

where the *partition function*

$$Z = \sum_x \exp(-E(x)) \quad (5)$$

is a normalizing constant and involves a summation over all possible configurations of X . Energy functions have been widely used

in spin-glass models of statistical physics. The minimum energy configuration corresponds to an ordered system of spins. $E(x)$ cannot take infinite values.

Our detour into Gibbs distributions is justified for the following reason. The Hammersley-Clifford theorem (Besag, 1974; Geman and Geman, 1984) states that any conditional distribution has a joint distribution, which is Gibbs (Dubes and Jain, 1989) if the following conditions hold.

Positivity: $\Pr(X = x) > 0$.

Locality: $\Pr(X_s = x_s | X_t = x_t, t \neq s, t \in S) = \Pr(X_s = x_s | X_t = x_t, t \in G_s)$.

Homogeneity: $\Pr(X_s = x_s | X_t = x_t, t \in G_s)$ is the same for all sites s .

The locality condition is the same as the Markov property described by Equation 1. The Hammersley-Clifford theorem allows us to shuttle between the conditional probability structure in Equation 1 and the joint probability in Equation 4.

The recipe for obtaining the joint density function is as follows: (1) assemble the different clique energies from the conditional probability, and (2) compute the energy function by adding up the clique energies.

We calculate the energy function for the simple first-order Gauss-Markov model:

$$\begin{aligned} E(x) &= \frac{1}{2} \left(\sum_{ij} \left[x_{ij}^2 - \frac{x_{ij}x_{i,j+1}}{2} - \frac{x_{ij}x_{i+1,j}}{2} \right] \right) \\ &= \frac{1}{8} \sum_{ij} [(x_{ij} - x_{i,j+1})^2 + (x_{ij} - x_{i+1,j})^2] \end{aligned} \quad (6)$$

It can be seen from the energy function $E(x)$ and the conditional density that the essence of the Hammersley-Clifford theorem lies in the clique energies. We examined the conditional density and teased apart the different orders of cliques (first and second order) and the associated clique energies. Then, all clique energies were summed (taking care to count each clique only once), yielding the energy function $E(x)$. Our presentation has been quite terse, and further details on cliques and the transition from the conditional to the joint probability distribution can be found in Besag (1974), Geman and Geman (1984), and Dubes and Jain (1989).

The Prior and Degradation Models

Naturally, we are not content with merely obtaining MRF-Gibbs image models. These models can be used in a variety of image processing and analysis tasks. As mentioned previously, MRF modeling fits perfectly into a Bayesian estimation/inference paradigm. A Bayesian setup consists of two ingredients—the prior and the degradation model. The prior model is defined on the set of image attributes X that are of interest. In edge-preserving image restoration (Geman and Geman, 1984), for example, X includes the set of image intensities and a further set of binary-valued edge labels. In texture segmentation (Lakshmanan and Derin, 1989), X includes the image intensities and a set of texture labels at each location. The degradation model is a model of the physical process by which the observations are generated. Usually, we are faced with noisy and incomplete observations. Denote the set of observations by Y , and let the degradation model also be a Gibbs-Markov distribution:

$$\Pr(Y = y | X = x) = \frac{1}{Z_D(x)} \exp(-E_D(x, y)) \quad (7)$$

where

$$Z_D(x) = \sum_y \exp(-E_D(x, y)) \quad (8)$$

In general, the partition function $Z_D(x)$ is a function of the image attributes x . $E_D(x, y)$ is the energy function corresponding to the degradation model. For example,

$$E_D(x, y) = \frac{1}{2} \sum_s \left(y_s - \sum_t \mathcal{H}_{st} x_t \right)^2$$

yields a Gaussian degradation model wherein Y is obtained by blurring X with a *blur function* \mathcal{H} and adding additive Gaussian noise at each site s . This type of degradation model routinely occurs in image restoration (Geman and Geman, 1984) and (with some modifications) in tomographic reconstruction (Lee et al., 1995).

A Bayesian Posterior Energy Function

Given the degradation and prior models, Bayesian estimation/inference proceeds as follows. The posterior distribution $\Pr(X = x|Y = y)$ is obtained by using Bayes's theorem:

$$\Pr(X = x|Y = y) = \frac{\Pr(Y = y|X = x)\Pr(X = x)}{\Pr(Y = y)} \quad (9)$$

Once the posterior distribution is obtained, an estimate (\hat{X}) of X is found by minimizing the expected cost, which is a measure of the distance between the true and estimated values:

$$C = \sum_x C(x, x^*)\Pr(X = x|Y = y) \quad (10)$$

where x^* is the true value. When the familiar squared-error cost is used, the estimator (MMSE) turns out to be the conditional mean $\mathcal{E}(X|Y = y)$ (\mathcal{E} denotes the expectation operator). If the cost *equally* penalizes all x different from x^* ($C(x, x^*) = \delta_{x,x^*}$), the maximum a posteriori (MAP) estimator results.

When the degradation and prior models are Gibbs, the posterior is Gibbs as well. To see this, assume a prior energy function $E_P(x)$ giving $\Pr(X = x) = (1/Z_P) \exp(-E_P(x))$. The posterior distribution (using Equation 9) is

$$\Pr(X = x|Y = y) = \frac{\exp(-E_D(x, y) - \log(Z_D(x)) - E_P(x))}{\sum_x \exp(-E_D(x, y) - \log(Z_D(x)) - E_P(x))} \quad (11)$$

The posterior energy function $E(x) = E_D(x, y) + \log(Z_D(x)) + E_P(x)$. In the case of the MAP estimate, the entire Bayesian estimation engine reduces to minimizing just this posterior energy function $E(x)$, since the partition function of the posterior is independent of x . However, when the MMSE estimate is desired, the expected value of X in the posterior distribution needs to be computed. This computation is usually intractable, since it involves computing the partition function of the posterior distribution.

MAP Estimation

Restricting our focus to MAP estimation, we observe that MAP estimation reduces to minimizing the posterior energy function $E(x)$. This minimization involves the different kinds of processes that make up X . For example, in edge-preserving image restoration (Geman and Geman, 1984), the process X includes both continuous-valued image intensities and binary-valued edge variables. Consequently, the minimization of the posterior objective function is a difficult problem, owing to the presence of nontrivial local minima. A general technique for finding global minima is simulated annealing (Geman and Geman, 1984; Lakshmanan and Derin, 1989) or, more recently, Markov chain Monte Carlo (MCMC) (Zhu et al., 1997), but these methods are usually computationally very intensive. A lot of effort has been expended in obtaining good suboptimal solutions to the MAP estimation problem (Yuille and

Kosowsky, 1994; Lee et al., 1995; Gold and Rangarajan 1996). Deterministic annealing (DA) is a general method that has emerged. Deterministic annealing methods begin with a modified posterior:

$$\Pr(X = x|Y = y) = \frac{1}{Z(\beta)} \exp(-\beta E(x)) \quad (12)$$

where $\beta > 0$ is the inverse temperature. Note that the partition function is now a function of the inverse temperature. The terminology is inherited from statistical physics. The idea of cooling a system slowly to reach a minimum energy configuration has a computational parallel in MRFs. The basic idea is to embed the posterior in a β exponentiated manner and to track the maximum of this posterior through a gradual increase in β . In this manner, the posterior energy function is increasingly closely approximated by a sequence of smooth, continuous energy functions.

The main reason for doing this is based on the following statistical mechanics identity:

$$F(\beta) \stackrel{\text{def}}{=} -\frac{1}{\beta} \log Z(\beta) = \mathcal{E}(E(x)) - \frac{1}{\beta} S(\beta) \quad (13)$$

where S is the entropy (defined as $-\sum_x \Pr(X = x|Y = y) \log(\Pr(X = x|Y = y))$). The entropy is proportional to the logarithm of the total number of configurations, and as the temperature is reduced (and fewer configurations become likely), it gradually goes to zero. Also, the expected value of the posterior energy goes to the minimum value of the energy. The key idea in deterministic annealing is to minimize the *free energy* F instead of $E(x)$ while reducing the temperature to zero. The free energy (at low β) is a smooth approximation to the original, nonconvex energy function and approaches $E(x)$ as β tends to infinity. However, the free energy involves the logarithm of the partition function, which is intractable! An approximation to the free energy (usually called the naive mean field approximation) is minimized instead. Although the details are beyond the scope of this article (see Geiger and Girosi, 1991; Yuille and Kosowsky, 1994; Lee et al., 1995; and Gold and Rangarajan, 1996), we present an example illustrating the method. Let the energy function contain only binary-valued variables and take the following form:

$$E(x) = \sum_{ij} T_{ij} x_i x_j + \sum_i h_i x_i, \quad x_i \in \{0, 1\} \quad (14)$$

The free energy F is given by

$$F(v) = \sum_{ij} T_{ij} v_i v_j + \sum_i h_i v_i + \frac{1}{\beta} \sum_i [v_i \log(v_i) + (1 - v_i) \log(1 - v_i)] \quad (15)$$

where $v_i \in [0, 1]$. The free energy consists of two terms. The first term can be seen as an approximation to the expected value of the energy once the identification $v_i \approx \mathcal{E}(x_i)$ is made. Now,

$$\mathcal{E}(E(x)) = \sum_{ij} T_{ij} \mathcal{E}(x_i x_j) + \sum_i h_i \mathcal{E}(x_i) \quad (16)$$

When the expected value of the product $x_i x_j$ is replaced by the product of the expected values ($v_i v_j$), the naive mean field approximation results. The third term in Equation 15 is an approximation to the entropy. At each setting of β , Equation 15 is minimized with respect to v , after which β is increased. In this manner, a deterministic network is obtained. There are questions regarding the choice of annealing schedules and the quality of the minima obtained, and for the most part, except for very specific posterior energy functions, there is a dearth of analytical results in this area. However, the method is quite general and has been applied with varying degrees of success to a variety of image processing and analysis tasks, such as tomographic reconstruction (Lee et al., 1995), flow

field segmentation (Konrad and Dubois, 1992), surface reconstruction (Geiger and Giroi, 1991), and object recognition (Gold and Rangarajan, 1996).

Parameter Estimation

So far we have concentrated on estimating X given the noisy observations Y . We have emphasized that Gibbs-Markov models are specified by local clique energies (from which the global distribution can be obtained). Consider a prior distribution

$$\Pr(X = x|\theta) = \frac{1}{Z(\theta)} \exp \left(-\frac{1}{2} \sum_k \sum_{\langle s,t \rangle_k} \theta_k (x_s - x_t)^2 \right) \quad (17)$$

where θ_k is a parameter associated with clique $\langle s, t \rangle$. Since pairwise interactions are used, a clique between pixels s and t is denoted by $\langle s, t \rangle$. This is the general form of the Gauss-Markov model. The model is a generalization of our earlier model (Equation 6) since it has the same clique form, although with a more general neighborhood structure. The partition function involves a sum over the configurations of X and is a function of θ . Other than the estimation/inference problem, we are also saddled with the problem of parameter estimation.

The parameters can be estimated by maximizing the joint probability of X with respect to the unknown parameters (Lakshmanan and Derin, 1989). In most cases, this computation is intractable in its pure form, and approximations have to be devised. The typically available approximations are pseudo-likelihood, mean-field, and MCMC. The computational requirements of the different methods range from low for pseudo-likelihood to moderate for mean-field to high for MCMC. The availability of a suitable training set is critical to both likelihood and pseudo-likelihood parameter estimation. When a training set is not available, parameter estimation and cost minimization proceed in lockstep, resulting in the so-called joint MAP procedure wherein the parameters θ and the states X are bootstrapped (Lakshmanan and Derin, 1989). From a theoretical standpoint, there are important issues of consistency and efficiency of the parameter estimates; for details, see Kashyap and Chellappa (1983).

Discussion

The MRF framework is well suited to a wide variety of image processing and analysis tasks. Our exposition has been brief, and we have ignored important issues such as validation, choice of the order of MRF models, and sizes of training sets. Validation, for example, takes us into the bias/variance dilemma (Geman, Bienenstock, and Doursat, 1992). MRF models, being parametric, introduce a certain kind of bias into the image representation. This seems to be the right kind of bias (in terms of reducing variance) for tasks like image restoration, tomographic reconstruction, and texture segmentation. However, if the order of the chosen model is incorrect, high bias could result. It is in bias/variance terms that MRF image models should be compared alongside "mechanical" (as opposed to probabilistic) models such as splines, generic representations like radial basis functions (RBFs), and tabula rasa, feedforward neural networks. Also, there are interesting similarities between Gauss-Markov models and thin-plate splines (Lee et al.,

1995; Wahba, 1990); see GENERALIZATION AND REGULARIZATION IN NONLINEAR LEARNING SYSTEMS). For example, the simple case of the first-order Gauss-Markov model with the parameters $\theta_1 = \theta_2 = (1/4)$ is identical to the discrete membrane (first-order thin-plate spline in two dimensions). Correspondences of this sort should be expected, since MRF models, splines, and RBFs impose local smoothness constraints, although in different ways. In recent years there has been increased interest in scale-space and multiresolution image processing and analysis methods. Although there are deep unresolved issues in integrating scale into Markov models, this situation has not deterred researchers from using multiresolution MRFs in specific applications (Lakshmanan and Derin, 1989; Krishnamachari and Chellappa, 1997). Finally, and very recently, interesting interrelationships have been discovered between Bayesian MAP estimation on Gibbs posterior energy functions and Bayesian belief propagation algorithms (Weiss, 2000; see GRAPHICAL MODELS: PROBABILISTIC INFERENCE).

Road Map: Vision

Related Reading: Hidden Markov Models; Probabilistic Regularization Methods for Low-Level Vision

References

- Besag, J., 1974, Spatial interaction and the statistical analysis of lattice systems, *J. R. Statist. Soc. B*, 36:192–236. ♦
- Dubés, R. C., and Jain, A. K., 1989, Random field models in image analysis, *J. Appl. Statist.*, 16:131–164. ♦
- Geiger, D., and Giroi, F., 1991, Parallel and deterministic algorithms from MRFs: Surface reconstruction, *IEEE Trans. Pattern Anal. Machine Intell.*, 13:401–412.
- Geman, S., Bienenstock, E., and Doursat, R., 1992, Neural networks and the bias/variance dilemma, *Neural Computat.*, 4:1–58.
- Geman, S., and Geman, D., 1984, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741.
- Gold, S., and Rangarajan, A., 1996, A graduated assignment algorithm for graph matching, *IEEE Trans. Pattern Anal. Machine Intell.*, 18:377–388.
- Kashyap, R. L., and Chellappa, R., 1983, Estimation and choice of neighbors in spatial interaction models of images, *IEEE Trans. Inform. Theory*, 29:60–72.
- Konrad, J., and Dubois, E., 1992, Bayesian estimation of motion vector fields, *IEEE Trans. Pattern Anal. Machine Intell.*, 9:910–926.
- Krishnamachari, S., and Chellappa, R., 1997, Multiresolution Gauss-Markov random field models for texture segmentation, *IEEE Trans. Image Proc.*, 6:251–267. ♦
- Lakshmanan, S., and Derin, H., 1989, Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing, *IEEE Trans. Pattern Anal. Machine Intell.*, 8:786–799.
- Lee, S. J., Rangarajan, A., and Gindi, G., 1995, Bayesian image reconstruction in SPECT using higher order mechanical models as priors, *IEEE Trans. Med. Imaging*, 14:669–680.
- Wahba, G., 1990, *Spline Models for Observational Data*, Series in Applied Mathematics, vol. 59, Philadelphia, PA: SIAM.
- Weiss, Y., 2000, Correctness of local probability propagation in graphical models with loops, *Neural Computat.*, 12:1–41.
- Yuille, A. L., and Kosowsky, J., 1994, Statistical physics algorithms that converge, *Neural Computat.*, 6:341–356. ♦
- Zhu, S. C., Wu, Y. N., and Mumford, D., 1997, Minimax entropy principle and its application to texture modeling, *Neural Computat.*, 9:1627–1660.

Memory-Based Reasoning

David L. Waltz

Introduction

Memory-based reasoning (MBR) refers to a family of nearest-neighbor-like methods (Dasarthy, 1991) for making decisions or classifications. MBR differs from other nearest-neighbor methods primarily in its metrics for computing the distance between examples, and in MBR's suitability for use with symbolic-valued features. Nearest-neighbor methods generally use a simple overlap distance metric, which defines distance as the number of mismatched features between two instances. MBR uses metrics related to the value distance metric (VDM), introduced in Stanfill and Waltz (1986). Considerable effort on recoding training cases is generally required in order to use neural nets on symbolic problems. In general, symbolic items need to be mapped to numerical values, feature vectors, or some related form. MBR is specifically designed to handle such cases directly; MBR uses the statistical similarity of outcomes to automatically generate similarity metrics for symbolic inputs, without the need for recoding. MBR has been used for software recommendation agents such as the Firefly (Maes and Kozierok, 1993), classification of news articles (Linoff, Masand, and Waltz, 1992) and U.S. Census Bureau long forms (Creecy et al., 1992), computational biology (Zhang, Mesirov, and Waltz, 1992; Yi and Lander, 1993; Cost and Salzberg, 1993), and a variety of other tasks. See Aha (1997) for a survey of MBR and related systems and applications, and discussions of trade-offs in case-based system design.

Comparisons of MBR and Neural Nets

Several projects have compared MBR with backpropagation neural nets. In terms of decision accuracy, MBR has often outperformed neural nets, and in some cases all the other learning methods with which it has been compared (Zhang et al., 1992; Cost and Salzberg, 1993; Rachlin et al., 1994); results for MBR are generally comparable with the best other learning methods. MBR has also been applied to cases that are beyond the representational reach of current neural net methods, such as the classification of free-text examples of arbitrary lengths (Creecy et al., 1992). Like neural nets, MBR systems can be built easily and quickly, with very little programming required. Unlike neural nets, no learning phase is necessary. Updating is therefore simple, requiring only additions, deletions, and modifications to MBR's database of examples, and decisions are always based on all known data, even the most recent. Also, unlike neural nets, MBR does not generally require elaborate re-representation schemes. MBR provides "justifications" for decisions, namely the nearest example(s) from the database, whereas considerable analysis effort is generally required to understand why a neural net system behaves as it does. Thus, debugging and tuning tend to be easier with MBR. And finally, MBR methods can provide confidence levels.

MBR does have a major disadvantage: although it does not require a training phase, MBR's decision phase is computationally expensive, and MBR systems have high memory requirements. In general, the entire database of examples is kept, whereas with neural nets, the total storage requirements for a trained net are generally much smaller than the training set used to create it. Because of this, early MBR systems were implemented on massively parallel computers or special-purpose hardware.

"Eager" Versus "Lazy" Systems

Lazy systems (such as MBR) defer classification decisions until a case is presented, while eager systems (like backpropagation and

other learning systems) do most of their classification work ahead of time in a training phase (Aha, 1997). Systems intermediate between eager and lazy often have attractive properties (Kasif et al., 1998).

How Does MBR Work?

Every MBR system requires a similarity metric for judging the distance between an item to be classified and all items in the example database. If the MBR system uses k -nearest neighbors, then a scheme for combining the information from k examples is also needed. We can illustrate the idea with a simple metric, the modified value difference metric (MVDM) (Cost and Salzberg, 1993), which is similar to the ones that have been most commonly used in MBR applications. Assume the database is relational, with n -tuples consisting of $n - 1$ predictor fields (p_1, p_2, \dots, p_{n-1}) plus a goal field G . (This is not the fully general case, but it is the simplest and most common one, and serves well for purposes of illustration.) Each case in the database is a vector of predictor values (a_1, a_2, \dots, a_{n-1}), plus a goal G that can take on any of a finite set of values: G is an element of (g_1, g_2, \dots, g_m) . Then the distance between a novel situation $B = (b_1, b_2, \dots, b_{n-1})$ and a case A from the database $A = (a_1, a_2, \dots, a_{n-1})$ is:

$$\sum_j \left[\sum_i |\text{Prob}(g_j|b_i) - \text{Prob}(g_j|a_i)| \right]$$

where j ranges over all possible values for the goal field and i ranges over all predictor fields. To keep computation tractable, the set of goal fields may be limited, for example by only indexing over goals that correspond to database instances with non-zero overlap metrics (which are much cheaper to compute).

The MVDM metric basically compares the distribution of cases for each pair of predictor field values of A and B . The distance between A and B is small if these two-predictor field values are associated with similar distributions of goal field values.

The result of applying this metric is a rank-ordered list of cases from the database, with distance values for each. The classification proposed for the novel situation is then the goal field value of the case with the minimum distance (single neighbor case), or the goal field value whose weighted sum of distances over the k closest cases is the smallest (k -nearest-neighbor case).

NETtalk Task

MBRtalk, the first MBR system (Stanfill and Waltz, 1986), used as its main example the NETtalk database, and compared its performance with NETtalk (Sejnowski and Rosenberg, 1987). The NETtalk task is to produce pronunciations for all English words, based on a small (700-word) training set. Sejnowski and Rosenberg's NETtalk, a backpropagation system (with some special output processing), achieved a 78% letter-by-letter generalization performance. MBRtalk produced a 78% letter-by-letter generalization performance on the original NETtalk database, the same as that reported by Sejnowski and Rosenberg. MBRtalk demonstrated a 93% correct generalization using a 16,000-word corpus, a task that has not been attempted with a neural net system and one that would probably require very long training times. MBR does not require training, although it is possible to do some precalculation in order make the MBR decision phase run faster.

Protein Structure Prediction

For several years, the best protein secondary structure prediction systems were based on neural nets (Qian and Sejnowski, 1988). After considerable experimentation and tuning, Zhang et al. (1992) showed that MBR outperforms backpropagation on this task, albeit by a small margin—64.5% correct for MBR, versus 63.5% for a three-layer backpropagation system and 64.0% for a cascaded backpropagation system (Qian and Sejnowski, 1988), all using eight-way cross-validation on the same 19,861-residue database. Interestingly, MBR and neural nets agreed with each other only about 80% of the time, and both methods agreed with a statistical system with 63.5% performance about as often. A hybrid system was constructed that used a backpropagation net to combine the outputs of the three methods, yielding a performance of 66.4%, an improvement that is better than any of the others alone, with high statistical significance (Zhang et al., 1992). Current methods, often combining recurrent neural nets with MBR-like methods, perform significantly better, in the range of 76% (Baldi et al., 1999), although a considerable portion of this improvement may be the result of more complete and accurate protein databases.

Tests on UCI Repository of Machine Learning Databases

Several papers have reported results on data sets from the University of California–Irvine (UCI) repository that allow us to compare MBR, specifically the widely used PEBLS system (Cost and Salzberg, 1993), with backpropagation neural nets.

As reported in Kasif et al. (1998), PEBLS outperformed or equaled both a Bayes classifier and a Hamming distance nearest-neighbor system on six of eight tasks, and was within 1.1% on the other two tasks. This paper also showed that MBR always outperforms a Bayes classifier in domains with nonplanar decision boundaries. Neural nets outperformed MBR on small databases (soybean disease with 289 examples, iris database with 150 examples).

Discussion

MBR as a Neural Model?

It has been suggested that the cerebellum stores many examples of motor movements that can then be interpolated to provide smooth motor movements (Atkeson, Moore, and Schaal, 1997). MBR, as a variant of case-based reasoning (CBR), has been proposed as an associative memory implementation.

Why Do These Methods Perform Differently?

Generally stated, MBR and neural nets form decision surfaces differently, and so will perform differently. MBR can become arbitrarily accurate if large numbers of cases are available, and if these cases are well-behaved and properly categorized. Neural nets cannot respond well to isolated cases but tend to be good at smooth extrapolation. To give an example, consider training a NETtalk

system to pronounce the letter *p*, and assume that all *ps* are either pronounced *P* (as in *pig*) or *F* (as in *photo*) except for one example, where *p* is silent (*psychology*). Given that there are many *P* and *F* examples, these will statistically dominate the hidden units for a backpropagation net, and it is very unlikely that words beginning with *ps* will be correctly pronounced. But MBR is able to pronounce a *ps* word correctly, even with only a single near example.

For other examples, different MBR metrics can be used for systems for tasks that cannot be currently handled by neural nets. In Creecy et al. (1992), for example, a text similarity metric (“vector similarity”—basically normalized weighted word overlap) allowed an MBR system to assign keywords to Census Bureau data with free-text fields. Such a task is beyond current neural net training methods.

Road Map: Artificial Intelligence

Related Reading: Data Clustering and Learning; Pattern Recognition

References

- Aha, D., Ed., 1997, *Lazy Learning* (special issue), *Artif. Intell. Rev.*, 11:7–423.
- Atkeson, C., Moore, A., and Schaal, S., 1997, Locally weighted learning for control, *Artif. Intell. Rev.*, 11:75–113.
- Baldi, P., Brunak, S., Frasconi, P., Pollastri, G., and Soda, G., 1999, Exploiting the past and the future in protein secondary structure prediction, *Bioinformatics*, 15:937–946.
- Cost, S., and Salzberg, S., 1993, A weighted nearest neighbor algorithm for learning with symbolic features, *Machine Learn.*, 10(1):57–78. ♦
- Creecy, R., Masand, B., Smith, S., and Waltz, D., 1992, Trading MIPS and memory for knowledge engineering, *Commun. ACM*, 35(8):48–64.
- Dasarthy, B., 1991, *Nearest Neighbor (NN) Norms*, Washington, DC: IEEE Computer Society Press.
- Kasif, S., Salzberg, S., Waltz, D., Rachlin, J., and Aha, D., 1998, A probabilistic framework for memory-based reasoning, *Artif. Intell.*, 104:287–311.
- Linoff, G., Masand, B., and Waltz, D., 1992, Classifying news stories using memory based reasoning, in *Proceedings of the SIGIR Conference*, Copenhagen, pp. 59–65.
- Maes, P., and Kozierok, R., 1993, Learning interface agents, in *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Cambridge, MA: MIT Press, pp. 459–465.
- Qian, N., and Sejnowski, T., 1988, Predicting the secondary structure of globular proteins using neural network models, *J. Mol. Biol.*, 202:865–884.
- Rachlin, J., Kasif, S., Salzberg, S., and Aha, D., 1994, Towards a better understanding of memory-based and Bayesian classifiers, in *Proceedings of the Eleventh International Conference on Machine Learning*, New Brunswick, NJ, pp. 242–250.
- Sejnowski, T., and Rosenberg, C., 1987, Parallel networks that learn to pronounce English text, *Complex Systems*, 1:145–168.
- Stanfill, C., and Waltz, D., 1986, Toward memory-based reasoning, *Commun. ACM*, 29(12):1213–1228. ♦
- Yi, T.-M., and Lander, E., 1993, Protein secondary structure prediction using nearest neighbor methods, *J. Mol. Biol.*, 232:1117–1129.
- Zhang, X., Mesirov, J., and Waltz, D., 1992, A hybrid method for protein secondary structure prediction, *J. Mol. Biol.*, 225:1049–1063.

Minimum Description Length Analysis

Richard S. Zemel

Introduction

In this article, we review a variety of ways in which ideas relating to minimum description length (MDL) have been applied to neural networks. We begin with a brief introduction to the historical roots

of MDL, and then describe the direct relationship between MDL and Bayesian model selection methods. We divide the applications of MDL to neural networks into two categories corresponding to the two main classes of learning in networks: supervised and unsupervised.

Historical Background

The underlying approach in MDL—applying coding theory to determine simplicity—grew out of work from the mid-1960s, when Kolmogorov, Solomonoff, and Chaitin introduced theories concerning the information content of an object. Instead of relating information to probabilities, as Shannon had (e.g., Shannon, 1948), they adopted a computational approach. They defined the information in a binary string to be the length of the shortest program with which a general-purpose computer can generate the string. The resulting algorithmic theory of information has significant implications in many areas, but it has not had much impact on the practical construction of programs, because the proposed form of information is not computable in its pure form. However, a number of learning techniques (including MDL) have been derived by making approximations to this information measure. The interested reader should consult Li and Vitanyi (1993) for a detailed mathematical review of the history of MDL and its relationship to many other current inductive inferencing techniques.

Defining the Minimum Description Length Principle

MDL can be seen as a principled version of Occam's razor, where the goal is to find the simplest accurate description of a set of data. An informal definition of the MDL principle (Rissanen, 1989) is that the best model to explain a set of data is the one that minimizes the summed length, in bits, of (1) the description of the model and (2) the description of the data, when encoded with respect to the model.

In algorithmic information theory, the model is a general computational model, i.e., a Turing machine. The MDL approach makes the problem tractable by considering particular classes of models. For example, if the goal is to infer decision trees, then the model is a decision tree, while if the goal is to learn the weights of a neural network, the model may be a network with a particular architecture. In this article, a model refers to various aspects of a network, such as its weights and activities.

It is useful to formulate MDL based on a communication protocol in which these terms are unified into a single encoded message that must be decoded in order to reproduce the data. The sender transmits a message, encoded in the description language \mathcal{L} , that conveys both the model M and the data D with respect to the model; this second term can be seen as the residuals, i.e., aspects of the data not predicted by the model. The standard goal of inferring an optimal M from the data is then equivalent to minimizing the length of this encoded message:

$$|\mathcal{L}(M, D)| = |\mathcal{L}(M)| + |\mathcal{L}(D \text{ using } M)| \quad (1)$$

The notion of comparing models based on simplicity can equivalently be expressed from a Bayesian perspective. The goal is to infer a model M from a set of observations D . Bayes's theorem states that the posterior probability of a model is:

$$p(M|D) = \frac{p(M)p(D|M)}{p(D)} \quad (2)$$

The most plausible model is then inferred by comparing these posterior probabilities:

$$\arg \max_M [p(M)p(D|M)] = \arg \max_M [\log p(M) + \log p(D|M)] \quad (3)$$

The trade-off inherent in both these approaches between simpler, more constrained networks and more complex, general networks echoes the *bias-variance dilemma* in statistics: introducing many parameters incurs high variance, while restricting the number of parameters incurs high bias in the set of possible solutions (Geman,

Bienenstock, and Doursat, 1992). MDL and Bayesian analysis (see BAYESIAN METHODS AND NEURAL NETWORKS) offer an approach to this dilemma by formalizing the Occam's razor idea—a complex network is preferred only when its predictions are sufficiently more accurate—as an inference rule.

The link between the two objectives (Equations 1 and 3) is provided by the *optimal coding theorem* (Shannon, 1948), which states that x can be communicated at a cost that is bounded below by $-\log_2 p(x)$ bits.

Applying this theorem produces the general MDL equation:

$$-\log p(M, D) = -\log p(M) - \log p(D|M) \quad (4)$$

Shannon's theorem describes the optimal code if the true probability distribution for a set of discrete alternatives is known. In general, however, one does not know the true distribution; so, because coding from a description language based on the wrong probability distribution will always take more bits on average, selecting an appropriate probability distribution for the codes is a key aspect of MDL applications. MDL provides a method of comparing these choices based on the resulting code lengths.

The distribution must be chosen to suit the nature of the task. For example, in a classification task, the data (given the model) consist of a number of discrete alternatives, each with some probability of occurrence. Here we can save bits by simply communicating the fact that the model output is correct on the correctly classified examples. When the information to be encoded takes on real values, a continuous distribution is required. A coding distribution that is often (implicitly) selected is a Gaussian. In this case, if we assume that the residuals (the second term in Equation 1) are independent and have a zero-mean, fixed-variance Gaussian distribution, then, if the values are encoded to some fixed accuracy, the code length is the familiar summed-squared-error cost function.

An Example of Applying MDL to Neural Networks

One of the standard approaches to improving generalization in neural networks can be formulated as an MDL technique. This approach adds an extra term to the error function that penalizes the complexity of the network, so that the objective function used to train the network involves a trade-off between the data misfit and the network complexity:

$$\text{Cost} = \alpha \text{ Complexity} + \text{Error} \quad (5)$$

If we regard each possible weight vector of the network as a potential model, then this complexity term is simply the cost of specifying the model in the definition of MDL above.

Applying Shannon's theorem then equates the complexity of a network to the negative log probability of its weights. Thus the critical question becomes the choice of encoding scheme, or prior distribution on the weights. A simple prior is a radially symmetric, mean-zero Gaussian. Setting the variance of this Gaussian to $1/\alpha$ yields an often used complexity term—the sum of the squares of the weights, $\sum_j w_j^2$. Differentiating this measure produces simple weight decay in the learning rule, which forces weights with small gradients from the error term to decay away, leaving only the required weights for the task. This example points out the key role of probability distributions in MDL; now we consider other encoding distributions and see how this choice affects the models that MDL favors.

Applications of MDL to Neural Networks

The MDL principle has been applied in a wide variety of areas over the past couple of decades. With respect to neural networks, the duality between Bayesian analysis and MDL means that a range of applications of Bayesian techniques to neural networks may also

be expressed in MDL terms. Few neural network methods have referred directly to minimal length encoding, but this relationship to Bayesian techniques makes many network methods relevant to this article.

We separate MDL applications into two classes involving fundamentally different formulations in terms of the communication protocol. In supervised learning, each data item consists of an input-output pair. The sender and receiver both have access to the inputs. The task of the sender is to succinctly communicate the desired outputs for each input. The network is a generative model of the output given the input; this model includes the network architecture and weights, while the data are the residuals.

In unsupervised learning, the receiver does not have access to the input, and the sender must provide enough information to allow an accurate input reconstruction. For example, in a clustering task, the sender first communicates the cluster centers (i.e., the weights of a competitive learning network). Then she only needs to say which cluster each input belongs to, and the residual error, or the distance to the cluster center. Given this information, the receiver can recover the actual input. Thus, here the network is a generative model of the input itself.

Note that this communication protocol formulation is only a device to derive an MDL objective function that can be used to train a neural network. The algorithms described below are not actually interested in sending the message, but rather in developing good models of a data set.

Supervised Learning

In supervised learning, the primary application of MDL techniques has been to improve the generalization performance of networks. Good generalization requires that the amount of information required to specify the output vectors of the training cases must be considerably larger than the number of independent parameters of the network. When only a small amount of labeled training data is available, a large network will not readily produce a good solution.

A range of techniques have been proposed to address this problem. These include weight sharing, weight/unit pruning (see *LEARNING NETWORK TOPOLOGY*), and cross-validation training. MDL techniques offer an alternative approach to this problem. In this section, we highlight two types of techniques. The first class assumes that the network architecture is given, and uses MDL to limit the complexity of the network weights. The second class uses MDL to select between potential network architectures.

Using MDL to determine network weights. The standard neural network training problem of finding the appropriate set of weights can usefully be formulated in MDL terms. We showed above that a radially symmetric Gaussian prior on the weights produces a weight decay term in the learning rule. Many different types of priors have been discussed in the literature. MacKay (1992) compared several priors for a single learning problem. He showed that for this problem, using the simple weight-decay prior with a different α for separate weight classes—weights into hidden units, hidden unit biases, and weights into output units—achieves better generalization than using a single α for all the weights.

More complicated priors on the weights can produce better generalization. For example, the prior could be a mixture of two zero-mean Gaussian distributions, a very wide one and a narrow one. The narrow Gaussian encourages small weight values to approach zero; the broader distribution takes responsibility for larger weights, and provides little pressure to change these values. The combined effect is to simplify the network by eliminating small weights.

The prior may also be adapted to the data. For many problems, such as translation-invariant recognition, improved network per-

formance can be achieved by constraining particular subsets of the weights to share the same value. Nowlan and Hinton (1992) accomplished this by fitting a mixture of Gaussians to the weights, allowing the network to decide which weights should be tied together.

Finally, Hinton and van Camp (1993) extended this work to consider the general MDL objective where the cost function is the sum of two encoding costs: the weights of the network and the output error residuals. On a sample problem, coding the weights using an adaptive mixture-of-Gaussians prior allowed the network to find three sharp clusters for the weights. Discovering this structure avoided overfitting the data, as the network was able to generalize even though the number of training cases was less than the dimensionality of the input vector.

Note that the MDL objectives described above involve several hyperparameters, such as the regularization constant α controlling the trade-off between the error and the complexity terms. Several methods have been proposed for determining hyperparameters (MacKay, 1992; Neal, 1996).

Recent work has extended MDL to include hyperparameters, which allows the application of MDL principles to a different class of learning methods. MDL formalizes the Occam's razor idea of finding the simplest model for a given data set. Yet the simplest model is not always one with few parameters. Neal (1996) showed that a particular neural network in the limit of an infinite number of parameters is a Gaussian process, which is actually a simple model that can be handled tractably. Other function approximation methods use a very large set of basis functions to model the data. Rasmussen and Ghahramani (2001) show that learning using these large models can also be expressed in MDL terms, where the model description applies not to parameters but instead to the functions included in the model.

Using MDL methods to determine a network architecture. MDL methods have also been applied to evaluate network architectures (see *BAYESIAN METHODS AND NEURAL NETWORKS*). The weights for various network architectures are learned using an MDL objective, and then these networks are compared based on the same data fit/complexity trade-off. MacKay (1992) demonstrated how this MDL/Bayesian procedure accurately predicts the appropriate number of hidden units on a small interpolation problem, where the target is determined by examining how well the various architectures generalize to an unseen test set.

Kendall and Hall (1993) proposed an MDL approach to network construction in which the model and data misfit are encoded over a discrete space. They computed the code length of the network parameters from a histogram of the weight values; since the task is assumed to be classification, the data code length is simply the cost of specifying which training cases are incorrectly predicted by the model. The authors found that using a genetic optimization algorithm to minimize the total description length succeeded in finding optimal network architectures on some simple problems.

MDL methods have been applied to architecture selection in other areas relevant to neural networks. Stolcke and Omohundro (1993) described an algorithm for inferring the structure of a hidden Markov model that involves a Bayesian/MDL approach. The algorithm begins with a model that directly encodes the training data, and then successively merges states based on a description length criterion that penalizes models according to the number of transitions and output values at each state. In addition, the MDL principle has been applied to learn the structure and parameters of *BAYESIAN NETWORKS* (q.v.), (e.g., Suzuki, 1999). Roughly speaking, a Bayesian network can be viewed as a form of neural network in which the nodes correspond to random variables and the weights encode conditional distributions. Encoding a Bayesian network with n nodes entails encoding the parents of each node and its set of con-

ditional probabilities. The MDL approach formalizes the trade-off between the simplicity of the Bayesian network structure and its ability to model the data.

Unsupervised Learning

For unsupervised learning networks, in which the goal of learning can be viewed as a form of probability density estimation, MDL techniques have been applied in several ways. The most popular approach is to use MDL to learn the number of distributions in the estimate, as well as the parameters of those distributions. A second approach involves applying MDL methods not only to the weights of the network, but also to the activities of the hidden units.

MDL and finite mixture models. A standard statistical technique for unsupervised classification can be formulated in an MDL framework. In finite mixture models, each datum in a training set is assumed to be drawn from one of J different classes. While learning the class parameters, we can also determine the optimal J by maximizing the posterior distribution of this number given the data set. This search for J resembles the architecture selection approach described above.

AutoClass (Cheeseman et al., 1988) is an unsupervised classification system based on this approach. It determines J by starting with more classes than are believed to be present and iteratively eliminating them. A prior that makes classes accounting for little data improbable can be seen as a description length prior on the model complexity. Similar approaches have been used in clustering algorithms, where a complexity term penalizing complex clusterings (based on measures such as their summed entropy) is added to create an MDL-style objective trading of data fit for model simplicity (see DATA CLUSTERING AND LEARNING).

MDL and autoencoders. In the unsupervised schemes described above, the model cost was some function of the number of underlying components in the clustering or mixture model algorithm, while the data cost was simply the summed-squared error. This represents one type of MDL objective function. If we adopt a more general viewpoint, we see that a wide range of other objective functions is possible.

In particular, we can view unsupervised learning in terms of an autoencoder (i.e., a network that attempts to reproduce its inputs on its outputs) where the MDL objective is to minimize the total cost of communicating the input vectors to a receiver. There are three terms in the description length:

- The *representation cost* is the number of bits required to communicate the representation (the hidden unit activities) that the algorithm assigns to each input vector.
- The *model cost* is the number of bits required to specify the hidden-to-output weights of the network, which provide an estimate of the input from its representation.
- The *reconstruction cost* is the number of bits required to fix up errors in the estimate of the input.

The sum of these three terms provides an objective function for training the autoencoder.

We can view many unsupervised algorithms in terms of this framework by understanding how they encode each of these three terms. For example, in competitive learning, the representation is the identity of the winning hidden unit, so the average representation cost is at most the logarithm of the number of units, while the reconstruction cost is proportional to the squared difference of the winner's weight vector and the input. Standard competitive learning algorithms minimize this latter cost, while algorithms that attempt to limit the number of clusters can be seen as trading off

representation cost for reconstruction cost. Principal Component Analysis (PCA) can be viewed as a version of MDL in which we ignore the model cost and the representation cost and minimize the reconstruction cost. Factor analysis can be viewed as a version of MDL in which we ignore the model cost but minimize the representation cost and reconstruction cost.

Many new algorithms can be derived by adopting new assumptions about the structure of the data and using them to formulate different methods of encoding the three terms in this MDL cost function. Any method that communicates each hidden activity independently will tend to lead to factorial representations, because any mutual information between hidden units will cause redundancy in the communicated message, so the pressure to keep the message short will squeeze out the redundancy. Zemel (1994) and Hinton and Zemel (1994) describe algorithms derived from this MDL approach for learning factorial representations; Zemel and Hinton (1995) describe how this MDL approach can also be used to develop population codes in which the activities of hidden units are locally correlated so as to form a topographical map. See UNSUPERVISED LEARNING WITH GLOBAL OBJECTIVE FUNCTIONS for a discussion of other algorithms that can be expressed within the MDL framework.

Discussion

MDL methods have been applied in a variety of neural network training paradigms, both for learning the weights and for assembling the architecture. These methods involve selecting a class of models and an encoding scheme for the two terms in the description: the model and the data. Given these elements, an appropriate objective function can be constructed for either an unsupervised or a supervised learning problem. Because of a common underlying framework, many Bayesian inferencing methods can also be viewed in MDL terms.

Many important issues remain to be explored in this area. One key issue concerns when it is better to use other methods of improving generalization, such as stopping training based on performance on a validation set, versus using an MDL-based regularization. In the unsupervised learning area, an open problem concerns formulating appropriate priors for learning hierarchical representations. Finally, an area that is ripe for MDL applications is temporal learning, in which MDL can be used to develop concise models that can accurately predict sequential events.

Road Map: Learning in Artificial Networks

Related Reading: Bayesian Methods and Neural Networks; Helmholtz Machines and Sleep-Wake Learning; Learning Network Topology; Unsupervised Learning with Global Objective Functions

References

- Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D., 1988, AutoClass: A Bayesian classification system, in *Proceedings of the Fifth International Conference on Machine Learning*, pp. 54–62.
- Geman, S., Bienenstock, E., and Doursat, R., 1992, Neural networks and the bias/variance dilemma, *Neural Computat.*, 4:1–58.
- Hinton, G. E., and van Camp, D., 1993, Keeping neural networks simple by minimizing the description length of the weights, in *Sixth ACM Conference on Computational Learning Theory*, Santa Cruz, CA.
- Hinton, G. E., and Zemel, R. S., 1994, Autoencoders, minimum description length, and Helmholtz free energy, in *Advances in Neural Information Processing Systems 6* (J. D. Cowan, G. Tesauro, and J. Alspector, Eds.), San Mateo, CA: Morgan Kaufmann, pp. 3–10.
- Kendall, G., and Hall, T., 1993, Optimal network construction by minimum description length, *Neural Computat.*, 5:210–212.

- Li, M., and Vitanyi, P. M. B., 1993, *An Introduction to Kolmogorov Complexity and Its Applications*, Reading, MA: Addison-Wesley.
- MacKay, D., 1992, A practical Bayesian framework for backpropagation networks, *Neural Computat.*, 4:448–472.
- Neal, R., 1996, *Bayesian Learning for Neural Networks*, New York: Springer-Verlag. ♦
- Nowlan, S. J., and Hinton, G. E., 1992, Simplifying neural networks by soft weight sharing, *Neural Computat.*, 4:173–193.
- Rasmussen, C. E., and Ghahramani, Z., 2001, Occam's razor, in *Advances in Neural Information Processing Systems 13* (T. Leen, T. Dietterich, and V. Tresp, Eds.), Cambridge, MA: MIT Press, pp. 294–300.
- Rissanen, J., 1989, *Stochastic Complexity in Statistical Inquiry*, Singapore: World Scientific Publishing.
- Shannon, C. E., 1948, A mathematical theory of communication, *Bell System Tech. J.*, 27:379–423, 623–656.
- Stolcke, A., and Omohundro, S., 1993, Hidden Markov model induction by Bayesian model merging, in *Advances in Neural Information Processing Systems 5* (S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds.), San Francisco: Morgan Kaufmann, pp. 11–18.
- Suzuki, J., 1999, Learning Bayesian belief networks based on the minimum description length principle: Basic properties, *IEICE Transactions, Fundamentals*, E82-A, 9.
- Zemel, R. S., 1994, A minimum description length framework for unsupervised learning, Ph.D. diss., University of Toronto.
- Zemel, R. S., and Hinton, G. E., 1995, Developing population codes by minimizing description length, *Neural Computat.*, 7:549–564.

Model Validation

Joachim M. Buhmann

Introduction

Mathematical models of interacting neuronal assemblies occur in brain theory and neural networks research in various ways. In *computational neuroscience*, highly sophisticated and detailed computer models of neurobiological phenomena are developed to gain insight into the complex nonlinear behavior of interacting neuronal populations and synaptic circuitry. The key question—how well does a selected neural network model describe the neurobiology of a neuronal population studied in vivo, i.e., the interactions and the dynamics of living neurons?—should be answered in the tradition of the biological sciences and of scientific investigations into neurobiological phenomena (see PERSPECTIVE ON NEURON MODEL COMPLEXITY). In artificial neural networks, assemblies of neuronal units are combined to represent statistical estimators for supervised learning tasks, i.e., classification and regression, or they model the data source under investigation in an unsupervised fashion by self-organizing maps (see SELF-ORGANIZING FEATURE MAPS), recurrent networks (see RECURRENT NETWORKS), or even more complicated structures.

In both situations the data analyst tries to infer functional dependencies from empirical data (Vapnik, 1982). The usefulness of a model is determined by its ability to capture those properties of the real world that are of interest to the data analyst. Mathematically, we have to measure or estimate the difference between the neurobiological system under investigation and our model simulation. A good model fit should reproduce the behavior of the studied system in the relevant parameter range that is supposed to be explained by the model study. Outside of this parameter range we tolerate large deviations between the model system and the real system. In computational neuroscience as well as in artificial neural networks (ANNs), the model complexity has to be controlled to avoid the following two modeling errors: too simplistic models usually miss essential features of the considered process or system (underfitting), whereas too complicated models adapt to stochastic fluctuations in the data without revealing reliable and useful knowledge (overfitting).

The trade-off between too simple and too complex models is systematically studied in statistics. The statistical literature on simulation (see Deaton, 1988) distinguishes the following five steps in the design of a well-planned simulation study: (1) system identification, (2) model development, (3) model verification, (4) model validation, and (5) model analysis. The first three items in the list summarize the necessary design steps to build a qualitatively correct model. After identifying the essential functional dependencies

of a system and their parametric description, the modeler is in a position to develop a conceptual abstraction of these relations. It is important for the success of a simulation study, in particular for model development, to explicitly state the scientific questions that should be answered by the conceptual model. The complexity of reality is never fully captured by the model world, and the modeler should ensure that the conceptual abstraction is appropriate to understand the relevant scientific issues under discussion. In step 3, an implementation of this abstraction has to be verified to guarantee that the computer simulation reflects the behavior and properties of the model. After ensuring the correctness of the implementation, we validate the model in the fourth step by comparing the simulated behavior with observations of the real system. The last step describes the indirect study of real-world phenomena by studying the behavior of the simulation model. The crucial step (step 4) between the model synthesis part (steps 1–3) and the analysis part (step 5) assesses how appropriate the model is to gain insight in the real-world system. In the most likely case of an imperfect model with deviations between model and reality, the *model quality* must be measured by an appropriate weighting according to the relevance of these deviations for answering the scientific questions under investigation. Models might even be developed for one scientific question but be found, on validation, to be more appropriate for another phenomenon than originally intended. For example, the Ising model was supposed to capture the essential collective features of magnetism, but it seems to work even better in studies of surface adhesion, while the wide use of Ising models in neural network theory was never anticipated by his designer.

Model Quality

How should we compare the model's behavior with the real-world system? We will look into this question first for simulation models in general and then for the more restricted class of probabilistic models for classification and regression.

The deviation of a simulation model from its counterpart in the real world is difficult to quantify, since it strongly depends on the purpose of the model. For the sake of concreteness, let us assume that we have implemented a network of locally coupled neurons of the FitzHugh-Nagumo type (Koch, 1999), a simplified version of the Hodgkin-Huxley equations for neuronal dynamics. This level of detail—two coupled nonlinear ordinary differential equations per neuron—is likely too crude for an experimental physiologist who studies depolarization effects in the cell membrane of neurons at the level of ion channel dynamics, but the same model might be

more than adequate to analyze collective synchronization phenomena, even at a quantitative level. It is important to emphasize that the fine details of neural dynamics should be taken into consideration in the case of a physiological study, and therefore deviations between simulations and experiments should be weighted more strongly than in a simulation study of the collective behavior of many coupled BvP oscillators.

The underlying scientific issue of the sufficient explanatory power of a model and its necessary simplicity is traditionally studied in the philosophy of science. The limits—how scientific theories should be designed and compared with reality—are investigated in the *theory of knowledge*, also known as epistemology. Different concepts and ideas have been proposed to explain how general laws can be inducted from a finite number of empirical observations. Since a logically conclusive inductive principle is considered to be impossible, philosophers of empiricism, in particular Karl Popper (1968), have focused on the concept of falsifiability of theories. Models and theories should be designed in such a way that they can be tested, and the degree of empirical testability should allow us to measure how well a theory has been confirmed by experimental observations. In the case of very complex models, the *Turing test* provides a crude experimental procedure to quantify the appropriateness of a model. A panel of experts is asked to distinguish between the real system and its model on the basis of the respective outputs or measurements of system parameters. Statistical tests can then be used to quantify the significance of the experts' answers and their deviations from chance events. A perfect simulation is reached if the experts are unable to distinguish between the real system and its simulation model in a statistically significant way.

A much simpler situation exists in *supervised learning* of classifiers and regression functions. The model quality of regression is defined by the expected deviation between the regression function $f(x; \theta)$ and the random variable y , which characterizes the noise-perturbed functional dependency under consideration. The regression function $f(x; \theta) \in \mathcal{H}$ is an element of an application-dependent hypothesis (function) class \mathcal{H} , which is indexed by θ . The true model θ_0 is often assumed to be perturbed by additive noise $y(x) = f(x; \theta_0) + \eta(x)$ with $E[\eta(x)] = 0$, $E[\eta(x)^2] = \sigma^2$ in regression, or by classification noise $y(x) = f(x; \theta_0)$ with probability $1 - p$ and $y(x) = 1 - f(x; \theta_0)$ with probability p . The joint probability $\Pr(x, y)$ of the data point x and the random variable y weights the influence of data on the model quality and determines the expected risk:

$$R(\theta) = \int \ell(y, f(x; \theta)) \Pr(x, y) dx dy \quad (1)$$

The most popular choices of loss functions for regression and classification are the quadratic loss $\ell(x, f) = (y - f(x; \theta))^2$ and the classification error $\ell(x, f) = \mathbf{1}(y \neq f(x; \theta))$ (0–1 loss), respectively. The goal of regression validation is to estimate the minimum $\theta^* = \arg \min_{\theta} E[R(\theta)]$ on the basis of a training sample set $\mathcal{X} := \{(x_1, y_1), \dots, (x_n, y_n)\}$ without explicit knowledge of the joint distribution $\Pr(x, y)$. The true model $f(x; \theta_0)$ does not necessarily have to be an element of \mathcal{H} . The induction principle *empirical risk minimization* (ERM) advocates calculating the minimum of the empirical risk:

$$\hat{R}(\theta; \mathcal{X}) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i; \theta)) \quad (2)$$

i.e., the empirical risk minimizer $\hat{\theta} := \arg \min_{\theta} \hat{R}(\theta; \mathcal{X})$ is used as an estimate of the expected risk minimizer θ^* .

Model Complexity and Model Selection

The validation of models in statistical learning of classifiers and regression curves is inspired by Popper's philosophy of science

and the requirement that scientific theories be falsifiable. Vapnik and Chervonenkis (see Vapnik, 2000) have analyzed the nature of statistical learning and have identified necessary and sufficient conditions for learning classifiers by empirical risk minimization. They analyzed the probability $\Pr(R(\hat{\theta}) - R(\theta^*) > \varepsilon)$ that the expected risk of the classifier with minimal empirical risk $\hat{\theta}$ exceeds the best classifier in the hypothesis class θ^* by more than a constant ε in costs or risk. In the case of a finite hypothesis class, the probability of this large deviation is bounded by the cardinality of \mathcal{H} times a fit factor that decays exponentially fast with the size n of the sample set (see Devroye, Györfi, and Lugosi, 1996), i.e.,

$$\Pr(R(\hat{\theta}) - R(\theta^*) > \varepsilon) \leq |\mathcal{H}| \sup_{f(\cdot, \cdot) \in \mathcal{H}} \Pr(|\hat{R}(\theta) - R(\theta)| > \varepsilon/2) \equiv \delta \quad (3)$$

where the probability on the right-hand side scales as $\mathcal{O}(\exp(-\lambda n))$. The trade-off between the complexity term $|\mathcal{H}|$ and the model-fitting factor $\sup_{f(\cdot, \cdot) \in \mathcal{H}} \Pr(|\hat{R}(\theta) - R(\theta)| > \varepsilon/2)$ allows us to determine a necessary sample size $n_0(\varepsilon, \delta)$ of the training set given the hypothesis class \mathcal{H} . Too complex hypothesis classes yield very loose bounds on the large deviations, despite the fact that we have used the best-performing classifier on the training data. Too simple hypothesis classes, on the other hand, introduce a significant model mismatch or bias, and they constrain the parameter space so severely that the signal in the data might not be captured by the learning algorithm.

How can we measure the complexity of a model when the cardinality is infinite, e.g., the space of polynomial discriminant functions of degree p or smaller? Obviously, the complexity depends on the degrees of freedom that are available to adapt to signal properties of the data set. To measure the complexity of a model, statisticians have developed sophisticated mathematical techniques from combinatorics and functional analysis. A very crude but for many practical applications sufficient rule of thumb counts the number of parameters in the model and requires that we provide at least ten data points per parameter to estimate the model. This rule implicitly assumes that all parameters increase the complexity of the model by an approximately constant amount. Consequently, applied statisticians have advocated adding a complexity penalty to the quality measure, which is linear in the number of parameters, like the AIC or the BIC criterion (see Hastie, Tibshirani, and Friedman, 2001). From estimating regression curves, however, we know that the slope of a linear regression function and the frequency of a trigonometric function add a vastly different amount of complexity to the model class. That is, oscillatory functions with a sufficiently high frequency allow us to approximate all data points in a bounded range with arbitrarily high precision, whereas linear functions cannot in general interpolate more than two points. The concept of covering numbers of function spaces measures this effect and relates the fitting power of functions to general properties of the underlying hypothesis (function) class. The *Vapnik-Chervonenkis dimension* and the *fat-shattering dimension* (Anthony and Bartlett, 1999) (see VAPNIK-CHERVONENKIS DIMENSION OF NEURAL NETWORKS) are derived parameters that measure the complexity of a hypothesis class for classification and regression independent of the probability distribution of the data.

Cross-Validation and Bootstrap

Cross-Validation

Apart from analytical techniques to derive bounds on the validity of models, a large number of numerical techniques have been developed to validate statistical models. Numerical methods for model validation split the data into three different subsets, the training set for parameter estimation, the validation set for selection of

the model class, and the test set for estimation of the prediction error. The validation set is used for training if the model class has been selected a priori. One of the most widely used techniques in classification and regression is the *k-fold cross-validation* method, which gains statistical precision at the expense of computation. The data set is split into k subsets of approximately equal size. The learning algorithm uses $k - 1$ of these subsets to estimate the model parameters, e.g., to find the empirical risk minimizer based on the data of these $k - 1$ subsets. The data of the k th subset, which have been set aside for model testing, are then used to estimate the model quality on future data. This procedure can be repeated k times, resulting in k different estimators. There exists a rich literature on how these different estimators can be combined, ranging from averages with different weighting schemes in regression (model averaging) to majority votes in classification. Popular versions of cross-validation are fivefold or tenfold cross-validation.

The validation set is required in cases where we not only have to estimate model parameters but also have to select an appropriate model order, e.g., to decide whether we should use a model with four parameters or one with five or more parameters. In such a data analysis scenario, we first train the model on the basis of the training data, then we validate our choice of the model order based on the estimated prediction error for the validation data. The model order with the best performance on the validation data is then selected. At the end of the data analysis process, the test data set is used to estimate the prediction error of the selected model, e.g., how well the selected model performs on future data. Using the test data rather than the validation data for model order selection—an unfortunately common practice among applied data analysts—usually leads to an overly optimistic estimate of the model error, since a selection bias in favor of low errors obscures the true error of the model.

Bootstrap

One of the drawbacks of cross-validation is the loss of training data for validation and testing. In the small-sample-size scenario, which occurs quite often in practical data analysis problems, we have too few data compared to the desired model complexity, and therefore we cannot afford to set aside some data exclusively for testing. Efron (see Efron and Tibshirani, 1993; Davison and Hinkley, 1997) has proposed a *resampling scheme with replacements*, called *bootstrap*, which is schematically summarized in Figure 1. Conceptually, we replace the unknown true probability distribution of the

data with the available empirical distribution. Sampling from this empirical distribution generates B bootstrap sample sets $Z^{*b} = \{(x_1^{*b}, y_1^{*b}), \dots, (x_n^{*b}, y_n^{*b})\}$, $1 \leq b \leq B$. A reasonable number B of bootstrap samples is between 25 and 200. Note that the bootstrap samples are not just permutations of each other, since we sample with replacements. Each of these bootstrap sample sets allows us to fit a model by estimating the model parameters with statistics S , e.g., the sample means, variances, or medians for mixture models. Finally, we combine these B bootstrap models by an appropriate averaging or merging procedure to derive an averaged model with supposedly more robust behavior than any single model.

To estimate the quality of the models fitted by bootstrapping, we choose those data in the training samples for testing that have not been selected for the bootstrap sample. Using the complete training sample set Z for testing would result in an overly optimistic model quality, since a fraction of approximately $(1 - 1/e)n \approx 0.632n$ of the data have been used in the bootstrap sample for model fitting. Improved and more robust estimators of model quality have been suggested by Efron and others, which are known as the $S^{(0.632)}$ estimator and variants of it (Hastie et al., 2001).

Discussion

Model validation in inference defines the core problem in learning and statistical estimation. When we induce a general model from a finite number of samples, we have to consider the complexity of the model in relation to the amount of data available for inference. Too complex models feign a high quality on small sample sets that is not confirmed on (future) test data. Analytical and numerical methods have been proposed over the last 40 years of neural network research to bound the deviations between empirical and expected risk of a statistical model. Bounds of the VC type usually contain a complexity term that accounts for the richness and flexibility of the hypothesis class, and a fitting term that measures the contraction of measure due to the large number of samples. Both influences have to be controlled, either by numerical methods like cross-validation and bootstrap or by analytical techniques from computational learning theory. The trade-off between model complexity and goodness of fit and its relation to the computational complexity of learning remains one of the deep challenges for future research on learning.

Road Map: Learning in Artificial Networks

Related Reading: Data Clustering and Learning; Vapnik-Chervonenkis Dimension of Neural Networks

References

- Anthony, M., and Bartlett, P. L., 1999, *Neural Network Learning: Theoretical Foundations*, Cambridge, Engl.: Cambridge University Press.
- Davison, A. C., and Hinkley, D. V., 1997, *Bootstrap Methods and Their Application*, Cambridge, Engl.: Cambridge University Press.
- Deaton, M. L., 1988, Validation of simulation models, in *Encyclopedia of Statistical Sciences*, (S. Kotz and N. L. Johnson, Eds.), New York: Wiley, vol. 8, pp. 481–484.
- Devroye, L., Györfi, L., and Lugosi, G., 1996, *A Probabilistic Theory of Pattern Recognition*, New York: Springer-Verlag. ♦
- Efron, B., and Tibshirani, R., 1993, *An Introduction to the Bootstrap*, London: Chapman and Hall. ♦
- Hastie, T., Tibshirani, R., and Friedman, J., 2001, *The Elements of Statistical Learning*, New York: Springer-Verlag. ♦
- Koch, C., 1999, *Biophysics of Computation*, Oxford, Engl.: Oxford University Press.
- Popper, K., 1968, *The Logic of Scientific Discovery*, New York: Harper.
- Vapnik, V. N., 1982, *Estimation of Dependences Based on Empirical Data*, New York: Springer-Verlag.
- Vapnik, V. N., 2000, *The Nature of Statistical Learning Theory*, 2nd ed., New York: Springer-Verlag.

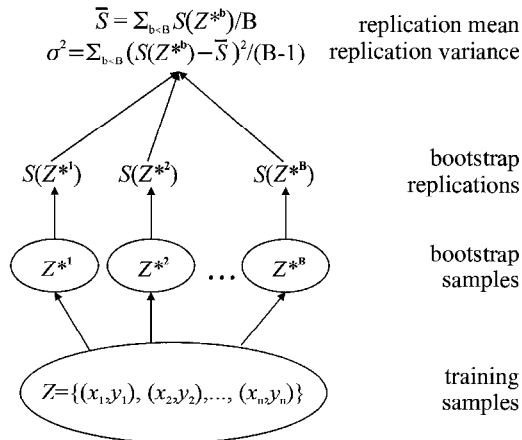


Figure 1. Processing pipeline of bootstrap estimation via bootstrap samples Z^{*b} and their associated statistics $S(Z^{*b})$.

Modular and Hierarchical Learning Systems

Michael I. Jordan and Robert A. Jacobs

Introduction

In this article we discuss the problem of learning in modular and hierarchical systems. Modular and hierarchical systems allow complex learning problems to be solved by dividing the problem into a set of subproblems, each of which may be simpler to solve than the original problem. Within the context of supervised learning—our focus in this article—modular architectures arise when we assume that the data can be well described by a collection of functions, each of which is defined over a relatively local region of the input space. A modular architecture can model such data by allocating different modules to different regions of the space. Hierarchical architectures arise when we assume that the data are well described by a multiresolution model—a model in which regions are divided recursively into subregions.

Modular and hierarchical systems present an interesting credit assignment problem—it is generally the case that the learner is not provided with prior knowledge of the partitioning of the input space. Knowledge of the partition would correspond to being given “labels” specifying how to allocate modules to data points. The assumption we make is that such labels are absent. The situation is reminiscent of the unsupervised clustering problem in which a classification rule must be inferred from a data set in which the class labels are absent, and indeed, the connection to clustering has played an important role in the development of the supervised learning algorithms that we present here.

The learning algorithms that we describe solve the credit assignment problem by computing a set of values—posterior probabilities—that can be thought of as estimates of the missing “labels.” These posterior probabilities are based on a probabilistic model associated with each of the network modules. This approach to learning in modular systems was developed by Jacobs et al. (1991). Jordan and Jacobs (1994) extended the modular system to a hierarchical system, made links to the statistical literature on classification and regression trees (Breiman et al., 1984), and developed an Expectation-Maximization (EM) algorithm for the architecture. We describe these developments in the remainder of the article, emphasizing the probabilistic framework.

The Mixture-of-Experts (ME) Architecture

The modular architecture that we consider is shown in Figure 1. The architecture is composed of N modules referred to as *expert networks*, each of which implements a parameterized function $\mu_i = f(\mathbf{x}, \boldsymbol{\theta}_i)$ from inputs \mathbf{x} to outputs μ_i , where $\boldsymbol{\theta}_i$ is a parameter vector. We attach a probabilistic interpretation to each of the expert networks by assuming that the experts generate outputs \mathbf{y} with probability $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_i)$, where μ_i is the mean of the conditional density P .

Because we assume that different expert networks are appropriate in different regions of the input space, the architecture requires a mechanism that identifies, for any given input \mathbf{x} , that expert or blend of experts that is most likely to produce the correct output. This is accomplished via an auxiliary network, known as a *gating network*, that produces as output a set of scalar coefficients g_i that serve to weight the contributions of the various experts. These coefficients are not fixed constants but vary as a function of the input \mathbf{x} .

The probabilistic interpretation of the gating network is as a *classifier*, a system that maps an input \mathbf{x} into the probabilities that

the various experts will be able to generate the desired output (based on knowledge of \mathbf{x} alone). These probabilities (the g_i) are constrained to be nonnegative and sum to one (for each \mathbf{x}).

There are many ways to enforce the probabilistic constraints on g_i . One approach is to utilize the *softmax* function, defined as follows. Let ξ_i denote an intermediate set of variables that are parameterized functions of the input \mathbf{x} :

$$\xi_i = \xi_i(\mathbf{x}, \boldsymbol{\eta}) \quad (1)$$

where $\boldsymbol{\eta}$ is a parameter vector, and define the outputs g_i in terms of ξ_i as follows:

$$g_i = \frac{e^{\xi_i}}{\sum_j e^{\xi_j}} \quad (2)$$

It is readily verified that the g_i are nonnegative and sum to one for each \mathbf{x} . This approach has the virtue of having a simple probabilistic interpretation: the ξ_i can be viewed as discriminant surfaces for a classification problem in which the class-conditional densities are members of the exponential family of probability distributions (Jordan and Jacobs, 1994).

The Mixture Model

Let us now specify the probabilistic model underlying the mixture-of-experts architecture more precisely. We assume that the training set $\mathcal{X} = \{(\mathbf{x}^{(l)}, \mathbf{y}^{(l)})\}_{l=1}^L$ is generated in the following way. Given the choice of an input \mathbf{x} , a label i is chosen with probability $P(i|\mathbf{x}, \boldsymbol{\eta}^0)$ (where the superscript 0 denotes the putative true values of the parameters). Given the choice of the label and given the input, the target output \mathbf{y} is assumed to be generated with probability $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_i^0)$. Each such data point is assumed to be generated independently in this manner.

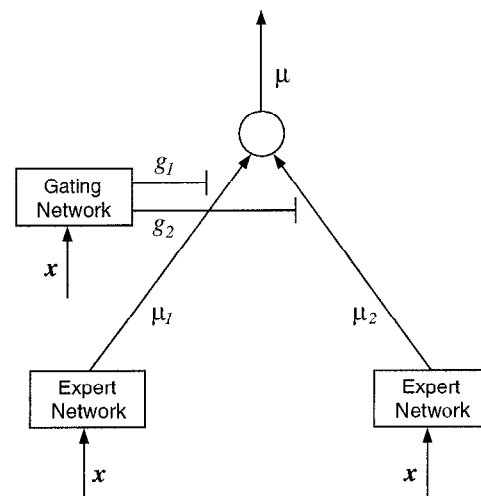


Figure 1. A mixture-of-experts architecture. The output μ is the conditional mean of \mathbf{y} given \mathbf{x} (see text).

Note that a given output \mathbf{y} can be generated in N different ways, corresponding to the N different choices of the label i . Thus, the total probability of generating \mathbf{y} from \mathbf{x} is given by the sum over i :

$$P(\mathbf{y}|\mathbf{x}, \Theta^0) = \sum_i P(i|\mathbf{x}, \boldsymbol{\eta}^0)P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_i^0) \quad (3)$$

where Θ^0 denotes the vector of all of the parameters ($\Theta^0 = [\boldsymbol{\theta}_1^0, \boldsymbol{\theta}_2^0, \dots, \boldsymbol{\theta}_N^0, \boldsymbol{\eta}^0]^T$). The density in Equation 3 is known as a *mixture density*. It is a mixture density in which the mixing proportions, $P(i|\mathbf{x}, \boldsymbol{\eta}^0)$, are conditional on the input \mathbf{x} .

It is the task of the gating network to model the probabilities $P(i|\mathbf{x}, \boldsymbol{\eta}^0)$, which can be construed as class probabilities in a multi-way classification problem of the input \mathbf{x} . We parameterize these probabilities via Equations 1 and 2, identifying the gating network outputs g_i with $P(i|\mathbf{x}, \boldsymbol{\eta})$.

It is straightforward to compute moments of the mixture density. For example, the conditional mean $\boldsymbol{\mu} = E(\mathbf{y}|\mathbf{x}, \Theta)$ is readily obtained by taking the expected value of Equation 3:

$$\boldsymbol{\mu} = \sum_i g_i \boldsymbol{\mu}_i$$

where $\boldsymbol{\mu}_i$ is the conditional mean associated with the probability distribution $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_i^0)$. The conditional mean is quite commonly used as the output of supervised learning systems, and this is reasonable in the mixture-of-experts setting as well, but only when no more than one value g_i is significantly different from zero for a given input \mathbf{x} . When more than one expert has a large value of g_i , however, the conditional distribution of \mathbf{y} given \mathbf{x} is multimodal, and it is important to make fuller use of the entire mixture density in such cases.

A Gradient-Based Learning Algorithm

To develop an algorithm for estimating the parameters of a mixture-of-experts architecture, we make use of the maximum likelihood (ML) principle. That is, we choose parameters for which the probability of the training set given the parameters (a function known as the *likelihood*) is largest. Taking the logarithm of the product of N densities of the form of Equation 3 yields the following log likelihood:

$$l(\mathcal{X}, \Theta) = \sum_i \log \sum_j P(i|\mathbf{x}^{(i)}, \boldsymbol{\eta})P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\theta}_i) \quad (4)$$

a function that we wish to maximize with respect to Θ . One approach to maximizing the log likelihood is to use gradient ascent (a better approach is to use the EM algorithm, as discussed in a later section). Computing the gradient of l with respect to $\boldsymbol{\mu}_i$ and ξ_i yields:

$$\frac{\partial l}{\partial \boldsymbol{\mu}_i} = \sum_j h_j^{(i)} \frac{\partial}{\partial \boldsymbol{\mu}_i} \log P(\mathbf{y}^{(j)}|\mathbf{x}^{(j)}, \boldsymbol{\theta}_i) \quad (5)$$

and

$$\frac{\partial l}{\partial \xi_i} = \sum_j (h_j^{(i)} - g_i^{(j)}) \quad (6)$$

where $h_i^{(j)}$ is defined as $P(i|\mathbf{x}^{(j)}, \mathbf{y}^{(j)})$. In deriving this result, we have used Bayes' rule:

$$P(i|\mathbf{x}^{(j)}, \mathbf{y}^{(j)}) = \frac{P(i|\mathbf{x}^{(j)})P(\mathbf{y}^{(j)}|\mathbf{x}^{(j)}, i)}{\sum_j P(j|\mathbf{x}^{(j)})P(\mathbf{y}^{(j)}|\mathbf{x}^{(j)}, j)}$$

where we have omitted the parameters to simplify the notation. This suggests that we define $h_i^{(j)}$ as the *posterior probability* of the

i th label, conditional on the input $\mathbf{x}^{(j)}$ and the output $\mathbf{y}^{(j)}$. Similarly, the probability $g_i^{(j)}$ can be interpreted as the *prior probability* $P(i|\mathbf{x}^{(j)})$, the probability of the i th label, given only the input $\mathbf{x}^{(j)}$. Given these definitions, Equation 6 has the natural interpretation of moving the prior probabilities toward the posterior probabilities.

An interesting special case is an architecture in which the expert networks and the gating network are linear and the probability density associated with the experts is a Gaussian with identity covariance matrix. In this case, Equations 5 and 6 yield the following on-line learning algorithm (on-line meaning that we have dropped the summation across l):

$$\Delta \boldsymbol{\theta}_i = \rho h_i^{(j)} (\mathbf{y}^{(j)} - \boldsymbol{\mu}_i^{(j)}) \mathbf{x}^{(j)T} \quad (7)$$

and

$$\Delta \boldsymbol{\eta}_i = \rho (h_i^{(j)} - g_i^{(j)}) \mathbf{x}^{(j)T} \quad (8)$$

where ρ is a learning rate. Note that both of these equations have the form of the classical LMS rule, with the updates for the experts in Equation 7 being modulated by their posterior probabilities.

It is also of interest to examine the expression for the posterior probability in the Gaussian case:

$$h_i^{(j)} = \frac{g_i^{(j)} e^{-1/2(\mathbf{y}^{(j)} - \boldsymbol{\mu}_i^{(j)})^T (\mathbf{y}^{(j)} - \boldsymbol{\mu}_i^{(j)})}}{\sum_j g_j^{(j)} e^{-1/2(\mathbf{y}^{(j)} - \boldsymbol{\mu}_j^{(j)})^T (\mathbf{y}^{(j)} - \boldsymbol{\mu}_j^{(j)})}} \quad (9)$$

This is a normalized distance measure that reflects the relative magnitudes of the residuals $\mathbf{y}^{(j)} - \boldsymbol{\mu}_i^{(j)}$. If the residuals for expert i are small relative to those of the other experts, then $h_i^{(j)}$ is large, otherwise, $h_i^{(j)}$ is small. Note, moreover, that the $h_i^{(j)}$ are positive and sum to one for each $\mathbf{x}^{(j)}$; this implies that credit is distributed to the experts in a competitive manner.

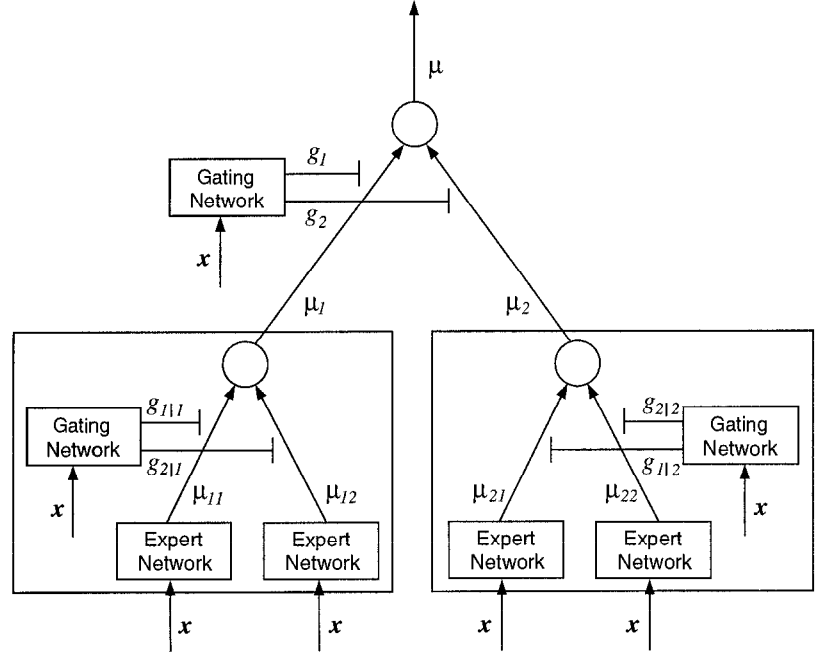
It is straightforward to utilize other members of the exponential family of densities as component densities for the experts, to allow dispersion (e.g., covariance) parameters to be incorporated in the model and to estimate the dispersion parameters via the learning algorithm (Jordan and Jacobs, 1994; Jordan and Xu, 1995).

The Hierarchical Mixture-of-Experts (HME) Architecture

The ME architecture solves complex function approximation problems by allocating different modules to different regions of the input space. This approach can have advantages for problems in which the modules are simpler than the large network that would be required to solve the problem as a whole. If we now inquire about the internal structure of a module, however, we see that the same argument can be repeated. Perhaps it is better to split a module into simpler submodules rather than to use a single module to fit the data in a region. This suggests a thoroughgoing divide-and-conquer approach to supervised learning in which a tree-structured architecture is used to perform multiple nested splits of the input space (Figure 2). The splitting process terminates in a set of expert networks at the leaves of the tree, which, because they are defined over relatively small regions of the input space, can fit simple (e.g., linear) functions to the data. This hierarchical architecture, suggested by Jordan and Jacobs (1994), has close ties to the classification and regression tree models in statistics and machine learning (e.g., Breiman et al., 1984). Indeed, the architecture can be viewed as a probabilistic variant of such models.

The mathematical framework underlying the HME architecture is essentially the same as that underlying the ME architecture. We simply extend the probability model to allow nested sequences of labels to be chosen, corresponding to the nested sequence of re-

Figure 2. A two-level binary hierarchical architecture. The top-level gating network produces coefficients g_i that effectively split the input space into regions, and the lower-level gating networks produce coefficients g_{ji} that effectively split these regions into subregions. The expert networks fit surfaces within these nested regions. Deeper trees are formed by expanding the expert networks recursively into additional gating networks and subexperts.



gions needed to specify a leaf of the tree. The probability model for a two-level tree is as follows:

$$P(\mathbf{y}|\mathbf{x}, \Theta) = \sum_i P(i|\mathbf{x}, \eta) \sum_j P(j|i, \mathbf{x}, \mathbf{v}_i) P(\mathbf{y}|\mathbf{x}, \theta_{ji}) \quad (10)$$

which corresponds to a choice of label i with probability $P(i|\mathbf{x}, \eta)$, followed by a conditional choice of label j with probability $P(j|i, \mathbf{x}, \mathbf{v}_i)$. This probability model yields the following log likelihood function:

$$l(\mathcal{D}, \Theta) = \sum_i \log \sum_i P(i|\mathbf{x}^{(i)}, \eta) \sum_j P(j|i, \mathbf{x}^{(i)}, \mathbf{v}_i) P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta_{ji}) \quad (11)$$

where, as in the one-level case, the prior probabilities $g_i^{(i)} = P(i|\mathbf{x}^{(i)}, \eta)$ and $g_{ji}^{(i)} = P(j|i, \mathbf{x}^{(i)}, \mathbf{v}_i)$ are defined in terms of underlying variables ξ_i and ξ_{ij} using the softmax function (cf. Equation 2). We also use Bayes' rule to define posterior probabilities in the obvious way:

$$h_i^{(i)} = \frac{g_i^{(i)} \sum_j g_{ji}^{(i)} P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta_{ji})}{\sum_j g_j^{(i)} \sum_k g_{kj}^{(i)} P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta_{jk})} \quad (12)$$

and

$$h_{ji}^{(i)} = \frac{g_{ji}^{(i)} P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta_{ji})}{\sum_j g_{ji}^{(i)} P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta_{ji})} \quad (13)$$

The posterior probability $h_i^{(i)}$ can be viewed as the credit assigned to the i th nonterminal in the tree, and the posterior probability $h_{ji}^{(i)}$ is the credit assigned to the branches below the nonterminals. The product $h_i^{(i)} h_{ji}^{(i)}$ is therefore the credit assigned to expert (i, j) .

A recursive relationship is available to compute the posterior probabilities efficiently in deep trees. The recursion proceeds upward in the tree, passing the denominator of the conditional posterior upward, multiplying by the priors, and normalizing (cf. the computation of h_i from h_{ji} in Equations 12 and 13).

We obtain a gradient ascent learning algorithm by computing the partial derivatives of l . If, as in the previous section, we assume linear experts and a linear gating network, as well as Gaussian probabilities for the experts, we obtain the following LMS-like learning algorithm:

$$\Delta \theta_{ji} = \rho h_i^{(i)} h_{ji}^{(i)} (\mathbf{y}^{(i)} - \mu_{ji}^{(i)}) \mathbf{x}^{(i)\top} \quad (14)$$

$$\Delta \eta_i = \rho (h_i^{(i)} - g_i^{(i)}) \mathbf{x}^{(i)\top} \quad (15)$$

and

$$\Delta \mathbf{v}_{ji} = \rho h_i^{(i)} (h_{ji}^{(i)} - g_{ji}^{(i)}) \mathbf{x}^{(i)\top} \quad (16)$$

Each of these partial derivatives have a natural interpretation in terms of credit assignment. Credit is assigned to an expert by taking the product of the posterior probabilities along the path from the root of the tree to the expert (cf. Equation 14). The updates for the gating networks move the prior probabilities at a nonterminal toward the corresponding posterior probabilities, weighting these updates by the product of the posterior probabilities along the path from the root of the tree to the nonterminal in question (cf. Equation 16).

An EM Algorithm

Jordan and Jacobs (1994) have derived an Expectation-Maximization (EM) algorithm for estimating the parameters of the ME and HME architectures. (See McLachlan and Krishnan, 1997, for a general treatment of the EM algorithm.) This algorithm, an alternative to gradient methods, is particularly useful for models in which the expert networks and gating networks have simple parametric forms. Each iteration of the algorithm consists of two phases: (1) a recursive propagation upward and downward in the tree to compute posterior probabilities (the *E-step*), and (2) solution of a set of local weighted maximum likelihood problems at the nonterminals and terminals of the tree (the *M-step*). Jordan and Jacobs (1994) tested this algorithm on a nonlinear system identification problem (the forward dynamics of a four-degrees-of-freedom robot arm) and reported that it converges rapidly, con-

verging nearly two orders of magnitude faster than backpropagation in a comparable multilayer perceptron network.

Discussion

Mixtures of experts should be compared to ensemble methods such as bagging and boosting (see ENSEMBLE LEARNING), which provide another general approach to building a supervised learning architecture out of collections of simple learners. In bagging and boosting, the overall input-output mapping is a convex combination of the members of the ensemble, much as in the case of mixtures of experts (in which the conditional mean is a convex combination of the outputs of the experts). However, the weights in this convex combination are constants in the case of bagging and boosting, whereas they are functions of the input for mixtures of experts. Moreover, in the mixture of experts, the weights have an interpretation as prior probabilities under a probabilistic mixture model and are explicitly parameterized as such. In particular, under this model, a single expert is assumed to be associated with each data point, an assumption that is not made for the ensemble methods. In essence, the mixture of experts approaches the supervised learning problem via a divide-and-conquer methodology reminiscent of unsupervised clustering, whereas ensemble methods approach the problem via superposition and averaging.

We conclude with a brief list of pointers to additional papers. The problem of model selection for HME architectures has been addressed by a number of authors. Several papers propose the use of greedy search procedures combined with some form of penalization for model complexity (Ramamurti and Ghosh, 1996; Saito and Nakano, 1996; Fritsch, Finke, and Waibel, 1997). Bayesian approaches for model selection or model averaging have also been presented, including methods based on Gibbs sampling (Jacobs, Peng, and Tanner, 1997) and variational approximation (Waterhouse, MacKay, and Robinson, 1994).

Theoretical analyses of the approximation and estimation rates for the ME (Zeevi, Meir, and Maierov, 1998) and the HME (Jiang and Tanner, 1998) are available. A basic result is that the HME achieves an approximation rate of $O(m^{-2/s})$, where m is the number of experts and s is the dimensionality of the input vector. Jordan and Xu (1995) present an analysis of the convergence rate for the EM algorithm for the HME. Kang and Oh (1997) provide an analysis of the HME using tools from statistical physics; in particular, they show that successive partitions in the HME can be analyzed as phase transitions.

There is a broad literature on engineering applications of the HME architecture, including applications to state-space filtering, optimization, control, vision, speech recognition, speaker identification, and time series analysis. Recent biological applications of the ME architecture have been presented by Haruno, Wolpert, and

Kawato (2001), who describe an architecture for human motor control based on a mixture of experts in which each expert is a paired forward-inverse model, and by Erickson and Kruschke (1998), who used the mixture of experts to build a model of human category learning that combines rule-based and exemplar-based representations.

Road Map: Learning in Artificial Networks

Related Reading: Competitive Learning; Sensorimotor Learning

References

- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., 1984, *Classification and Regression Trees*, Belmont, CA: Wadsworth International.
- Erickson, M. A., and Kruschke, J. K., 1998, Rules and exemplars in category learning, *J. Exp. Psychol. Gen.*, 127:107–140.
- Fritsch, J., Finke, M., and Waibel, A., 1997, Adaptively growing hierarchical mixtures of experts, in *Advances in Neural Information Processing Systems*, vol. 9 (M. Mozer, M. Jordan, and T. Petsche, Eds.), Cambridge, MA: MIT Press.
- Haruno, M., Wolpert, D. M., and Kawato, M., 2001, MOSAIC model for sensorimotor learning and control, *Neural Computat.*, 13:2201–2220.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E., 1991, Adaptive mixtures of local experts, *Neural Computat.*, 3:79–87. ♦
- Jacobs, R. A., Peng, F., and Tanner, M. A., 1997, A Bayesian approach to model selection in hierarchical mixtures-of-experts architectures, *Neural Netw.*, 10:231–241.
- Jiang, W., and Tanner, M. T., 1998, Hierarchical mixtures-of-experts for exponential family regression models: Approximation and maximum likelihood estimation, *Ann. Statist.*, 27:987–1011.
- Jordan, M. I., and Jacobs, R. A., 1994, Hierarchical mixtures of experts and the EM algorithm, *Neural Computat.*, 6:181–214. ♦
- Jordan, M. I., and Xu, L., 1995, Convergence properties of the EM approach to learning in mixture-of-experts architectures, *Neural Netw.*, 8:1409–1431.
- Kang, K., and Oh, J.-H., 1997, Statistical mechanics of the mixture of experts, in *Advances in Neural Information Processing Systems*, vol. 9 (M. Mozer, M. Jordan, and T. Petsche, Eds.), Cambridge, MA: MIT Press.
- McLachlan, G. J., and Krishnan, T., 1997, *The EM Algorithm and Extensions*, New York: Wiley.
- Ramamurti, V., and Ghosh, J., 1996, Structural adaptation in mixture of experts, in *Proceedings of the 13th International Conference on Pattern Recognition*, Los Alamitos, CA: IEEE Computer Society Press.
- Saito, K., and Nakano, R., 1996, A constructive learning algorithm for an HME, in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1268–1273.
- Waterhouse, S., MacKay, D., and Robinson, T., 1994, Bayesian methods for mixtures of experts, in *Advances in Neural Information Processing Systems*, vol. 8 (D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds.), Cambridge, MA: MIT Press.
- Zeevi, A., Meir, R., and Maierov, V., 1998, Error bounds for functional approximation and estimation using mixtures of experts, *IEEE Trans. Inform. Theory*, 44:1010–1025.

Motion Perception, Elementary Mechanisms

David C. Burr

Introduction

Visual motion is essential for many aspects of biological function, including rapidly detecting predators and prey, navigating through the visual environment, and constructing a three-dimensional visual representation from two-dimensional retinal input. However, motion information is not provided by the instantaneous retinal signal, but has to be computed from temporal variations in luminance over

the image. Although the neural mechanisms that achieve this vary considerably throughout the animal kingdom, the underlying principles of the algorithms seem to be very similar.

Models of Motion Perception

In biological visual systems, motion is initially analyzed in parallel by arrays of local motion detectors that exhibit certain basic prop-

erties: they require at least two spatially separate sampling units, one delayed with respect to the other, that are combined (usually nonlinearly) to create directional selectivity. Werner Reichardt (1961) was the first to provide a formal model of a motion detector based on these principles, in what has become known as a “correlator-type” model, or more simply, the *Reichardt detector*. The detector, at its simplest, is illustrated in Figure 1. The response of two spatially separated units ($\Delta\phi$ apart) are multiplied together (at M), after one has been delayed by ϵ . The figure illustrates two such units arranged as mirror images, symmetrically, using the same input. The unit on the left will respond best to rightward motion, maximally for speeds of $\Delta\phi/\epsilon$; that on the right will respond best to leftward velocities of $\Delta\phi/\epsilon$. Each unit M can be considered to be an elementary motion detector, in that it shows a direction preference. However, by combining the output of two such mirror-symmetrical units (subtractively in this case), the direction selectivity is further enhanced, to produce what is referred to as the *full Reichardt detector*.

The essential components of the Reichardt detector—spatial and temporal asymmetries and cross-correlation—can be implemented in many different ways. The initial model was inspired by the fly visual system, in which the two sampling points are adjacent ommatidia, and the temporal delay ϵ is introduced by some form of delay line, typically a low-pass filter. Models of human motion have been heavily influenced by the application of Fourier analysis to vision research, showing spatial and temporal filtering of the visual input at early stages. For moving stimuli, detectors are tuned in both space and time, leading to spatiotemporally oriented filters, or receptive fields. This concept has proven invaluable, not only in constructing physiologically plausible models of motion perception, but also in explaining how the form of moving objects is encoded (Burr and Ross, 1986).

One specific example of a model based on this concept is shown in Figure 2. The model starts with spatiotemporally oriented receptive fields tuned to a finite band of spatial and temporal frequencies, and hence to motion in a given direction (corresponding to a preferred orientation in the spatiotemporal plane). The orientation in space-time is readily achieved by linear combination of filters with appropriate spatial and temporal phase-shifts. In the particular model shown in Figure 2, the output of two such filters in quasi-quadrature phase in space and time, is squared then summed, to produce what has been termed “unidirectional motion energy.” This model responds to a drifting sinusoidal grating with a constant response, strongest when the velocity of the sinusoid corresponds to the orientation of the spatiotemporal receptive field,

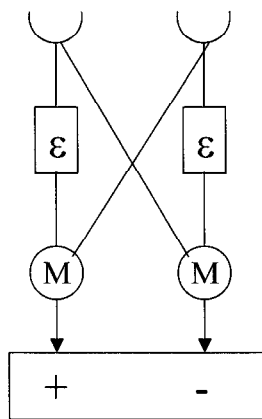


Figure 1. Simplified “full Reichardt detector.” (Adapted from Reichardt, 1961.)

Motion Energy Model

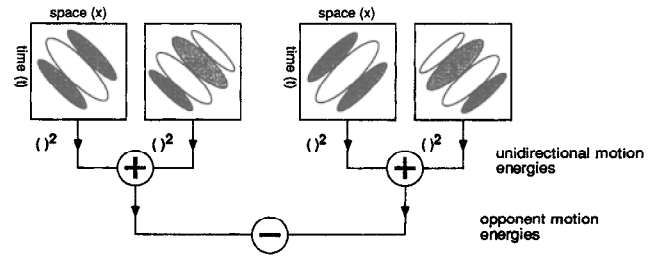


Figure 2. An example of a motion detector based on filters oriented in space-time. (Reproduced with permission from Adelson and Bergen, 1985.)

and weakest when in the orthogonal orientation (opposite direction). However, like the simple Reichardt detector, such a motion unit is not in itself a true motion detector, in that it will respond to many stationary transient stimuli, such as to a briefly flashed pattern of appropriate spatial frequency. Further specificity is achieved by inhibition between opponent motion energies, either by subtraction, as shown here, or by division. Interestingly, the full version of the motion energy model is formally equivalent to the full Reichardt motion detector, elaborated to include a spatial and temporal filtering stage, even though no part of the Reichardt detector corresponds to the unidirectional motion energy extractors (Adelson and Bergen, 1985).

Physiological measurements of neurons in macaque monkey visual cortex have identified plausible neural substrates for the two stages of the motion energy model (Qian and Andersen, 1994). Cells in the primary visual cortex V1 show directional selectivity, but also respond well to bidirectional motion; this is consistent with the expected performance of the first stage. However, cells in the middle temporal area (MT) show a strong inhibition by motion in the nonpreferred direction, consistent with opponent motion stage of the model. FMRI studies in humans provide support for this suggestion: V1 responds more strongly to counterphased sinusoidal gratings (that can be considered as the sum of two opposing drifting gratings) than to a single component drifting grating; whereas in MT complex, the result is reversed, with a much stronger response to the single component (Heeger et al., 1999).

Velocity Tuning

The selectivity to speed of the two motion detectors of Figures 1 and 2 can be varied by changing either the temporal or the spatial characteristics. For the Reichardt detector, the preferred speed can be increased either by increasing the spacing $\Delta\phi$ between the two sampling points, or by decreasing the delay ϵ . Similarly, for the energy model, where the spatial and temporal offsets are given by phase shifts, preferred speed will depend on both spatial and temporal frequency preference. In humans, it is possible to measure spatial and temporal selectivity, using a variety of techniques, including “masking,” in which one measures contrast sensitivity to a “test” stimulus in the presence of a high-contrast “mask.” The assumption is that the mask will cause maximum desensitization when its spatiotemporal characteristics match that of the detector responding to the test. To study motion perception, the test stimuli were drifting sinusoidal gratings of variable spatial and temporal frequency, displayed together with mask gratings, also varying in spatial and temporal frequency (Anderson and Burr, 1985). Over a wide range of spatial frequencies (0.025 c/deg to 15 c/deg), maximal masking occurs when the frequency of the mask matches that of the test. This suggests that there exist a battery of detectors with

preferred spatial frequency varying over this entire range, so that for any given test frequency the most sensitive detector will be tuned to that frequency; the most effective mask will therefore also be of that spatial frequency. For test frequencies lower than 0.025 c/deg or higher than 15 c/deg, maximum masking occurs not at the frequency of the test, but at 0.025 and 15 c/deg, respectively, suggesting that there do not exist motion detectors tuned to frequencies outside these bounds; a test of 0.01 c/deg will be detected by a mechanism tuned to 0.025 c/deg, so the most effective mask will be tuned to 0.025 c/deg, not 0.1 c/deg. In the temporal domain, the results are quite different. Maximal masking always occurs for masks near 10 Hz, irrespective of the temporal frequency of the test, implying that there is not a range of temporal tuning, but all detectors have similar temporal properties. Taken together, the results imply that in human vision, the variation in speed tuning is achieved not by varying temporal characteristics of the motion detector, but by varying spatial frequency preference, over a 600-fold range.

What is the range of speeds to which humans are sensitive? The lowest speed at which direction can accurately be discriminated is about 1 min/s for small stimuli moving over the fovea. This threshold increases steadily with eccentricity, reaching 8–10 min/s at 90° eccentricity (largely explained by the optical degradation in the periphery). However, the upper limit of motion detection is not a fixed speed but, as may be expected from the previous paragraph, varies considerably with the spatial frequency content of the stimuli (Burr and Ross, 1982). This is brought out clearly in Figure 3, showing contrast sensitivity (inverse of contrast thresholds) for biphasic bars (signal cycles of sinusoid) of various sizes, as a function of drift speed (abscissa). The small bars were seen best (required least contrast to discriminate their direction) when moving slowly, and could not be resolved at all at speeds above 100 deg/s. The largest bars, however, were best seen when moving at 500 deg/s, and could still be reliably resolved at 10,000 deg/s. Thus, the upper limit of motion perception is not so much a speed limit as a temporal frequency limit. The large variation in receptive field size ensures that human motion perception can operate over an extremely wide range of speeds, spanning nearly six orders of magnitude (0.015 to 10,000 deg/s).

Apparent Motion

Much of the motion we view daily at the cinema and on television is not real motion but an illusion created by displaying a series of still pictures in rapid succession (24 Hz for cinema, 60 Hz for

NSTC television). This type of motion is referred to as “apparent motion,” “stroboscopic motion,” or, most accurately, “sampled motion.” For some time it was thought that apparent motion may be detected by different processes from those detecting real motion, but recent studies find little justification for this view. Most motion detectors that incorporate spatiotemporal filtering will respond well to sampled motion, provided the sampling rate is sufficiently high. The spatiotemporal trajectory for apparent motion is a row of dots in space-time. If the spatiotemporal receptive fields (Figure 2) are oriented parallel to this trajectory, they will integrate the discrete samples, effectively causing the motion to become continuous (Burr and Ross, 1986).

The minimum theoretical sampling rate is given by the Nyquist limit, which requires that the image be sampled at at least twice the temporal frequency of image motion. Sampling below this frequency will cause *aliasing*, well-illustrated by the so-called “wagon-wheel” effect: periodic moving stimuli, such as wagon wheels in Westerns, are seen to stop and reverse direction as the wagon accelerates. When the repetition frequency of spokes exceeds half the sampling frequency (12 Hz for cinema), it will be undersampled, creating strong aliasing in the form of erroneous motion. The conditions under which sampled motion is indistinguishable from smooth motion can be predicted quantitatively from measurements of contrast sensitivity and linear systems analysis (Burr, Ross, and Morrone, 1986). Sampling a motion signal introduces spurious artifacts, whose frequency and amplitude depend on the sampling rate. Psychophysical measurements show that subjects are able to distinguish sampled from smooth motion if and only if the spurious frequencies produced by the sampling regime are not resolvable, as determined by measuring their thresholds for isolated sinusoids.

The spatiotemporally oriented receptive fields not only allow for the perception of discontinuous motion, but can also cause the image to be interpolated between the positions where it is displayed on each sample. The extrapolation is extremely accurate, and works over long ranges. Indeed, this property can be used to generate complex spatial forms from temporal information alone (Burr and Ross, 1986). When moving forms pass behind a “virtual slatted fence” (allowing information to be displayed only at discrete points), the visual system interpolates between the display points to give the impression of complete spatial forms. Thus, motion detectors not only encode velocity information about moving objects, but also participate in their spatial analysis.

Chromatic and Second-Order Motion

The examples discussed so far refer to motion of objects or images defined by luminance, typically bright or dark lines, sinusoidal grat-

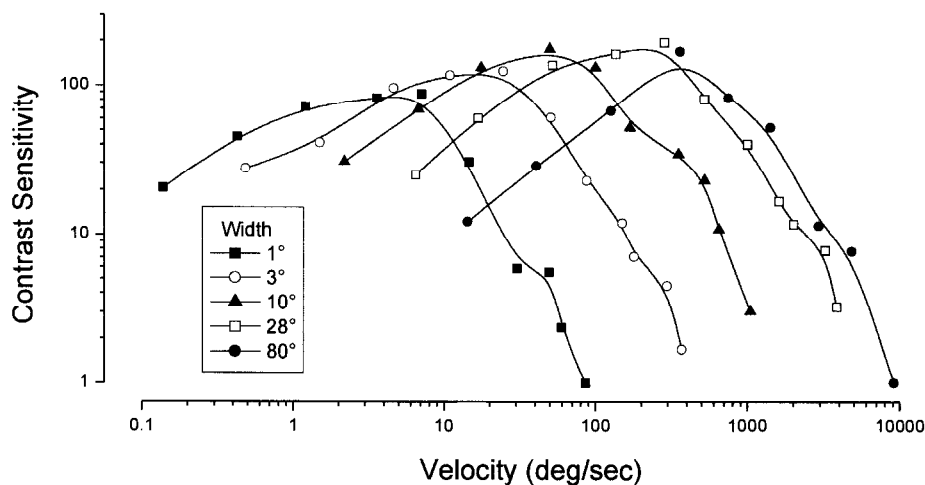


Figure 3. Contrast sensitivity for detecting the direction of motion biphasic bars of various sizes, as a function of speed. (Reproduced from Burr and Ross, 1982).

ings, or random dot patterns. However, luminance is not the only way to delineate objects: others include color, texture, and depth, and all these attributes can support motion. A well-studied example is the equiluminant class of stimuli, defined only by chromatic contrast. Movement of these stimuli yields a sensation of motion, albeit slower and jerkier than that for luminance patterns (Cavanagh, 1991).

Another very common stimulus in recent years is the class defined by variations in contrast, rather than luminance, giving rise to what is now called “second-order” motion (Chubb and Sperling, 1988). A typical example of second-order motion is a field of random dots multiplied (or amplitude-modulated) by a broad moving stimulus, typically a sinusoid. The interesting aspect of this stimulus is that although it gives rise to a strong and compelling sense of motion, neither the Reichardt detector of Figure 1 nor the motion-energy detector of Figure 2 would respond to it. However, a fairly simple extension can render both models sensitive to second-order motion: all that is needed is a “texture detector,” a filter responding to contrast instead of luminance, at the front stage, and the model will respond to amplitude-modulated motion. The “texture detector” need not be complicated: a simple half- or full-wave rectifier would suffice. It is still a debated point whether first- and second-order motions are detected by different neural structures, or by essentially the same mechanism with an add-on front-end texture detector. Evidence exists for both possibilities, such as mutual induction of aftereffects between the different types of motion, and differential selective activation during fMRI.

Two-Dimensional Motion

The models shown previously are essentially one-dimensional, discriminating leftward from rightward motion. There are various ways of extending these models to cover the two spatial dimensions, such as constructing many such units with spatial subfields oriented in various directions. Further spatial selectivity can be achieved by extending the spatial filters, or receptive fields, orthogonally to their direction of motion selectivity, emulating the physiological characteristics of receptive fields of mammalian vision. However, these two-dimensional motion units will demonstrate an inherent ambiguity about stimulus direction, usually referred to as the “aperture problem.” This stems from the fact that motion along a given trajectory can be decomposed into vectors spanning a range of 180° , so a vast range of detectors will be stimulated by any given trajectory (Figure 4). Various schemes have

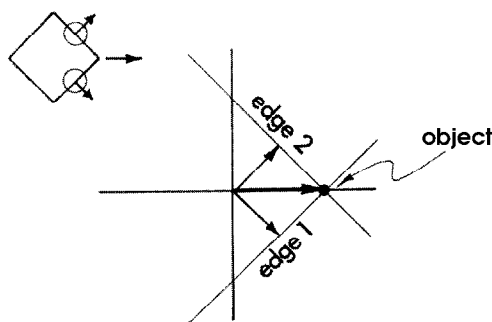


Figure 4. Illustration of the inherent ambiguity of two-dimensional motion. As the diamond moves rightward, the motion of the edges, within the receptive fields indicated by the circles, is diagonally upward or downward. In primate cortex, cells in V1 respond to the “component motion” of the edges, while some (but not all) cells in motion area MT respond to the direction of global motion (rightward).

been proposed for disambiguating the problem, usually involving the combination of signals from more than one detector, either in the form of a “vector sum” of motion units, or “intersection of constraints.” There is physiological evidence that the primate visual system adopts one of these schemes (Movshon et al., 1985). When stimulated with “plaids” (two orthogonal sinusoidal gratings) drifting in various directions, neurons in primary visual cortex V1 respond best when the direction of drift is such as to orient one or other of the components appropriately for that neuron, irrespective of the pattern drift. However, in the motion-specialized area MT, neurons respond best when the global motion of the plaid is in the appropriate direction, even though each component is then 45° off-axis. This suggests that as well as being responsible for the opponent stage of the motion detector, MT may help to disambiguate the two-dimensional direction of motion signals.

Other solutions have been proposed for the aperture problem, including the novel suggestion of Bill Geisler (1999; see also Burr, 2000). Geisler points out that given the temporal integration of the visual system, a small, localized target will leave a motion streak, much like the “speed lines” used by cartoonist to caricature motion. These static streaks provide potential information to disambiguate direction. A series of masking and motion aftereffect studies suggests that this spatial information is in fact integrated with motion information, and may help disambiguation. Another quite different class of experiment has shown that spatial structure of a certain type of moiré pattern can bias otherwise truly apparent motion, showing the influence of static structure on motion direction. Interestingly, however, although the moving streaks may be used to help sense motion, they are not perceived as streaks by the visual system. Although we integrate over time for 120 ms or so, the smear left by moving objects is far less, quite unlike what a camera with that shutter speed would record (Burr and Ross, 1986). Our motion detectors are based on receptive fields that are oriented in space-time, aligning themselves with the motion trajectory, and this should reduce the perceived blur.

This article has concentrated on basic motion mechanisms, the early mechanisms that analyze motion locally. Local-motion signals are combined in various ways, depending on the task. Analysis of optic flow requires integration of local-motion signals over large areas and complex trajectories. On the other hand, the ability to see transparent motion, and to localize accurately the position of small moving objects, requires that the local signals are kept distinct. How these conflicting goals are achieved is the subject of much modern research into motion perception.

Road Map: Vision

Related Reading: Directional Selectivity; Global Visual Pattern Extraction; Motion Perception: Navigation; Visual Cortex: Anatomical Structure and Models of Function

References

- Adelson, E. H., and Bergen, J. R., 1985, Spatiotemporal energy models for the perception of motion, *J. Opt. Soc. Am.*, A2:284–299.
- Anderson, S. J., and Burr, D. C., 1985, Spatial and temporal selectivity of the human motion detection system, *Vision Res.*, 25:1147–1154.
- Burr, D. C., 2000, Motion vision: Are “speed lines” used in human visual motion? *Curr. Biol.*, 10(12):R440–R443. ♦
- Burr, D. C., and Ross, J., 1982, Contrast sensitivity at high velocities, *Vision Res.*, 23:3567–3569.
- Burr, D. C., and Ross, J., 1986, Visual processing of motion, *Trends in Neuroscience*, 9:304–306. ♦
- Burr, D. C., Ross, J., and Morrone, M. C., 1986, Smooth and sampled motion, *Vision Res.*, 26:643–652.
- Cavanagh, P., 1991, Vision at equiluminance, in *Visual Function and Dys-*

- function: *Volume 5* (J. Cronly-Dillon, Ed.), London: Macmillan, pp. 234–250. ♦
- Chubb, C., and Sperling, G., 1988, Drift-balanced random stimuli: A general basis for studying non-Fourier motion perception, *J. Opt. Soc. Am.*, A5:1986–2007.
- Geisler, W. S., 1999, Motion streaks provide a spatial code for motion direction, *Nature*, 400:65–69.
- Heeger, D. J., Boynton, G. M., Demb, J. B., Seidemann, E., and Newsome, W. T., 1999, Motion opponency in visual cortex, *J. Neurosci.*, 19:7162–7174.

- Movshon, J. A., Adelson, E. H., Gizzi, M. S., and Newsome, W. T., 1985, The analysis of moving visual patterns, in *Pattern Recognition Mechanisms* (R. G. C. Chagas and C. Gross, Eds.), The Vatican, Pontificiae Academiae Scientiarum Scripta Varia, pp. 117–151.
- Qian, N., and Andersen, R., 1994, Transparent motion perception as detection of unbalanced motion signals. II. Physiology, *J. Neurosci.*, 14:7367–7380.
- Reichardt, W., 1961, Autocorrelation, a principle for evaluation of sensory information by the central nervous system, in *Sensory Communications* (W. Rosenblith, Ed.), New York: John Wiley, pp. 303–317.

Motion Perception: Navigation

Constance S. Royden and Ellen C. Hildreth

Introduction

When an observer moves through the world, the resulting image motion on the retina, known as *optical flow*, can inform the observer about his own motion through space and about the three-dimensional (3D) structure and motion of objects in the scene. This information is essential for tasks such as the visual guidance of locomotion through the environment and the manipulation and recognition of objects.

This article focuses on the recovery of observer motion from optical flow. We include strategies for detecting moving objects and avoiding collisions, discuss how this information may be used to control actions, and describe the neural mechanisms underlying heading perception.

The Image Flow Field

This section describes the relationship between two-dimensional (2D) image motion and the 3D translation and rotation of the observer relative to the scene. The mechanisms for deriving the 2D image velocities that result when the observer or objects move are described elsewhere (Hildreth and Koch, 1987; Mitche and Bouthemy, 1996; see also MOTION PERCEPTION: ELEMENTARY MECHANISMS).

Consider an observer moving relative to a stationary scene, with a coordinate system fixed to the observer and the z -axis directed along the optical axis. The instantaneous translation of the observer can be expressed in terms of translation along three orthogonal directions, given by the vector $\mathbf{T} = (T_x, T_y, T_z)^T$. Observer rotation can be expressed in terms of rotation around each of these axes, given by $\mathbf{R} = (R_x, R_y, R_z)^T$. Let $\mathbf{P} = (X, Y, Z)^T$ be the position of a point in space, as shown in Figure 1. The 3D velocity of \mathbf{P} in the observer's coordinate frame is given by

$$\mathbf{V} = (\dot{X}, \dot{Y}, \dot{Z})^T = -\mathbf{T} - \mathbf{R} \times \mathbf{P}$$

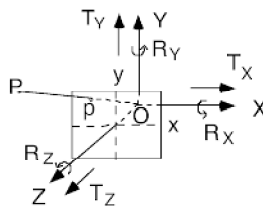


Figure 1. Coordinate system for a moving observer who is located at the origin.

where

$$\begin{aligned}\dot{X} &= -T_x - R_y Z + R_z Y \\ \dot{Y} &= -T_y - R_z X + R_x Z \\ \dot{Z} &= -T_z - R_x Y + R_y X\end{aligned}$$

If we assume perspective projection onto the image plane, using a focal length of 1, the projection of \mathbf{P} onto the image (x, y) is given by

$$x = X/Z, \quad y = Y/Z$$

The projected velocities in the image plane (v_x, v_y) are therefore

$$\begin{aligned}v_x &= (-T_x + xT_z)/Z + R_x xy - R_y(x^2 + 1) + R_z y \\ v_y &= (-T_y + yT_z)/Z + R_x(y^2 + 1) - R_y xy - R_z x\end{aligned}$$

The first term represents the component of image velocity due to observer translation and depends on the depth Z of each point in the scene. The remaining terms represent the component of image velocity due to the observer's rotation and do not depend on depth. The translational component yields a radial pattern of velocity (Figure 2A) that emanates from a single location in the image, called the *focus of expansion* (FOE). The FOE corresponds to the observer's heading and occurs at the location $(T_x/T_z, T_y/T_z)$ in the image. In contrast, the image flow field that results from a pure rotation of the observer is nearly constant over this region of the image. The image flow field for combined translation and rotation of the observer (Figure 2B) is the vector sum of the two flow fields from translation and rotation.

For observer translation, one can locate the FOE by finding the point of intersection of lines through the velocity vectors in the image. This simple strategy fails for combined translation and rotation, which occurs when the observer moves along a curved path or rotates his eyes or head, because the additional rotation components of velocity eliminate the FOE. This strategy also fails for nonrigid scenes that contain moving objects whose paths of motion deviate from the radial translational flow lines.

The Perception of Heading

Perceptual studies show that people judge 3D motion accurately under many conditions (Hildreth and Royden, 1998; Warren, 1998a; van den Berg, in Lappe, 2000). When translating toward a stationary scene, people exhibit discrimination thresholds as low as 0.2° when the heading is near the line of sight. Thresholds rise with more peripheral headings. Heading judgments are performed successfully with sparse, discontinuous flow fields, and require a relatively small field of view if the rotational flow is small. For pure translation, people recover heading with moderate accuracy

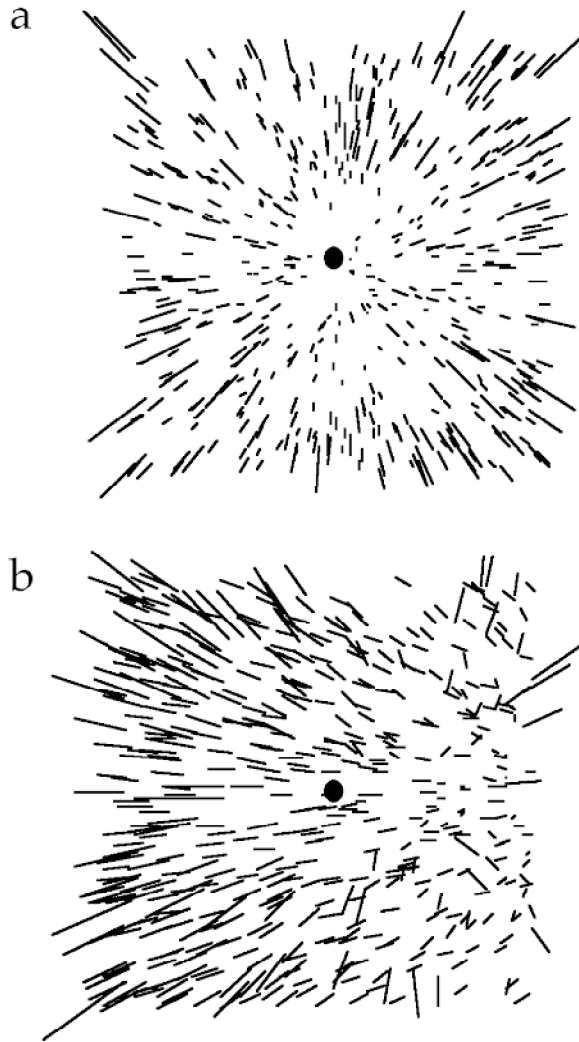


Figure 2. A, Radial flow field resulting from an observer moving in a straight line through a 3D cloud of dots. B, Flow field resulting from an observer translating and rotating (about a vertical axis) through a 3D cloud of dots. The solid circles indicate the direction of observer translation.

from only a 90 ms presentation, but improve up to about 300 ms of viewing time. Heading judgments remain accurate in the presence of moderate amounts of noise in the image flow field. The addition of other static and stereoscopic depth cues can enhance the accuracy of heading judgments in the presence of added noise or observer rotations.

People judge their translational heading accurately in the presence of small rotational rates generated by slow eye movements. Faster eye movements require extraretinal information about the speed of eye movement for accurate heading recovery (Hildreth and Royden, 1998; Warren, 1998a). In contrast, people accurately judge their motion along curved paths at low and high rotation rates (Warren, 1998a; van den Berg, in Lappe, 2000).

Under many conditions, a moving object has no effect on heading judgments. However, when the object crosses the observer's path, small biases in observer heading judgments result (Royden and Hildreth, 1996; Warren, 1998a). The ability to judge heading does not deteriorate when observers attend a second, object-related task (Hildreth and Royden, 1998).

These observations suggest that the human mechanism for judging heading from visual stimuli is remarkably robust and performs well under a variety of nonoptimal conditions.

Models of Observer Motion Recovery

Computational approaches for recovering heading can be divided into several categories, as described in this section. Many models fit more than one category; we present individual models within a category that best represents the particular approach.

Discrete Models

In discrete models, image features are tracked over time. Their sequence of positions forms the input to a system of equations whose solution yields the parameters of 3D structure and motion, assuming that the features move in a rigid configuration relative to the observer. Computer experiments indicate that such algorithms are vulnerable to error in the image motion measurements, although the use of motion measurements over an extended time can yield better performance (Martin and Aggarwal, 1988).

Differential Models

These models recover 3D motion and structure parameters from first and second spatial derivatives of the image flow field. One approach uses the *differential invariants* of the flow field, divergence (expansion/contraction), curl (rotation), and two components of deformation, dilation and shear (for references see Hildreth and Royden, 1998). Divergence and deformation depend only on the observer's translation and surface slant, and are invariant under observer rotations. In principle, these measures can be used to recover the observer's translation and the 3D shape of object surfaces. Most models that use differential invariants require a continuous, smooth optical flow field. In contrast, the human system can recover heading reliably from a few, sparse features that are sampled from a discontinuous flow field.

Motion Parallax Models

Motion parallax models use the fact that the translational components of the image velocities depend on the depth of the points in the scene, while the rotational components are independent of depth (Longuet-Higgins and Prazdny, 1980). Consequently, subtracting the image velocities from two points located at a depth discontinuity eliminates the rotational components. One can locate the translational heading using the resulting "difference vectors" by calculating the best point of intersection of lines through these vectors.

These models provide a method for quickly assessing heading, independent of the recovery of observer rotation and 3D scene structure. Because they combine information from multiple velocity vectors, they work fairly well in the presence of noisy velocity inputs. Simulations with a motion parallax model developed by Hildreth (1992) show behavior consistent with that observed in earlier perceptual studies. Motion differences computed by neurons in the middle temporal (MT) area of the primate visual system may be used to compute observer translation in the presence of rotations (Royden, 1997).

Error Minimization Models

Error minimization models compute observer motion and 3D structure parameters that yield a flow field that best fits the measured optical flow. For example, Bruss and Horn use this approach to derive observer motion parameters and surface structure that best

account for the measured flow field in a least-squares sense (see Hildreth and Royden, 1998). The error minimization strategy, together with spatial pooling of motion measurements over an extended image region, allows the algorithm to tolerate substantial error in the individual image motion measurements. Many models proposed for recovery of observer motion incorporate some form of error minimization. Notably, Heeger and Jepson (1992) presented an error minimization model that has been implemented in a neural network form by Lappe and Rauschecker (in Lappe, 2000).

Template Models

Template models use special-purpose computational mechanisms, such as a family of templates, tailored to detect patterns of optical flow corresponding to specific observer motion parameters. For example, a template for detecting forward motion along the line of sight would respond optimally to a radially expanding pattern of image velocities whose FOE is located at the center of the visual field. Template models deal effectively with noise in the input velocity measurements by integrating over a large area. Perrone and Stone (1994) proposed a template model that computes heading using components that respond to motion similarly to neurons in the primate visual area MT.

Eye Movement Models

In addition to retinal information, the human visual system can use the oculomotor signal to obtain information about eye rotation, either from an efference copy of the signal or from proprioceptive feedback from the extraocular muscles. Royden, Crowell, and Banks suggest that this information is essential for recovering heading accurately in the presence of fast eye rotations. The flow field corresponding to a known eye rotation could be subtracted from the overall flow field before the observer's heading is calculated (Hildreth and Royden, 1998). Lappe (in Lappe, 2000) and van den Berg and Beintema (see Lappe, in Lappe, 2000) have developed neural models that explicitly incorporate eye movement signals.

Cutting (1986) noted that when an observer fixates a point in space, the most rapidly moving objects in the vicinity of the fixation point can be used to judge heading relative to the fixation direction. One can locate the heading with successive fixations on objects in the scene. This model requires little computation from the flow field itself; however, it fails for certain configurations of scenes and eye fixations. It also requires multiple saccades to locate heading, something that is not essential for human heading judgments.

Neural Network Models

Several neural network models use training algorithms to learn to compute heading from optic flow input (Lappe, in Lappe, 2000). Hatsopoulos and Warren created a two-layer neural network that is trained using the Widrow-Hoff learning rule to recognize the correct translational heading for an observer moving along a straight line. The input layer consists of units tuned to direction and speed of motion. After training, the weights connecting the input and output layers adapt so that the output neurons detect radial patterns of motion corresponding to particular headings. The network only interprets flows derived from pure observer translation. Zemel and Sejnowski developed a learning network that segments the scene according to the motion of objects relative to the observer. Heading can be estimated from the resulting encoding.

Motion on a Curvilinear Path

When an observer moves along a curvilinear path, his instantaneous translation and rotation are the same as those for an observer pur-

suing straight-line motion with eye movement, resulting in an ambiguity. Distinguishing these situations requires an analysis of the flow field over an extended time. Alternatively, eye movement information may be used to disambiguate these conditions. Human observers distinguish curved from straight paths with high accuracy and judge the path curvature well. This finding suggests that the visual system computes both translation and rotation components of observer motion (Warren, 1998a; van den Berg, in Lappe, 2000).

Coping with Moving Objects

For most models, the presence of moving objects in the scene can adversely affect the derivation of observer motion. Image points associated with moving objects may move in a direction inconsistent with the observer's motion, causing errors in the heading estimate. Some models first detect moving objects and then compute heading from the remaining stationary components of the scene. Another approach computes an initial estimate of observer motion by combining all available data or by performing separate computations within limited image regions. One can then identify moving objects by finding areas of the scene for which the image motion differs significantly from that expected from these initial motion parameters (e.g., Hildreth, 1992). See Hildreth and Royden (1998) for a review of models of moving object detection.

Visuomotor Transformations for Navigation

Successful navigation requires that visual information be used to control motor actions to move through the world. This requires a transformation between the retinocentric heading coordinates computed by the models described in the previous section to a body-centered coordinate system, taking into account eye and head movements. It seems likely that the visual system uses extraretinal information, such as eye movement and vestibular signals, to account for rotations of the head and eyes. Neurons in the medial superior temporal (MST) area of visual cortex may combine these extraretinal signals with visual information to compute the body-centric heading (see Andersen et al., in Lappe, 2000).

The mechanisms for transformation of the visual information into motor control commands are not yet understood, but several approaches have been described. In one approach, motor planning takes place based on the computed heading of the observer. For example, to reach a desired goal, the motor system could initiate turning commands that minimize the error between the computed heading and the direction to the goal. Visual feedback allows constant refinement of the motor strategy to keep errors in heading from accumulating (Warren, 1998b).

Another approach is based on specific tasks the observer must perform to navigate through the environment. Such tasks include steering toward a goal, pursuing prey, braking, avoiding obstacles, or computing time to contact with an approaching surface. It has been suggested that each of these tasks may be accomplished through a task-specific subsystem that uses only the information in the flow field necessary to complete the task (Aloimonos, 1997; Warren, 1998b). For example, time to contact can be computed from the ratio, τ , given as the ratio of size/(rate of size change). The coupling between the task-specific information and the resulting action can be modeled as a nonlinear dynamical system. For example, when steering toward a goal, visual information provides the angle, β , between the FOE and the direction of the goal. This angle can be used to control the observer's rate of turning. The result is a system with a stable fixed point at $\beta = 0$, corresponding to the observer heading toward the goal. Complex motor behavior may emerge through interactions between loosely coupled subsystems underlying different tasks (Warren, 1998b).

Neural Mechanisms of Heading Computation

In primates, visual area MST is probably involved in computing heading (Duffy, in Lappe, 2000). Neurons in MST respond well to large motion patterns and receive direct input from cells in area MT, which is known to process motion (see also MOTION PERCEPTION: ELEMENTARY MECHANISMS). Some MST neurons prefer expanding or contracting radial patterns of motion, as would be generated by an observer moving in a straight line forward or backward. These cells have different preferred centers of expansion, so they could be involved in finding the FOE in an optical flow field. Other cells respond well to uniform motion in a single direction, and yet others respond to rotating patterns of motion. Many cells respond to some combination of these.

It is unclear how these cell responses contribute to the computation of heading in the presence of rotations; however, several models have been developed that could explain this. Hatsopoulos and Warren (Warren, 1998a) and Perrone and Stone (1994) proposed template models that use components that behave similarly to neurons in area MT in their response to motion. In both models, these components connect to another layer of cells with properties similar to the cells in MST. The connection patterns are such that the cells in the second layer respond to spatial patterns that mimic the flow fields that result from particular observer motion parameters. The Hatsopoulos and Warren model deals only with pure observer translation. Perrone and Stone used templates that deal with combinations of translations and the rotations that result when the observer makes eye movements to track an object in the scene. This model also recovers the relative depths of surfaces in the scene.

Royden (1997) developed a model that makes use of the motion-opponent properties of MT neurons to deal with observer rotations. Many neurons in MT have both excitatory and inhibitory regions within their receptive fields. The Royden model uses operators with this receptive-field layout to eliminate the observer rotation at the initial processing stage. These cells project to a second layer of cells, similar to those in MST, that are tuned to radial patterns of input. As with the motion parallax models described earlier, the centers of these radial patterns correspond to observer headings.

Finally, Lappe (in Lappe, 2000) and van den Berg and Beintema (cited by Lappe, in Lappe, 2000) developed neural models that explicitly incorporate eye movement signals to deal with rotations generated by eye movements. In Lappe's model, extraretinal input compensates for the image motion induced by eye movements. In van den Berg and Beintema's model, the responses of template cells tuned to retinal flow are multiplied by a "rate-coded" measure of eye velocity, producing a layer of cells that have a preferred flow field that changes dynamically to compensate for eye movements.

Currently, there is insufficient physiological or psychophysical data to distinguish among these models of neural computation of heading. It seems likely that some compensation for eye movements occurs in area MST (see Andersen et al., in Lappe, 2000); however, this compensation could be incorporated into the models that do not currently use it. The models are all reasonably consistent with the known behavior of MT and MST cells. Determination of which, if any, most accurately describes the neural computation awaits further experimentation.

Discussion

People judge heading well under many conditions; however, it is still uncertain how the visual system accomplishes this task. Superficially, most of the models cited here exhibit general biological plausibility, in that they can be implemented by a network of simple, local processing mechanisms operating in parallel. Physiological observations reveal the general properties of the representation of optic flow information and provide some indication that heading computations take place in areas MT and MST of the primate visual system. It remains a challenge to incorporate all of the important aspects of recovery of 3D observer motion into a neuronal model that exhibits a broad range of human behavior and incorporates the details of physiological observations.

Road Map: Vision

Related Reading: Motion Perception: Elementary Mechanisms; Robot Navigation

References

- Aloimonos, Y., Ed., 1997, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, Mahwah, NJ: Erlbaum.
- Cutting, J. E., 1986, *Perception with an Eye for Motion*, Cambridge, MA: Bradford/MIT Press.
- Heeger, D. J., and Jepson, A. D., 1992, Subspace methods for recovering rigid motion: I. Algorithm and implementation, *Int. J. Comput. Vision*, 7:95–117.
- Hildreth, E. C., 1992, Recovering heading for visually-guided navigation, *Vision Res.*, 32:1177–1192.
- Hildreth, E. C., and Koch, C., 1987, The analysis of visual motion: From computational theory to neuronal mechanisms, *Annu. Rev. Neurosci.*, 10:477–533.
- Hildreth, E. C., and Royden, C. S., 1998, Computing observer motion from optic flow, in *High-Level Motion Processing: Computational, Neurobiological and Psychophysical Perspectives* (T. Watanabe, Ed.), Cambridge, MA: MIT Press, pp. 269–293. ♦
- Lappe, M., Ed. 2000, *Neuronal Processing of Optic Flow*, *Int. Rev. Neurobiol.* vol. 44. (special issue),
- Longuet-Higgins, H. C., and Prazdny, K., 1980, The interpretation of a moving retinal image, *Proc. R. Soc. Lond. B*, 208:385–397.
- Martin, W. N., and Aggarwal, J. K., Eds., 1988, *Motion Understanding: Robot and Human Vision*, Boston: Kluwer.
- Mitiche, A., and Boutheimy, P., 1996, Computation and analysis of image motion: A synopsis of current problems and methods, *Int. J. Comput. Vision*, 19:29–55. ♦
- Perrone, J. A., and Stone, L. S., 1994, A model of self-motion estimation within primate extrastriate visual cortex, *Vision Res.*, 34:2917–2938.
- Royden, C. S., 1997, Mathematical analysis of motion-opponent mechanisms used in the determination of heading and depth, *J. Opt. Soc. Am. A*, 14:2128–2143.
- Royden, C. S., and Hildreth, E. C., 1996, Human heading judgments in the presence of moving objects, *Percept. Psychophys.*, 58:836–856.
- Warren, W. H., 1998a, The state of flow, in *High-Level Motion Processing: Computational, Neurobiological and Psychophysical Perspectives* (T. Watanabe, Ed.), Cambridge, MA: MIT Press, pp. 315–358. ♦
- Warren, W. H., 1998b, Visually controlled locomotion: 40 years later, *Ecol. Psychol.*, 10:177–219.

Motivation

Alan G. Watts

Introduction

Motivated or goal-directed behaviors are sets of motor actions that direct an animal toward a particular goal object, an interaction that promotes the survival of an individual or maintains the species. Goal-directed behaviors consist of sleep/wake, ingestive, reproductive, thermoregulatory, and aggressive/defensive behaviors. At the simplest level, motivated behaviors can be considered the behavioral adjuncts of those physiological (i.e., homeostatic) processes concerned with maintaining the composition of the internal environment. They are often accompanied by emotion or affect.

Despite their apparent utility, defining the terms *drive*, *instinct*, and *motivation* with respect to the neural substrates of behavior has always been controversial (Grossman, 1979; Pfaff, 1982). Although the most rigorous use of the terms drive and motivation has been as intervening variables between stimulus and behavioral response, there is an attraction to trying to assign particular brain regions responsibility for putting the motivation into behavior. However, “neuralizing” drive, and particularly motivation, has often been criticized because it has been thought impossible to identify and measure their specific neural properties (e.g., Hinde, 1970).

With this in mind, a neural systems approach will be adopted here that downplays the notion of associating motivation with specific neural mechanisms. To this end, this article will concentrate more on discussing what and how particular parts of the brain contribute to the expression of behaviors that have a motivated character. The framework adopted here is based to a large extent on Hullerian incentive models of motivation (see Bindra, 1978, and Toates, 1986, for further discussion), where the probability of a particular behavior being expressed at any one time is dependent on the integration of sets of afferent information: information from systems that control circadian timing and regulate arousal state, inputs derived from interoceptive information that encode internal state (e.g., hydration state, plasma glucose, leptin), modulatory hormonal inputs such as gonadal steroids that mediate sexual behavior, and inputs derived from classic sensory modalities (i.e., exteroceptive information). The advantage of this approach for identifying neural substrates is that it allows us to utilize the common experimental paradigm of tracing how information derived from sensory inputs known to generate specific behaviors is distributed within the brain (e.g., Swanson and Mogenson, 1981). With the advent of sophisticated functional neuroanatomical methods, this approach is proving quite useful (Watts, 2001; Watts and Swanson, 2002).

Temporal Organization of Motivated Behavior

A scheme describing the temporal organization of motivated behavior was first outlined by Wallace Craig in 1918, and later elaborated by Mogenson and colleagues (Swanson and Mogenson, 1981). Here, behavior is initiated following interactions among sensory information, the neural systems that control arousal state, and those systems that control sensory object representation. These interactions determine the value of the “drive” associated with a particular behavior. In turn, the integral of competing drives then determines which series of actions will generate the most appropriate procurement (or appetitive) phase, where the goal object is actively sought. The motor events expressed during the procurement phase involve foraging behavior, are individualized for the particular situation, and can be quite complex.

When the goal object has been located, the subsequent consummatory phase involves more stereotypic rhythmic movements—

licking, chewing, copulating, etc.—that allow the animal to interact directly with the goal object. During the consummatory phase, how the animal structures the interaction (e.g., determines the duration and the amount consumed during a feeding episode, or the duration of the intermeal interval) is an important function that arises from the dynamic interaction of sensory inputs and the central neural networks that control motor function. Furthermore, reward/aversion functions, together with learning and memory, are also critical processes, particularly during the procurement phase; a previously rewarded or an aversive experience of a particular goal object, remembering where it is located, and remembering how to get there are important considerations that contribute to the integrative process. Finally, as the consummatory phase continues, interoceptive feedback signals are generated that increase the probability of its termination, most likely using inhibitory networks. Termination may also occur at any time following new exteroceptive signals (e.g., the presence of a predator) that override an ongoing behavior and allow the animal to switch immediately to another, more appropriate behavior (McFarland and Sibly, 1975).

Neural Substrates

At the simplest level, four broad-ranging neural systems are concerned with generating motivated behaviors: those involved with the transduction and processing of sensory signals, those that control arousal state and circadian timing, those involved with motor control, and those that process the types of information concerned with sensory object representation. These systems are represented at the simplest level in Figure 1 without reference to anatomical locus.

The notion of drive and the idea that particular behaviors are selected to reduce the level of specific drive states have together been very influential if somewhat controversial concepts in neuroscience. From the perspective of delineating neural systems, it is useful to think of drives as being dynamic properties within different sets of neural networks, each of which is concerned with regulating a specific motivated behavior. In this way, drives are properties of behaviorally specific networks within the motor control module of Figure 1. Drive states are determined by the inputs from sensory processing, arousal state control, and object representation systems. The values of drive states within these networks are altered by these inputs in a way that increases or decreases the probability of a particular behavior being expressed at any one time.

Figure 2 expands the scheme shown in Figure 1 to illustrate specific components within the object representational and motor control networks. It shows that there are four principal inputs that can activate motor systems. The most complex motor control processes are those that generate anticipatory behaviors. In some instances, information from systems controlling arousal state—for example, circadian timing—provide the predominant signals (input 1, Figure 2). But this type of anticipatory control often derives from interactions between processed sensory information and those forebrain systems concerned with encoding object representation, particularly learning and memory, reward, and spatial orientation and navigation. The integrated output of these regions then regulates motor control systems (input 2, Figure 2). However, increased drives for motivated behaviors can also be produced by hormones or internally generated deficit signals (e.g., the thirst arising from dehydration, or the hunger from starvation) that access motor control networks more directly (input 3, Figure 2).

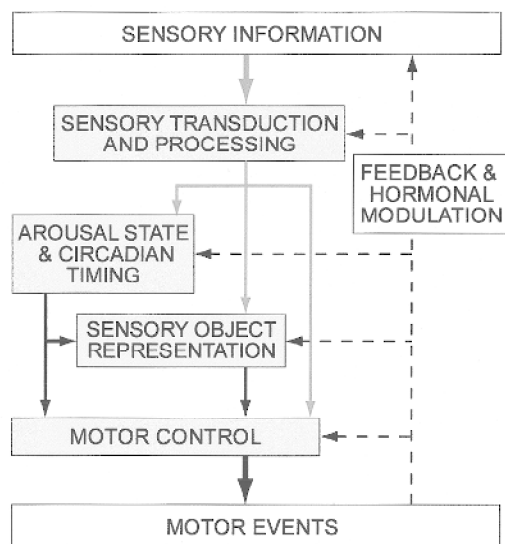


Figure 1. A schematic representation of the neural systems and their interactions involved with controlling motivated behaviors. Sensory inputs are shown in gray, central neural connections in black, and hormonal and feedback signals as dashed lines.

Collectively, inputs 1, 2, and 3 to the motor control networks can be thought of as “drive-determining” interactions. Mechanisms then integrate outputs from different drive networks to select the behavioral action most appropriate for reducing the drive state with the highest value, so initiating the appropriate procurement phase.

Finally, simple reflex actions are generated by direct sensory inputs to the premotor and motor networks with little higher-order processing (input 4, Figure 2). However, although these reflex actions lack any motivated character, they make important contributions to the consummatory phase of motivated behaviors.

Sensory Information

Neural systems that control motivated behaviors are regulated by a host of sensory inputs, which are defined either as interoceptive signals encoding internal state or as exteroceptive inputs that encode features of the goal object such as smell, taste, temperature, tactile properties, and appearance. Each of these sensory modalities has specific receptors, transduction mechanisms, and “labeled line” access to central processing networks located throughout the brain. Although important sensory processing occurs within the telencephalon, particularly sensory cortex, the initial sensory processing that occurs subcortically has important implications for controlling motivated behaviors; for example, altered sensitivity to the taste of sodium occurs in the hindbrain of hyponatremic animals and is an important adjunct to increased sodium appetite.

Some sensory signals directly access drive networks, as typified by the drinking initiated by increasing plasma osmolality or angiotensin II (A-II), or the deficit-induced feeding activated by adiposity signals (primarily leptin and insulin), which have direct hypothalamic actions (Elmqvist, Elias, and Saper, 1999). In both of these cases, however, it is not clear whether the outcome of processing the deficit signal in the hypothalamus requires close interaction with the object representation networks. The fact that hypothalamic regions involved with this type of sensory transduction project directly to regions concerned with ingestive motor control (Elmqvist et al., 1999) suggests that they may not.

Circadian Timing and Arousal State Control

Parts of the brain provide critical circadian timing information and control arousal state that enable motor command networks to generate anticipatory behaviors. The circadian timing system originates in the hypothalamic suprachiasmatic nucleus, which generates the signal that entrains virtually all neural activity within limits determined by the prevailing photoperiod. Catecholamine cell groups in the hindbrain (e.g., the locus coeruleus), histaminergic neurons in the tuberomammillary nucleus, the ventrolateral pre-optic nucleus, and the recently identified hypocretin/orexin neurons in the lateral hypothalamic area (LHA) supply information that is of critical importance for controlling arousal state.

Object Representation

Those systems that generate neural representations of sensory objects are important for controlling motivational behaviors. These include learning and memory mechanisms in the telencephalon and cerebellum; reward/aversion systems in the midbrain ventral tegmentum, parts of the basal forebrain (particularly the nucleus accumbens), amygdala, and parts of the cortex, particularly prefrontal regions; and systems in the hippocampus and parts of the parietal cortex responsible for allocentric and egocentric spatial representation. A great deal of exterosensory information is processed

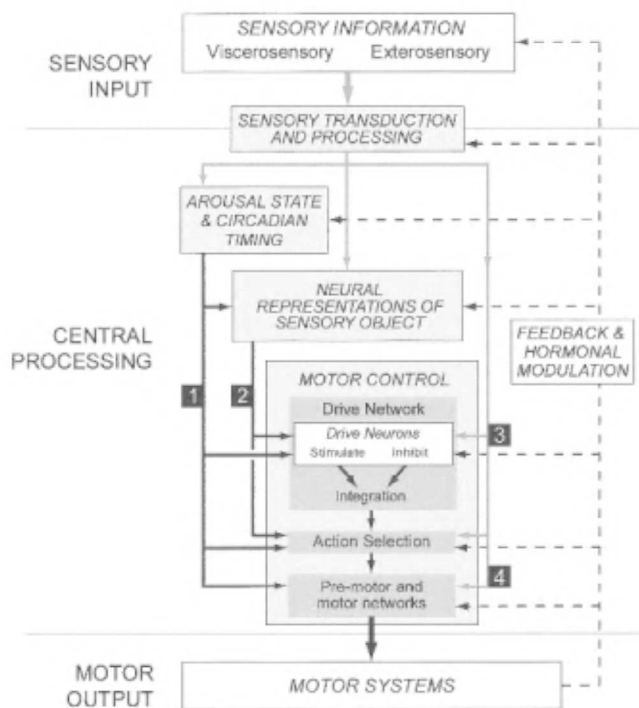


Figure 2. Motor control networks are organized at three levels: drive networks, which can either stimulate or inhibit behaviors; action selection networks, which integrate the outputs of drive networks with those of other systems; and executive premotor and motor neuron networks. The generation of motivated behavioral actions by motor control networks can be initiated by four different sets of inputs: (1) from systems controlling arousal state and circadian timing; (2) from systems that generate representations of sensory objects; (3) directly from modulatory hormone and the sensory signals that encode physiological deficits; and (4) from sensory signals that generate reflex actions by interacting directly with premotor and motor neuron networks. Sensory inputs are shown in gray, central neural connections in black, and hormonal and feedback signals as dashed lines.

through these networks, parts of which collectively assign what has been called “incentive value” to a particular goal object. Neural pathways mediating the interactions between the object representation and motor networks are not fully understood, but sets of bidirectional connections between the hypothalamus and cortical structures such as the prefrontal cortex and hippocampus, together with subcortical regions such as the amygdala, septal nuclei, bed nuclei of the stria terminalis, and basal ganglia, are all likely to be critical for the integrative operations that designate and coordinate these aspects of motivated behaviors (Saper, 1985; Risold, Thompson, and Swanson, 1997; Swanson and Petrovich, 1998).

Motor Control

Figure 2 shows that motor control systems operate at three levels: a series of drive networks that set up and coordinate the motor events associated with specific behaviors; regions that are concerned with action selection; and the premotor/motor neuron networks that execute the actions. Figure 2 also shows a clear distinction between drive networks and object representation systems. This derives from the fact that object representation systems appear, at least at the systems level, to be behaviorally nonspecific, whereas drive networks are explicitly concerned with specific behaviors. For example, an animal uses the same parts of the telencephalon for spatial navigation whether it is looking for food, a mate, or shelter, whereas parts of the hypothalamus are concerned specifically with feeding or sexual behavior. However, with regard to other neural structures (for example, cell groups in the basal ganglia), this distinction may not be so clear-cut, and here it may prove difficult to determine to which system particular sets of neurons belong.

The concept of drive networks as defined here evolved from the idea of discretely localized hypothalamic satiety and hunger centers that was popular during the 1950s and 1960s. With elaboration of these pioneering studies over the ensuing 30 years, the idea of isolated centers has been replaced with a scheme whereby sets of more widely distributed but highly interconnected motor control networks direct the motor responses for particular behaviors. Each drive network contains sets of command circuits that stimulate, inhibit, or disinhibit a particular motor event (Figure 2). The exact nature of the expressed behavior, or whether it is expressed at all, is determined by the integrated output of these circuits.

Determining how the neural substrates of specific drive networks are distributed throughout the brain has proved quite difficult. Lesion and electrical stimulation experiments identified some time ago that the hypothalamus was a key structure for the expression of motivated behaviors. More recently, a wealth of neurochemical data has revealed that many neuropeptides have either stimulatory or inhibitory effects on particular motor functions. In turn, the fact that many of these neuropeptides are synthesized in hypothalamic neurons, which in some cases project quite widely throughout the brain, has focused attention on this forebrain region as a key locus of specific drive network components. Furthermore, a synthesis of results from lesion, microinjection, and neuroanatomical tracing studies has identified specific regions of the hypothalamus—particularly cell groups in the medial zone of the hypothalamus (Risold et al., 1997; Watts and Swanson, 2002)—as being principal components of individual drive networks.

One critical point that has emerged from these studies is that it is often difficult to place individual hypothalamic cell groups within specific command circuits (Figure 2). This is because traditionally defined cell groups such as the LHA, arcuate (ARH), or paraventricular (PVH) nuclei appear to contain elements that, in terms of function, belong to more than one type of command cir-

cuit; it seems unlikely that there is a tight “one cell group—one command circuit” relationship in the hypothalamus.

A well-documented example of a stimulatory circuit is the one activated by circulating A-II to stimulate drinking. A-II is detected by a central sensory transducer, the subfornical organ (SFO), which then directly and specifically stimulates water intake. The SFO provides efferents, most of which also contain A-II (Swanson, 1987), to a relatively limited set of structures, including parts of the prefrontal cortex, substantia innominata, medial preoptic area, bed nuclei of the stria terminalis, zona incerta, PVH, supraoptic nucleus, and LHA (Swanson, 1987), and presumably it is these regions that constitute part of the stimulatory circuit that initiates the motor aspects of drinking. Neuropeptide Y (NPY) neurons in the ARH contribute to a well-known example of a stimulatory eating mechanism; results from many studies show that NPY contributes to a circuit that directly stimulates food intake. In terms of inhibitory circuits, feeding again provides good examples of hypothalamic components. For example, α -MSH, a peptide synthesized in ARH neurons, provides an inhibitory signal to feeding by way of melanocortin 4 receptors expressed by LHA and PVH neurons (Elmquist et al., 1999).

Finally, interactions between different drive networks, particularly in the hypothalamus, are of paramount importance. For example, the effects of starvation are not limited just to increasing the drive to eat, they also reduce reproductive capacity. Similarly, dehydration leads to severe anorexia, as well as increasing the drive to drink (Watts, 2001). This cross-behavioral coordination is part of the mechanism that selects the drive with the highest priority, and most likely involves hormonal modulation acting together with the divergent neuroanatomic outputs from individual drive networks.

Those parts of the brain concerned with the planning, selection, and the moment-to-moment execution of particular motor actions include parts of the motor cortex, basal ganglia, midbrain, and hindbrain. Like the object representational systems, these regions at a systems level are generally behaviorally nonspecific. Although they express topography with regard to the mapping of particular motor actions, they do not seem to be organized in the behaviorally specific manner of the drive networks. To organize the appropriate behavior, regions controlling action selection must receive the integrated outputs of the drive networks. Although not well understood, complex sets of projections from the hypothalamus are most likely involved with this function.

Alpha-motor neurons in the ventral horn of the spinal cord control the striate musculature and hence the expression of all behavior. In turn, sets of premotor networks directly control oscillatory and the more complex patterns of motor neuron firing. Simple rhythmic movement patterns develop from an interaction between oscillatory rhythm generators, which directly involve the motor neurons, and networks of premotor CPGs located somewhat more distally in the spinal cord and hindbrain. A critical feature of these pattern generators is that they are capable of producing rhythmic output without sensory input. In turn, pattern generator output is modulated further by afferents from those parts of the appropriate command networks in the diencephalon and telencephalon. These often highly varied inputs provide the critical drive and contextual information that select the most appropriate motor program at any particular time.

Hormonal Modulation

Hormones have been known for many years to be critical modulators of motivated behaviors that influence a variety of neural structures at all brain levels (Figure 2). In this manner, because they are not encoding aspects of internal state, they are not feedback

signals, but act more as permissive factors. Steroid hormones, particularly gonadal steroids, are important signals of this type.

Feedback

Finally, feedback is a critical feature of behavioral motor control, and sensory signals encoding the magnitude and consequences of generated motor actions can control the length of a motivated behavioral episode. For example, postabsorptive humoral feedback (e.g., increasing CCK or decreasing plasma osmolality) and intero-sensory signals (e.g., gastric distension, oropharyngeal metering) lead to the termination of ingestive behaviors and subsequent behavioral refractoriness.

Discussion

Sophisticated neuroanatomical and molecular techniques are beginning to clarify the organization of those neural circuits that are responsible for controlling motivated behaviors. They emphasize that understanding how motivated behaviors are controlled requires the interaction of neural systems distributed throughout the brain that are both behaviorally specific and nonspecific. Although the structure of the different hypothalamic drive networks is reasonably well established, future work will need to clarify how each drive network interacts with others, with other forebrain systems concerned with complex sensory object representation, and with those hindbrain circuits concerned more directly with reflex actions and motor execution.

Road Map: Psychology

Related Reading: Conditioning; Emotional Circuits; Reinforcement Learning

References

- Bindra, D., 1978, How adaptive behaviour is produced: A perceptual-motivational alternative to response reinforcement, *Behav. Brain Sci.*, 1:41–91.
- Elmquist, J. K., Elias, C. F., and Saper, C. B., 1999, From lesions to leptin: Hypothalamic control of food intake and body weight, *Neuron*, 22:221–232.
- Grossman, S. P., 1979, The biology of motivation, *Annu. Rev. Psychol.*, 30:209–242. ♦
- Hinde, R. A., 1970, *Animal Behaviour: A Synthesis of Ethology and Comparative Psychology*, 2nd ed., New York: McGraw-Hill. ♦
- McFarland, D. J., and Sibly, R. M., 1975, The behavioural final common path, *Philos. Trans. R. Soc. Lond. B*, 270:265–293.
- Pfaff, D. W., 1982, Motivational concepts: Definitions and distinctions, in *The Physiological Mechanisms of Motivation* (D. W. Pfaff, Ed.), New York: Springer-Verlag, pp. 3–24. ♦
- Risold, P. Y., Thompson, R. H., and Swanson, L. W., 1997, The structural organization of connections between hypothalamus and cerebral cortex, *Brain Res. Rev.*, 24:197–254.
- Saper, C. B., 1985, Organization of cerebral cortical afferent systems in the rat: II. Hypothalamocortical projections, *J. Comp. Neurol.*, 237:21–46.
- Swanson, L. W., 1987, The hypothalamus, in *Handbook of Chemical Neuroanatomy*, vol 5. (A. Bjorklund, T. Hökfelt, and L. W. Swanson, Eds.), Amsterdam: Elsevier, pp. 1–124.
- Swanson, L. W., and Mogenson, G. J., 1981, Neural mechanisms for the functional coupling of autonomic, endocrine and somatomotor responses in adaptive behavior, *Brain Res.*, 228:1–34.
- Swanson, L. W., and Petrovich, G. D., 1998, What is the amygdala? *Trends Neurosci.*, 21:323–331.
- Toates, F., 1986, *Motivational Systems*, Cambridge, Engl.: Cambridge University Press. ♦
- Watts, A. G., 2001, Neuropeptides and the integration of motor responses to dehydration, *Annu. Rev. Neurosci.*, 24:357–384.
- Watts, A. G., and Swanson, L. W., 2002, Anatomy of motivational systems, in *Stevens' Handbook of Experimental Psychology*, 3rd ed. (C. R. Gallistel, Ed.), New York: Wiley, vol. 3, pp. 563–632. ♦

Motoneuron Recruitment

Daniel Bullock

Introduction

Motoneurons are neurons that directly innervate muscle fibers. When motoneuron discharges cause muscle fibers to contract, the resultant forces oppose static loads, and produce active accelerations and decelerations of limb segments. Moreover, co-contractions of opposing muscles allow us to stiffen joints and thereby maintain desired postures despite perturbations of unexpected magnitude and direction. Because of the direct anatomical link between motoneurons and contractile fibers, there is a close relationship between motoneuron activity and force production.

A motoneuron together with the contractile fibers that it innervates constitutes a *motor unit*. The range of forces producible by one motor unit is small. To make it possible to generate large forces, motor units must be combined into larger aggregates, and the results of such aggregation are the muscles. Immediately associated with each muscle is a population or pool of motoneurons. Muscles are therefore composite structures, and their force-generating components, the motor units, are typically heterogeneous. For example, muscle fibers differ systematically in fatigability and the associated motoneurons differ systematically in their size. How are these heterogeneous aggregates of force-generating elements recruited in the service of reflexes, voluntary movement,

and posture? Such task-dependent recruitment is achieved by a combination of motor unit and neural network specializations.

Consider the simple question of control of force magnitude. If any excitatory input were sufficient to cause simultaneous excitation of all motor units, then the minimum force produced by the aggregate would be much too large for most purposes. To produce accurate movements, forces must be finely graded in response to the input to the motoneuron pool. The fine grading of forces required for accuracy favors a design that allows both partial activation of the motoneuron/fiber pool and finely graded changes, up or down, from preexisting states of activation.

Such force grading by a cells/fibers aggregate provides a functional context for understanding the *size principle* of motoneuron recruitment proposed in 1965 by Henneman, Somjen, and Carpenter (see Burke, 1998, for a review). The size principle encompasses many aspects of the design of motoneuron pools and their embedding within the sensorimotor system. In this design, an excitatory input often reaches all elements of the motoneuron pool at the same time. However, elements of the motoneuron pool differ in their activation thresholds. Because there is a distribution of threshold values from small to large, the larger the excitatory input to the pool, the more elements become active. This enables a continuously varying input signal to produce a graded force response from

the muscle. As the excitatory input to the pool grows, motoneurons are recruited in order by size from smallest to largest, because motoneurons with larger somatic volumes also have higher thresholds. As excitatory input declines, or inhibitory input increases, motoneurons are derecruited in order by size, from largest to smallest.

The grading of force by recruitment, which is necessarily quantal, is supplemented by finer grading through firing rate modulation of individual cells, because each cell's firing rate is sensitive to input fluctuations in its suprathreshold range. This design affords finely graded increments and decrements in force over the entire range of muscle force capability.

It might appear that the size principle serves to make each spino-muscular force generator a fixed-gain, near-linear, amplifier of excitatory inputs. However, many factors complicate the situation. First, the gain is not fixed because muscle force can become decoupled from motoneuron pool activation if a contraction-opposing load causes muscle yielding, or if the muscle fatigues. Second, the amplification function is often faster-than-linear because motoneurons with larger cell bodies, and thus higher recruitment thresholds, typically project by larger, faster-conducting axons to more muscle fibers, each of which exhibits shorter twitch contraction times. Third, twitch contractions of muscle fibers are slow relative to rapid fluctuations of excitatory inputs to motoneurons. Fourth, muscle obeys a *force-velocity law*: force output from a muscle decreases as its shortening velocity increases. Fifth, the conventional delimitation of a motor unit, although minimal, is somewhat arbitrary. Several other closely linked neural and sensory constituents appear in most mammalian muscle control systems as part of the apparatus for force generation (cf. Burke, 1998). For example, before exiting the spinal cord, the axons of most alpha-motoneurons give off collaterals that excite Renshaw cells (RCs), which inhibit those same alpha-motoneurons. Sixth, the net torque developed at a joint depends on both mechanical advantage and the balance of forces created by groups of muscles arranged into synergistically antagonistic sets. Each of these considerations reveals a need for network control of recruitment, to ensure that opponent muscle sets generate the right force balances through time.

Compensations for Fatigue and Yielding

Muscle fatigue and yielding make the functional relation between pool activation and force inherently variable, and network interactions provide compensations that reduce the variability in this linkage. Nichols and Houk (1973) argued that two feedbacks from muscle receptors to spinal motoneuron pools cooperate to reduce variability in *muscle stiffness*, the ratio of muscle force changes to muscle length changes. Muscle yielding events reduce stiffness while also increasing the activity of stretch-sensitive receptors, the spindles, and decreasing the activity of tension-sensitive receptors, the Golgi tendon organs (GTOs). Because spindle feedback directly excites alpha-motoneurons via type Ia sensory fibers, whereas GTOs can inhibit motoneurons via Ib interneurons, both feedbacks are compensatory. It is often noted that GTO feedback also has appropriate characteristics to compensate for muscle fatigue. Bullock and Grossberg (1989) argued that the covariation of motor unit sizes and contraction rates is also compensatory for yielding.

Linearization or Equalization of Pool Responses

By itself, the covariation of recruitment threshold, number of fibers contacted, and fiber contraction rates with motoneuron size can produce a faster-than-linear relationship between excitatory input to the motoneuron pool and the force output of the muscle, at least under isometric conditions when the system is not approaching saturation. Akazawa and Kato (1990) and Bullock and Grossberg

(1989) independently proposed that Renshaw feedback improves this transduction. The Akazawa and Kato analysis (1990) treated a single motor unit pool, and showed that inhibitory Renshaw feedback may be able to linearize the relationship between excitatory inputs and force outputs. Bullock and Grossberg (1989) sought to explain how spinal circuitry enabled the higher brain to achieve independent control of joint angle and joint stiffness. Accordingly, these authors analyzed the "FLETE" circuit (Figure 1), which encompassed a lumped pair of motor unit pools associated with biomechanically opposed muscles. By Factoring the Length and Tension properties of muscle, the FLETE network allows a descending co-contraction signal to stiffen and thereby stabilize the joint at any desired angle. Available data (Humphrey and Reed, 1983) indicate that voluntary stiffness adjustments are achieved by varying an excitatory signal relayed to both opponent motoneuron pools. Bullock and Grossberg (1989) showed that in the absence of Renshaw feedback, a descending co-contraction signal would generally be unequally amplified by recruitment events within opposing motoneuron pools. Such unequal amplification would lead to an undesired joint rotation as well as to a change in joint stiffness. They then showed that Renshaw-mediated feedback could help guarantee independent control of joint stiffness and joint angle by equalizing the two pools' amplifications of the co-contraction signal. This equalization, which need not involve global linearization of recruitment, is achieved by a local circuit that incorporates mutual inhibition between opponent Renshaw pools and between Ia reciprocal inhibitory interneurons, which, like alpha-motoneurons, are inhibited by Renshaw cells (RCs).

This view of the role of RCs is consistent with data that contradict alternative views. Pratt and Jordan (1987) showed that RCs fired in phase with alpha-motoneurons during fictive locomotion, but that they were not needed for generation of the locomotor cycle. This disconfirmed the hypothesis that they were an integral part of the spinal locomotor generator. Lindsay and Binder (1991) observed that although steady-state Renshaw inhibition caused similar synaptic currents in alpha-motoneurons of different sizes, IPSP amplitudes did correlate with cell size. They concluded that "the

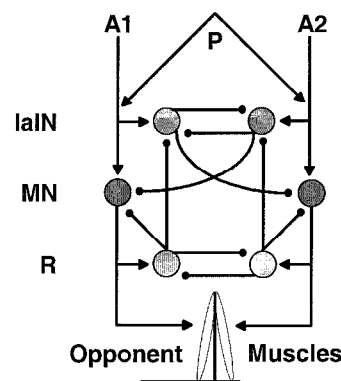


Figure 1. Partial connectivity of the FLETE model for independent control of joint angle and joint stiffness. To set desired joint angle, the higher brain reciprocally adjusts descending signals A1 and A2 directed to two opposing alpha motoneuron (MN) pools that project to opposing muscles. Descending signal P to both motoneuron pools adjusts joint stiffness without modifying joint angle if increments in P lead to equal increments in the force outputs of the two opposing muscles. Renshaw (R) cell feedbacks, among others, compensate for nonlinearities in the motoneuron response function and thereby help assure equal force increments in the two muscles affected by P. Renshaw feedback disinhibits opponent MNs via the Ia interneurons (IaIN). Arrow and dot line-endings, respectively, indicate excitatory and inhibitory synapses.

biggest impact of [RC] inhibition will be on the force output of motoneurons firing on the steep part of their force-frequency curve" (p. 176).

A subsequent extension of the FLETE model showed that the *triphasic* EMG bursts characteristic of rapid self-terminated joint rotations *emerge* within an arm-controlling network activated by *monophasic* descending control signals, if the network incorporates velocity-sensitive muscle spindles. Contreras-Vidal, Grossberg, and Bullock (1997) showed that the FLETE model is applicable to multi-joint arm movement control using both mono- and bi-articular muscles, and that the independent control property is enhanced by the incorporation of sensory feedbacks from spindle (Ia), GTO, and joint receptors. Moreover, van Heijst, Vos, and Bullock (1998) showed that connection weights consistent with the independent-control property will self-organize in the circuit of Figure 1 if local synapses are adjusted by a Hebbian learning process while the circuit is stimulated by a rhythmic input. Their developmental simulation modeled how such spinal circuits self-tune during prenatal episodes of rhythmic activity in avian and mammalian embryos.

Adaptive Central Control of Motoneuron Gain

Renshaw cells also mediate descending modulation of the motoneuron recruitment process. Stimulation in nucleus interpositus (NIP) of the cerebellum, or in its target, the Red Nucleus (RN), which projects to spinal pools via the rubrospinal pathway, enhances the gain of the monosynaptic stretch reflex by inhibiting RCs, thereby releasing alpha-motoneurons from recurrent inhibition. The NIP or RN stimulation also excites motoneurons. Bullock and Grossberg (1989) proposed that the implied bivalent rubral projection to RCs and alpha-motoneurons afforded adaptive, i.e., learning-based, control of the "gain" of movement commands directed to motoneuron pools. Contreras-Vidal et al. (1997) introduced a neural network comprising a central trajectory generator, an extended FLETE model, and a model cerebellar network capable of learning to modulate motoneuron recruitment via a bivalent output to RCs and alpha-motoneurons. Simulations of the circuit (Figure 2) showed that if the cerebellum received both a desired velocity signal and an error feedback routed from spindles to cerebellum via the inferior olive, then a learning-adjusted cerebellar output substantially enhanced the dynamic tracking characteristics of the limb by transiently exciting, and removing inhibition from, the agonist motoneuron pool (Figure 1). This model is consistent with recent biophysics-based models of cerebellar adaptive timing (e.g., Fiala, Grossberg, and Bullock, 1996), and with common observations of phasic RN and interpositus activity during learned movements. A closely related modeling treatment, encompassing cerebellar modulation of the Figure X circuit in the context of realistic sensory lags, has recently appeared (Spoelstra, Schweighofer, and Arbib, 2000).

Roles of Motor Cortex in Motoneuron Recruitment

Many cells in the primary motor cortex (M1) of primates excite motoneurons via mono- or short polysynaptic pathways, and the pathway for the long-loop stretch reflex traverses M1. Moreover, cooling of the dentate nucleus of the cerebellum, which affects M1 via the thalamus, eliminates anticipatory, force-related, components of normal M1 activity. Many studies have strongly implicated M1 in load compensation achieved by direct recruitment of motoneurons, although a subset of M1 cells are relatively load insensitive (Kalaska et al., 1989). Yet other studies have appeared to implicate M1 in a high-level representation of the direction of movement in Cartesian space. Recently, two models have begun

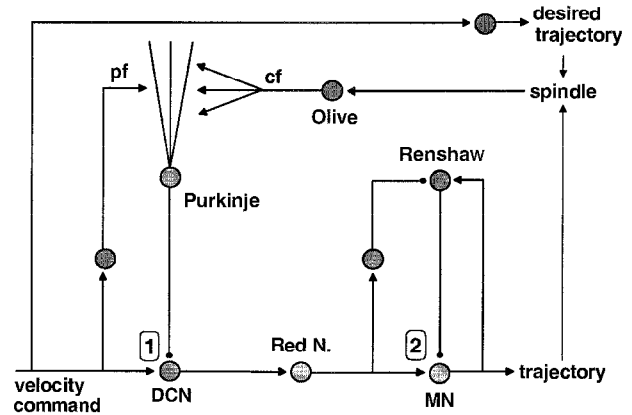


Figure 2. Network model incorporating two sites for controlling motoneuron excitation by release from inhibition. Prior to learning, a velocity control signal directed toward a muscle via the deep cerebellar nuclear (DCN) pathway will have a negligible effect due to Purkinje (P) cell inhibition of DCN sites and Renshaw cell inhibition of alpha-motoneurons (MN). However, trajectory errors detected by muscle spindles activate the inferior olive, whose climbing fibers (cf) reach the dendrites of Purkinje cells. Climbing fiber activity causes long term depression of coactive parallel fiber (pf) synapses that excite Purkinje cells. Depression of Purkinje excitation causes disinhibition of DCN sites. This "opens the gate" for the velocity control signal to activate the Red Nucleus. The Red Nucleus both excites alpha-motoneurons and inhibits Renshaw cells.

to address the dilemma posed by these observations. The extended Vector Integration To Endpoint (VITE) model of Bullock, Cisek, and Grossberg (1998) proposed a circuit involving 6 electrophysiologically identified cell types in M1 and parietal area 5 to explain the distinct computational roles of load-sensitive and load-insensitive cells in both arm trajectory generation *and* load compensation. This model's relatively load-insensitive cells have polysynaptic links to alpha-motoneurons, whereas the most load-sensitive cells have monosynaptic links. Todorov (2000) proposed a model (pertinent primarily to load-sensitive cells) based on the assumption that M1 recruitment compensates for the negative effects of the force-velocity law on the ability of muscle to sustain force when shortening at a significant velocity.

If some M1 cells directly control motoneuron recruitment, and thus force generation, then theories of sensorimotor transformations (e.g., Barreca and Guenther, 2001) predict that the preferred spatial directions of such M1 cells must be strongly posture dependent—and they are. Several recent simulations based on this premise have succeeded in predicting posture- and trajectory-dependent tuning properties of M1 cells and the muscles to which they project (Ajemian, Bullock, and Grossberg, 2001; Scott and Kalaska, 1997).

Discussion

Neural network analyses have begun to clarify how local spinal circuits cooperate with central adaptive circuits for task-dependent control of motoneuron recruitment, but many basic questions remain to be addressed. Too little is known about the pathways for descending control of gamma- versus alpha-motoneurons. Also, the behavioral functions of many known aspects of the recruitment system, such as motoneuronal plateau potentials, remain to be elucidated by computational analyses. Models must also be elaborated to accommodate the unique connectivities that govern recruitment in different species, which differ dramatically in biomechanical, behavioral, and neuronal specializations.

Road Map: Mammalian Motor Control

Related Reading: Cerebellum and Motor Control; Equilibrium Point Hypothesis; Limb Geometry, Neural Control; Muscle Models; Vestibulo-Ocular Reflex

References

- Ajemian, R., Bullock, D., and Grossberg, S., 2001, A model of movement coordinates in motor cortex: Posture-dependent changes in the gain and direction of single cell tuning curves, *Cerebral Cortex*, 11:1124–1135.
- Akazawa, K., and Kato, K., 1990, Neural network for control of muscle force based on the size principle of motor unit, *Proc. IEEE*, 78:1531–1535.
- Barreca, D. M., and Guenther, F. H., 2001, A modeling study of potential sources of curvature in human reaching movements, *J. Motor Behav.*, 33:387–400.
- Bullock, D., Cisek, P. E., and Grossberg, S., 1998, Cortical networks for control of voluntary arm movements under variable force conditions, *Cerebral Cortex*, 8:48–62.
- Bullock, D., and Grossberg, S., 1989, VITE and FLETE: Neural modules for trajectory formation and postural control, in *Volitional Action* (W.A. Hershberger, Ed.), Amsterdam: North-Holland/Elsevier, pp. 253–298. ♦
- Burke, R. E., 1998, Spinal cord: Ventral horn, in *The Synaptic Organization of the Brain* (G.M. Shepherd, Ed.), New York: Oxford, pp. 77–120. ♦
- Contreras-Vidal, J. L., Grossberg, S., and Bullock, D., 1997, A neural model of cerebellar learning for arm movement control: Cortico-spino-cerebellar dynamics, *Learning and Memory*, 3:475–502.
- Fiala, J. C., Grossberg, S., and Bullock, D., 1996, Metabotropic glutamate receptor activation in cerebellar Purkinje cells as substrate for adaptive timing of the classically conditioned eye blink response, *J. Neurosci.*, 16:3760–3774.
- Humphrey, D. R., and Reed, D. J., 1983, Separate cortical systems for control of joint movement and joint stiffness: Reciprocal activation and coactivation of antagonist muscles, *Adv. Neurol.*, 39:347–372. ♦
- Kalaska, J. F., Cohen, D. A. D., Hyde, M. L., and Prud'homme, M. J., 1989, A comparison of movement direction-related versus load direction-related activity in primate motor cortex, using a two dimensional reaching task, *J. Neurosci.*, 9:2080–2102.
- Lindsay, A. D., and Binder, M. D., 1991, Distribution of effective synaptic currents underlying recurrent inhibition in cat triceps surae motoneurons, *J. Neurophysiol.*, 65:168–177.
- Nichols, T. R., and Houk, J. C., 1973, Reflex compensation for variations in the mechanical properties of a muscle, *Science*, 181:182–184.
- Pratt, C. A., and Jordan, L. M., 1987, Ia inhibitory interneurons and Renshaw cells as contributors to the spinal mechanisms of fictive locomotion, *J. Neurophysiol.*, 57:56–71.
- Scott, S. H., and Kalaska, J. F., 1997, Reaching movements with similar hand paths but different arm orientations. I. Activity of individual cells in motor cortex, *J. Neurophysiol.*, 77:826–852.
- Spoelstra, J., Schweighofer, N., and Arbib, M. A., 2000, Cerebellar learning of accurate predictive control for fast reaching movements, *Biol. Cybernetics*, 82:321–333.
- Todorov, E., 2000, Direct cortical control of muscle activation in voluntary arm movements: A model, *Nature Neurosci.*, 3:391–398.
- van Heijst, J. J., Vos, J. E., and Bullock, D., 1998, Development in a biologically inspired spinal neural network for movement control, *Neural Networks*, 11:1305–1316.

Motor Control, Biological and Theoretical

R. Christopher Miall

Introduction

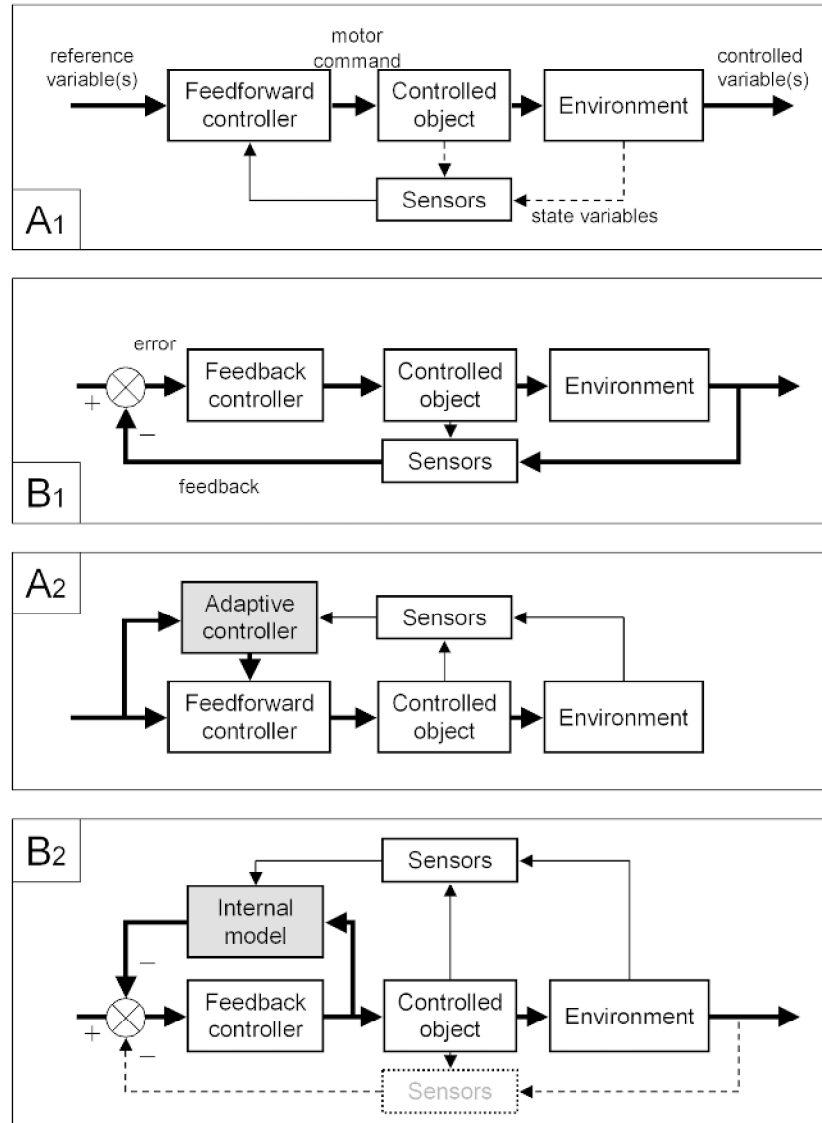
Biological motor control can be characterized as a problem of controlling nonlinear, unreliable systems whose states are monitored with slow and sometimes low-quality sensors. In response to changing sensory inputs, internal goals, or motor errors, the motor system must solve several basic problems: selection of an appropriate action and transformation of control signals from sensory to motor coordinate frameworks; coordination of the selected movement with other ongoing behaviors and with postural reflexes; and monitoring the movement to ensure its accuracy. These stages may be interlinked, so that separation of any one particular problem into these individual stages may not be possible. This article describes some of the ways we think that biological motor systems solve these tasks, based on principles (and terminology) whose origins are in engineering and cybernetics. The field of cybernetics has developed from Norbert Wiener's initial ideas on communication and control theory in complex mechanical and biological systems, which focused on feedback mechanisms.

A motor control system acts by sending motor commands to a controlled object, often called the “plant,” which in turn acts on its local environment (Figure 1). The plant or the environment has one or more variables that the motor system attempts to regulate, either to maintain them at a steady reference level in the face of disturbances (a “regulator”) or to follow some changing reference value (a “controller”). The motor control system may make use of sensory signals from the environment, from its reference inputs, and from the plant to determine what actions are required. Sensory inputs from the plant can provide information about the *state* of

the controlled object. Here, the state can be considered as all relevant variables that adequately describe the controlled object. But note that the sensory inputs to the controller do not necessarily provide direct measures of the true state of the system: they may be inaccurate or delayed, as discussed later. If controller output is based on signals that are unaffected by the plant output, it is said to be a *feedforward controller*: the feedforward control path is the thick line from left to right in Figure 1A₁, which requires no return signals. If the controller output is instead based on a comparison between the reference and the controlled variables, it is a *feedback controller* (Figure 1B₁): the control pathway is a closed loop. One can add more complex control strategies to these simple systems (Figure 1A₂, 1B₂), as described in more detail below.

The advantage of feedforward control is that it can, in the ideal case, give perfect performance with no error between the reference and the controlled variable. The main disadvantages for biological systems include the potential difficulty in generating an accurate controller for a complex system and the lack of error corrections. If the controller is not accurate, if the plant is unreliable, or if unexpected external disturbances occur, output errors go unchecked. Since no biological system can be both perfectly accurate and perfectly free of external disturbances, error correction is usually necessary. In contrast, the major advantage of negative feedback control lies in its very simple, robust strategy. The controller drives the plant so as to cancel the feedback error signaled by the comparator. Because it constantly seeks to cancel the error, it operates well, even without exact knowledge of the controlled object and despite internal or external disturbances. But feedback control strategies also have disadvantages: errors cannot be avoided but

Figure 1. *Feedforward control:* A_1 , The black arrows represent the on-line control signals; the dotted lines are off-line signals used to update controller. A_2 , Adaptive controllers using off-line information can adjust parameters of the feedforward controller to reflect changes in the plant properties. *Negative feedback control:* B_1 , The black circle represents a comparison between the reference and feedback signals. The black arrows now form a closed control loop; the crossed circle is a comparator. B_2 , An internal model of the controlled object can replace the external feedback loop with a rapid feedback estimate.



must occur and be corrected, and feedback control—especially in biological systems—tends to be slow.

Feedback Control

The design criteria for negative feedback control are dominated by the closed-loop gain. Gain is defined as the ratio of a system's output to its input. For a linear servo controller, the gain should be close to unity, so that a given input (the reference value) evokes an output of almost equal magnitude. In a feedback circuit (Figure 1B₁), one can define both open-loop and closed-loop gains. The open-loop gain K_o is given by the ratio of the response to the error; it gives the response expected if the feedback path shown in Figure 1B₁ is cut, thus opening the loop. The closed-loop gain K_c is given by the ratio of response to reference amplitudes. The closed-loop gain K_c is determined by the open-loop gain where $K_c = K_o / (1 + K_o)$. For ideal control, K_c should be unity under all conditions; thus the open-loop gain K_o should be as high as possible, ensuring that K_c approaches unity. In practice, K_o is usually frequency dependent and can never reach infinity; hence K_c is also frequency dependent and less than unity.

The design of nonlinear and multidimensional feedback systems is beyond the scope of this article, except to note that, in many instances, complex control problems can be simplified and linearized around the current state of the system. This may be particularly true of biological systems, in which control is often only approximate.

Notice that the comparison of the reference value with the controlled variable to give an error signal (Figure 1B₁) is affected by the dynamics of the motor control and sensory systems. When a command is issued by the controller, its effects are not immediately apparent to the comparator, but are delayed by the plant and sensor dynamics and by transport delays on both the forward and feedback paths. In biological systems, where sensor delays are inevitable, the comparison is always out of date. Hence in any feedback system there will be a frequency at which these delays combine to impart a 180° phase lag. The open-loop gain K_o at that frequency now only needs to be unity (instead of very large) to make $K_c > 1.0$, forcing the system into instability. Any small error or disturbance will be overcorrected and result in even bigger errors, leading to yet bigger corrections. Human examples of instability are indeed seen when control delays are artificially increased in man-machine interfaces

(Miall and Wolpert, 1996) or as a result of increased neural transport delays in neuropathies such as multiple sclerosis.

Physiological Feedback Circuits

Although feedback control circuits are found throughout physiology, let us consider just two examples from vertebrate motor systems. The major tension-producing fibers of the vertebrate muscle, known as extrafusal fibers, contract following excitation by alpha motor neurons. However, the amount of tension produced by the muscle in response to a motor command varies with the length of the muscle, its speed of contraction, level of fatigue, and so on. The muscles are therefore provided with numerous sensory structures, muscle spindles, that signal back to the CNS the length and rate of stretch of the muscle. Spindles are complex sensorimotor structures combining contractile elements (intrafusal fibers, excited by specialized gamma motor neurons) with a central stretch-sensitive region. Their axons project onto alpha motor neurons in the spinal cord that serve the same muscle and synergistic muscles. This circuit (the stretch reflex) is a feedback controller for muscle length. If the muscle is stretched, the spindles respond, exciting the alpha motor neurons, and the resulting reflex contraction of the extrafusal fibers restores the muscle to its original length, silencing the spindles again. Thus the spindles signal a deviation from their regulated length, and the controller (the alpha motor neuron) acts to cancel the error. The muscles also contain Golgi tendon organs (GTOs), which are attached to the tendons of muscle and respond to increased tension in the tendon. They excite interneurons that inhibit motor neurons of that muscle and other muscles acting around the same joint, and also act in a feedback manner. If muscle tension increases due to an external load, for example, the GTOs are activated and, via the inhibitory interneuron, inhibit the motor neurons, causing the muscle to relax. This reduces tension, and thus the negative feedback loop serves to maintain a controlled level of tension. This description of the spindle and GTO is oversimplified, ignoring aspects such as control of muscle stretch velocity, but emphasizes their basic control properties. Together, they act to maintain a muscle in its current state: changes in length or in tension will be automatically opposed.

Feedforward Control

Feedforward control schemes may be grouped as those based on direct control and those based on indirect control using internal models. Here, direct control means control without *explicit* knowledge of the behavior of the plant (see REINFORCEMENT LEARNING IN MOTOR CONTROL). In practice, a controller that can store and issue appropriate motor programs must have implicitly, if not explicitly, captured knowledge of the plant. Hence feedforward controllers must be matched to the properties of the plant they control. As a physiological example, the equilibrium point hypothesis (see EQUILIBRIUM POINT HYPOTHESIS) makes use of the spring-like properties of muscles. For any set of springs pulling across the multiple joints of a limb, there will be a stable position into which the limb passively settle. Thus, the CNS could define the “end-point” muscle tensions and the limb would move to the desired position without the controller’s knowing either its starting position or its behavior during the movement. An alternative direct scheme is to generate the appropriate commands—a temporal sequence of required changes in muscle force, acquired and stored as a motor program—but again without any explicit knowledge of the plant. In the limit one could use a memorized lookup table to store appropriate motor commands for each input-output pair. However, the memory demands grow explosively if a motor command is stored for every possible pairing. Some form of generalization is assumed to avoid this problem (see SENSORIMOTOR LEARNING)

such that a coarse-grained representation is achieved, with interpolation.

Physiological Feedforward Control

Muscle spindles and GTOs are used to ensure that actions occur as planned. By sending motor commands both to the alpha and to the gamma motor neurons, both the force-producing extrafusal fibers of the muscle and the much weaker intrafusal fibers of the spindle co-contract. If the joint fails to move fast enough owing to an unexpected load, the spindle contractile elements shorten within the main muscle, the stretch-sensitive sensory region is stimulated, and additional excitatory drive is reflexively added to the spinal alpha motor neurons to overcome the load. The original position control theory proposed by Merton has had to be supplemented by tension and velocity control; but this simple description, while incomplete, highlights the main principles. Note that by co-activating alpha and gamma motor neurons, the reference values of the feedback circuit described earlier are predictively modified. Thus for the supraspinal centers driving the movement, the spinal circuits can be treated as a feedforward controller, autonomously regulating the muscles without the need for feedback to these higher centers. Of course, if errors become large, cortical control can be invoked. This demonstrates an important principle: biological motor circuits are often hierarchical, with lower levels regularizing the behavior of the controlled object and higher systems providing increasingly indirect control (Loeb, Brown, and Cheng, 1999).

Another example of feedforward control is found in the oculomotor system (see COLLICULAR VISUOMOTOR TRANSFORMATIONS FOR GAZE CONTROL). Human eye muscles have muscle spindles, but they do not seem to have a functional stretch reflex: passive movements of the eyes are not reflexively adjusted, and even seem to be ignored. As Helmholtz noted, if one pushes on the side of one’s own eye, the resulting retinal movement is reported by the visual system as movement of the external world. The reason the oculomotor system may be able to operate in feedforward mode is that the mechanical load (the spherical eyeball) is relatively constant, unaffected by external weights or gravity, and is therefore more easily controlled than a multi-jointed limb. Functionally, of course, there is powerful *visual* feedback: if the eyes drift from the target of gaze, the error is reported as slip of the visual image over the retina. Retinal slip drives “on-line” corrective velocity adjustment during smooth eye movement. Because saccades are of short duration, errors are corrected “off-line” with a secondary saccade. Consistent saccadic under- or overshooting errors lead to long-term changes in the feedforward controller, an example of adaptive control.

Adaptive Control and Internal Models

Adaptive Control

Adaptive control relies on monitoring performance over a longer time scale than that used by negative feedback control to generate a measure of average performance rather than of moment-to-moment error. The adaptive controller is then used to adjust the motor responses, for example, by modulating the feedforward controller as indicated in Figure 1A₂ or by modulating the open-loop gain of a feedback controller. The advantage of adaptive control is that it can compensate for gradual changes in the motor performance of the controlled object. Controllers can also be designed to track predictable changes in the reference value. Because the performance of physiological systems (as well as the goals of behavior) changes over time, all biological control systems are to some extent adaptive through mechanisms as diverse as evolutionary change, growth, or learning and memory. In control of eye movements, there is good evidence that the cerebellum is involved in adaptation (Robinson and Fuchs, 2001).

Internal Models

Two forms of internal model can be distinguished. An ideal feedforward controller will ensure that the plant output (the controlled variable) is always identical to the reference value. Thus it inputs the reference value (and often also the state signals, Figure 1A₁) and outputs a motor command; the motor command shifts the plant into a new state, which should equal the reference value. Thus one can describe the ideal feedforward controller as an *inverse* of the plant: the plant translates commands into states whereas the inverse controller translates desired states into commands. If the transfer function of the plant is represented as P , its inverse is P^{-1} , and the transfer function of the complete system (from reference value to controlled variable) is $P \cdot P^{-1} = 1$. Again, this implies that the perfect system has a gain of unity. Inverse modeling is covered in more detail in Jordan (1994).

The alternative type of internal model is known as a forward model of the plant (Figure 1B₂). Its inputs are a copy of the motor command being sent to the plant and also the current feedback of the plant state, and its output is an estimate of the next state of the plant or of the controlled variables. This estimate is available to the feedback controller more rapidly than actual feedback. Thus, the external feedback loop can be replaced by an internal loop, which avoids the feedback delays mentioned above. A negative feedback loop with negligible delay and a high open-loop gain will rapidly and accurately drive its plant in a direction to minimize the comparator error. Thus, a fast internal loop including a forward model is functionally equivalent to an inverse dynamic model. Of course, viewed from outside the loop, it functions as a feedforward controller: it disregards the actual feedback and hence is no longer error correcting. The oculomotor feedforward controller may be an inverse model like that shown in Figure 1A₂ (Krauzlis and Lisberger, 1989); an alternative proposal suggests an internal forward model as in Figure 1B₂ (Robinson, 1975).

Schemes that combine feedback with feedforward control (Hoff and Arbib, 1992; Miall et al., 1993) depend on estimation of the expected feedback signal, including its delay. Recent theories have proposed combined forward and inverse models, working in pairs for system identification, control, and adaptation (Wolpert, Miall, and Kawato, 1998).

Physiological Internal Models

Visual guidance of the human arm is based on sensory information from the visual system with processing delays of up to 100 ms. Motor commands issued by the CNS may take 50 ms to initiate muscle contraction, and these changes are signaled by vision and by proprioceptors with delays of perhaps 50 to 100 ms. So feedback signals from the environment will lag significantly behind the issue of each motor command. Despite this, we control our limbs skillfully and accurately with movement durations of well under half a second. Thus, our motor control cannot be based entirely on feedback signals; we also employ feedforward control. It is likely (although not yet certain) that control is based on internal representations of the motor system—internal models (Miall and Wolpert, 1996).

Can we identify these internal models in the brain? The cerebellum is a strong contender for internal model representations (Ito, 1984; Wolpert et al., 1998). The model should receive as inputs either the motor goal or an efferent copy of the motor command, and also receive proprioceptive information about the current state of the body. There must be a mechanism to allow the model to adapt to predict accurately the behavior of the limb, i.e., a neural learning mechanism. And the output of the model must form either the motor command or a sensory prediction of the action outcome. The cerebellum can satisfy all these constraints, but this alone is not proof. Other possible sites are the motor cortex, parietal cortex,

and the spinal cord, although a spinal representation would probably be more closely related to individual muscles than a model of the whole arm.

There are strong connections from the motor cortical areas and posterior parietal cortex to the lateral hemispheres of the cerebellum, and from there, ascending paths back to premotor and motor cortices or descending to brainstem nuclei. Spinocerebellar tracts provide a large array of proprioceptive signals, updating the cerebellum on the current state of the limb. For adaptation, we know that coincident activity in climbing fiber and parallel fiber inputs to Purkinje cells results in a sustained change in the strength of the parallel fiber-Purkinje cell synapse (see CEREbellum: NEURAL PLASTICITY). Some researchers therefore suspect that the cerebellum acts as an adaptive inverse model on the feedforward control pathway (Figure 1A₂; Ito, 1984; Kawato and Gomi, 1992). Ito viewed the cerebellum as an adaptive side path to the descending systems, modulating the feedforward commands issued by cerebral control centers. Kawato views it as an alternative to these cerebral systems, replacing their control function. The alternative forward model-based scheme (Figure 1B₂) is also valid; hence the cerebellum may represent an adaptive forward model on a feedback pathway (Miall et al., 1993). This Smith predictor theory places the forward model within the closed cerebrocerebellar loop as the controller and incorporates feedback via an adaptive delay module. Each module is learned independently, with different time courses. It is difficult to distinguish between inverse dynamics models (Figure 1A₂) and internal feedback loops containing forward models (Figure 1B₂) unless one can get access to their internal structure. One might block the internal feedback loop needed for a forward model or decoding the input and output signals. However, many recent psychophysical, electrophysiological, and functional imaging experiments have generated strong evidence of the use of internal models for motor control, for state estimation, and for planning and interpretation of actions.

Road Maps: Mammalian Motor Control; Robotics and Control Theory

Related Reading: Action Monitoring and Forward Control of Movements; Cerebellum and Motor Control; Optimization Principles in Motor Control; Sensorimotor Learning

References

- Hoff, B., and Arbib, M. A., 1992, A model of the effects of speed, accuracy and perturbation on visually guided reaching, in *Control of Arm Movement in Space: Neurophysiological and Computational Approaches*, vol. 22, *Experimental Brain Research Series* (R. Caminiti, Ed.), Berlin: Springer-Verlag, pp. 285–306.
- Ito, M., 1984, *The Cerebellum and Neural Control*, New York: Raven Press.
- Jordan, M., 1994, Computational aspects of motor control and motor learning, in *Handbook of Motor Control* (H. Heuer and S. Keele, Eds.), Berlin: Springer-Verlag, pp. 1–65.
- Kawato, M., and Gomi, H., 1992, The cerebellum and VOR/OKR learning models, *Trends Neurosci.*, 15:445–453.
- Krauzlis, R. J., and Lisberger, S. G., 1989, A control systems model of smooth pursuit eye movements with realistic emergent properties, *Neural Computation*, 1:116–122.
- Loeb, G. E., Brown, I. E., and Cheng, E. J., 1999, A hierarchical foundation for models of sensorimotor control, *Exp. Brain Res.*, 126:1–18.
- Miall, R. C., and Wolpert, D. M., 1996, Forward models for physiological motor control, *Neural Networks*, 9:1265–1279. ♦
- Miall, R. C., Weir, D. J., Wolpert, D. M., and Stein, J. F., 1993, Is the cerebellum a Smith predictor? *J. Motor Behav.*, 25:203–216.
- Robinson, D. A., 1975, Oculomotor control signals, in *Basic Mechanisms of Ocular Motility and Their Clinical Implications* (G. Lennerstrand and P. Bach-y-Rita, Eds.), Oxford: Pergamon Press, pp. 337–374.
- Robinson, F. R., and Fuchs, A. F., 2001, The role of the cerebellum in voluntary eye movements, *Annu. Rev. Neurosci.*, 24:981–1004.
- Wolpert, D. M., Miall, R. C., and Kawato, M., 1998, Internal models in the cerebellum, *Trends Cogn. Sci.*, 2:338–347.

Motor Cortex: Coding and Decoding of Directional Operations

Bagrat Amirikian and Apostolos P. Georgopoulos

Introduction

Two fundamental issues—how does the brain work, and how can we build intelligent machines?—are the leitmotifs of this *Handbook*. There are many strategies for attacking these questions, depending on what particular aspects of these broad issues we are interested in. One approach lies in the behavioral-neurophysiological domain. The recording of the activity of single cells in the brain of behaving animals provides a tool for directly studying how a particular behavioral pattern is represented and generated. Studies along this line intend to answer the first question: How does the brain work?

In the framework of this approach, the firing of a single neuron or a population of neurons can be correlated with one or several behavioral variables changing in time. The main challenge is to solve a pair of complementary problems: the coding/specification problem and the decoding/implementation problem. The former addresses the question of how the information about a particular behavioral variable is encoded in the neuronal activity being produced (see POPULATION CODES). The latter concerns the neural mechanisms by which encoded variables generate a behavioral pattern unfolding in time.

In the behavioral-neurophysiological domain, the constructive framework for attacking the issue of building “intelligent” machines could be formulated in the context of the decoding problem, namely: How can we design adaptive systems that would transform neuronal signals recorded in the brain of behaving animals into the physiologically appropriate behavioral pattern generated by an artificial machine? (see BRAIN-COMPUTER INTERFACES).

The work reported here summarizes a series of studies based on experimental work and abstract modeling. It exemplifies the successful application of the above-mentioned paradigms to the study of the arm motor system of the monkey and to the design of adaptive systems that transform chronically recorded brain signals into the motor output of artificial actuators. For that purpose, relatively simple but behaviorally meaningful motor actions such as a reaching movement and an exertion of force were chosen. We address the question of how movement variables are encoded in the motor cortex and how this information could be used to drive a simulated actuator that mimics the primate arm.

Cortical Representation of Movement

Coding by Single Cells

A common and behaviorally meaningful movement is reaching to targets in space. Reaching involves well-coordinated motion about the shoulder and elbow joints for transporting the hand in space and bringing it to a desired location. A reaching movement can be regarded as a vector, pointing from its origin to its target, with direction and amplitude.

A relation between the direction of reaching, \mathbf{M} , and the cell discharge rate, d , has been established for several brain areas, including the motor cortex, the premotor cortex, area 5, the cerebellar cortex, and the deep cerebellar nuclei (Georgopoulos, 1996). This relation is characterized by a broad tuning function, $d(\mathbf{M})$, the peak of which denotes the “preferred” direction of the cell, \mathbf{C} , that is, the direction of movement for which the cell’s activity would be highest (Figure 1). Typically, cell activity (discharge rate) varies

as a linear function of the cosine of the angle, $\theta_{\mathbf{MC}}$, between the preferred direction of the cell \mathbf{C} and the direction of reaching \mathbf{M} :

$$d(\mathbf{M}) = b + k \cos \theta_{\mathbf{MC}} \quad (1)$$

where b and k are cell-specific regression coefficients. (Although other functions could fit the data [Amirikian and Georgopoulos, 2000], the cosine function is a simple one that explains a good percentage of variation in cell activity.) Equation 1 holds both for reaching movements in a two-dimensional (2D) plane and for free reaching movements performed in three-dimensional (3D) space (Georgopoulos, 1996). Preferred directions of single cells range over the directional continuum and are multiply represented in the motor cortex.

Broad directional tuning of motor cortical cells was also observed with respect to the force pulses exerted by the monkey arm against an immovable object (Georgopoulos et al., 1992). A monkey was trained to exert forces in a 2D plane on an isometric handle in the presence of a constant force bias. First, the monkey was required to exert a postural (static) force \mathbf{P} , which compensated a given bias force \mathbf{B} ($\mathbf{P} + \mathbf{B} = 0$). After a holding period, a cue instructed the monkey to exert a force \mathbf{S} such that the net force \mathbf{N} acting on the handle (i.e., the force exerted by the monkey \mathbf{S} plus the bias force applied to the handle \mathbf{B}) would be in a visually specified (instructed) direction. Note that the net force \mathbf{N} is congruent to the incremental (dynamic) component \mathbf{I} of the force \mathbf{S} exerted by the animal: $\mathbf{I} = \mathbf{S} - \mathbf{P} = \mathbf{S} + \mathbf{B} = \mathbf{N}$. Eight instructed directions and eight bias force directions evenly distributed in the 2D plane were employed. Recordings of neuronal activity in the motor cortex revealed that when the arm exerted a force without moving, the activity of single cells showed approximately the same broad directional tuning properties as when the arm moved through space. Cells were tuned not to the direction of force \mathbf{S} exerted by the animal but to the direction of the dynamic component \mathbf{I} of the force \mathbf{S} .

Fu, Suarez, and Ebner (1993) investigated the relations between neuronal activity at the single-cell level and the amplitude of motor action. In these studies monkeys moved a handle over a planar working surface in eight directions (0 – 360° , in 45° intervals) and six amplitudes (1.4–5.4 cm, in 0.8-cm increments) in a pseudo-random order. The activity of cells in the motor and premotor cortex was directionally tuned in a cosine fashion, as described previously; the preferred direction was very similar for movements of different amplitudes. Cell activity generally increased with movement amplitude. Two aspects of this latter finding are noteworthy: first, the highest increase in neuronal activity with movement amplitude was not always along the cell’s preferred direction; and second, the strongest relations with movement amplitude were observed for cell activity during but not before the movement (in the latter case, the direction of movement is the most important factor). These findings indicate that the motor cortex is involved primarily in the *specification* of movement direction, as the movement is planned during the reaction time, and in *monitoring* movement amplitude, as the movement evolves during the movement time.

The relationship of ongoing cell activity to evolving movement variables such as position, speed, and acceleration was also studied (Ashe and Georgopoulos, 1994; Schwartz and Moran, 2000). It was found that cell activity was related to all these parameters, although movement velocity and target direction provided the main contribution. Therefore, the activity of motor cortical cells can be tuned to several movement parameters.

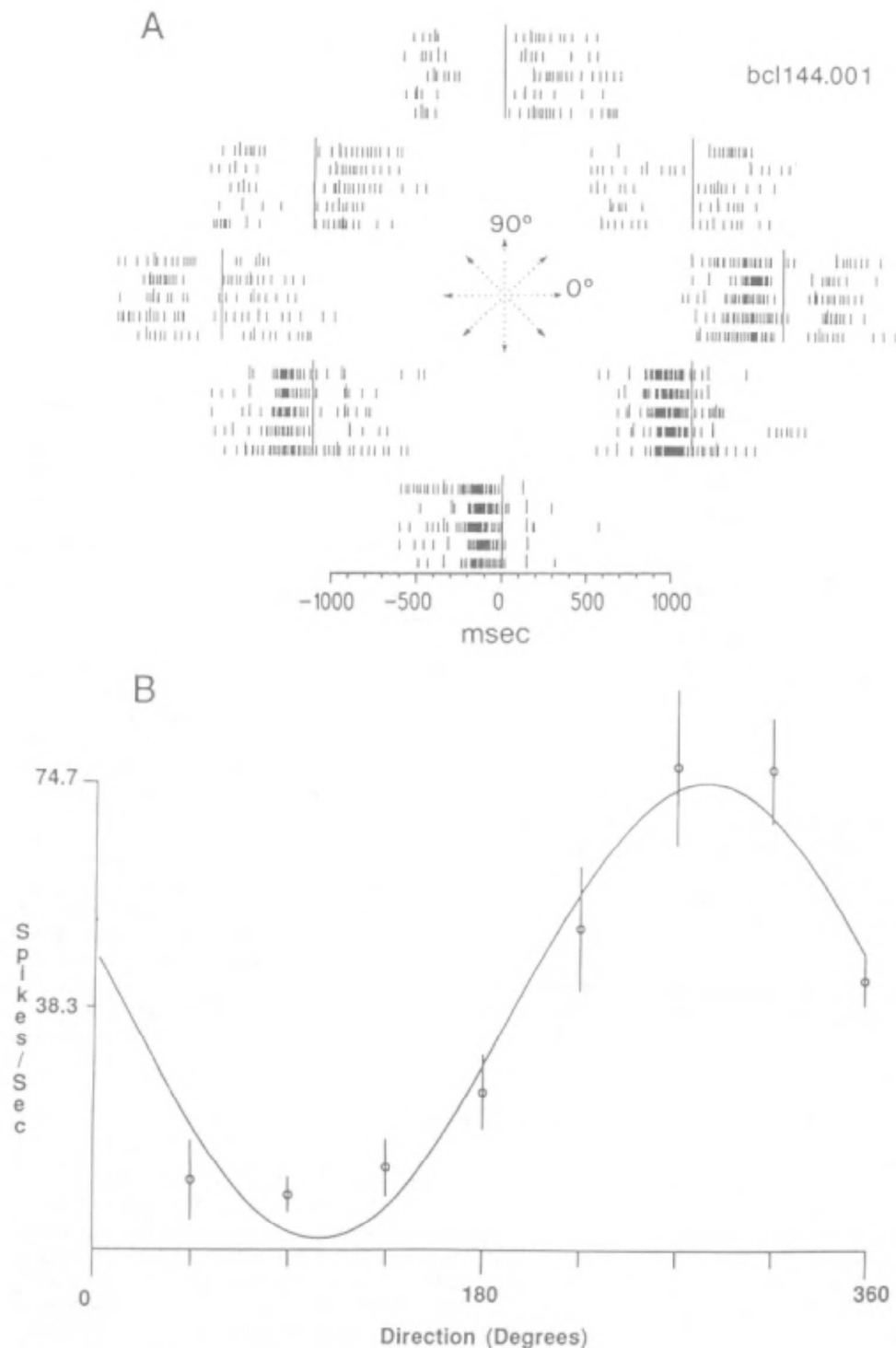


Figure 1. Discharge patterns during the center-out task. *A*, Rasters are arranged schematically at each target location around the center start position. Each raster is aligned to the time of exit from the center start position (zero time). The first long tickmark of each trial is the target onset time, the second is the time of movement onset, and the third is the time of target acquisition. *B*, The cosine tuning function was derived from the average rate of discharge (circles) between target onset and target acquisition for

each movement. The error bars show the standard deviation of the discharge rate. The regression coefficients (see Equation 1 in the text) were as follows: $b = 37.9$ spikes/s, $k = 36.1$ spikes/s. The proportion of variance in discharge rate explained by the regression was $R^2 = 0.95$. (From Schwartz, A. B., 1992, Motor cortical activity during drawing movements: Single-unit activity during sinusoid tracing, *J. Neurophysiol.*, 68:528–541. Reprinted with permission.)

The findings of the studies reviewed above regard discrete reaching movements. Schwartz (1992) studied the neural mechanisms of continuous, drawing movements by recording the activity of single cells in the motor cortex while the monkey traced on a touch screen sinusoids of various amplitudes and spatial frequencies. Under these conditions, the direction and speed of movement changed continuously in time. In another task, monkeys made equal-amplitude movements from a central point to peripheral targets (center-out task). The following were found: (1) in the center-out task, cell activity varied in a cosine fashion with the direction of the movement, as found previously; (2) in the tracing task, the ongoing direction of movement explained most of the variance in ongoing cell activity; and (3) a good proportion of the remaining, nondirectional variance could be accounted for by the ongoing speed of the movement. This relation to speed was best observed for movements near the cell's preferred direction.

Recently, a thorough assessment of single-cell recording studies in primates (Johnson, Mason, and Ebner, 2001) emphasized the idea of a multiparametric control of movement. Particularly, it was pointed out that the activity of single cells in central motor structures relates to several motor parameters (e.g., end-point force, acceleration, velocity, position), the relative contribution of which may vary in time and is influenced by the motor behavioral context.

Coding by Neuronal Populations

The broad directional tuning of single-cell activity indicates that a given cell participates in movements of various directions; from this result, and from the fact that preferred directions range widely, it follows that a movement in a particular direction will engage a whole population of cells. A unique code for the direction of movement (Georgopoulos, 1996) regards this population of directionally tuned cells as an ensemble of vectors in which each vector stands for the contribution of an individual cell. Specifically, the i th cell is represented by a vector that points in the cell's preferred direction \mathbf{C}_i and has a length $w_i(\mathbf{M})$ proportional to the change in cell activity associated with a particular movement direction \mathbf{M} . The vector sum of these neuronal contributions is the "population vector":

$$\mathbf{P}(\mathbf{M}) = \sum_{i=1}^N w_i(\mathbf{M})\mathbf{C}_i \quad (2)$$

where N is the number of cells in the population. The population vector points in the direction of the movement for discrete movements in 2D and 3D space.

The length of the population vector is proportional to the instantaneous speed of the movement (Schwartz and Moran, 2000). The time series of population vectors calculated during the movement were added successively tip-to-tail, resulting in a "neural" trajectory that predicted well the ensuing trajectory of the actual movement by an average time lead of approximately 120 ms. Therefore, the population vector carries information concerning the unfolding movement trajectory.

The population vector algorithm was also used to retrieve information encoded in the ensemble of directionally tuned cells recorded while the monkey executed the isometric force task (Georgopoulos et al., 1992) described above. It turned out that the time-varying population vector reflected neither the force \mathbf{S} exerted by the animal, which changed appreciably in direction and magnitude during individual trials, nor the static component \mathbf{P} of the force \mathbf{S} , which compensated for the constant force bias \mathbf{B} . Instead, the direction of the population vector remained invariant during the trial and pointed in the direction of the dynamic component \mathbf{I} of the force \mathbf{S} exerted by the monkey. Based on this analysis, it was hypothesized (Georgopoulos et al., 1992) that the motor cortex provides the dynamic component of the force signal during force

development, while other, possibly subcortical structures provide the static compensatory signal. These signals could converge in the spinal cord and provide an ongoing integrated signal to the motor neuronal pools.

It is important to realize that the population vector is a simple algorithm that retrieves an encoded variable from the activity of cells tuned to that variable, without making any assumptions about how the tuning itself emerges. There is controversy, however, as to what kind of variables these cells are coding: low-level variables such as muscle forces, or high-level variables such as end-point velocity (Flash and Sejnowski, 2001; Johnson et al., 2001). The matter has been recently sharpened by Todorov (2000), who proposed a simple model that (under certain assumptions) explicitly related cell activity to end-point force, acceleration, velocity, and position during small-amplitude movements. Interestingly, the model is consistent with experimental observations of multiparametric tuning of motor cortical cells (Johnson et al., 2001), and directly states that a *required* directional cell tuning is defined by a linear combination of end-point force, acceleration, velocity, and position terms. The relative contribution of each of these terms to the tuning depends on model parameters as well as experimental conditions (e.g., external loads). The main emphasis of Todorov's paper, however, is the "reinterpretation of the population vector." Since model cells were not explicitly related to high-level parameters but directly controlled low-level parameters (muscle forces), it was claimed that the view that motor cortex codes low-level variables "is in principle correct." Conversely, experimentally observed correlations with high-level motor variables (Johnson et al., 2001) is an epiphenomenon rather than a true neural code (Scott, 2000).

Todorov's model, however, does not allow one to make such far-reaching claims. The point is that the model is not "closed": the directional tuning is not an emergent property of the model but rather is built into it. Therefore, the derived expression for cell activity (Equation 2, Todorov, 2000), which is a starting point for all of Todorov's results, specifies a *required* activity of directionally tuned cells in order to generate a particular movement. The model, however, does not concern itself with *how* this required time-varying activity is produced in the course of movement. Note that cortical cell activity is a response to dynamic inputs received from cells in a local environment as well as inputs received from remote cortical areas. None of these factors is present in the model. Therefore, whether the information conveyed to individual cells via these inputs is related to low- or high-level motor variables cannot be answered in the framework of the model. The question of a *true* code is simply beyond the scope of this model. In a sense, the situation is similar to experimental studies in which cell activity, but not the cause of this activity, is recorded.

Decoding Motor Cortical Signals

The population vector code, by combining activities of broadly tuned cells, provides an unambiguous and reliable estimation (see POPULATION CODES) of the upcoming motor output (movement or force). However, the population vector allows only reading out as a single vector the cortical representation of distributed motor commands. It does not answer the question of how the motor commands, encoded in the cell activities, are translated into coordinated contraction of limb muscles to generate a desired motor action. This problem is closely related to the design of adaptive systems that transform neuronal signals chronically recorded from the motor cortex into physiologically appropriate motor output of artificial actuators such as multijoint prosthetic limbs (Schwartz, Taylor, and Helms Tillery, 2001; see BRAIN-COMPUTER INTERFACES).

Despite apparent similarities between these two problems, the approaches suitable for attacking each one pursue different goals.

The methods used to solve the first problem are heavily based on anatomical, neurophysiological, and biomechanical properties of the actual biological structures involved. The ultimate goal of this data-driven approach is to understand how real biological systems implement motor control. In contrast, the methods used to solve the second problem are usually based on theoretical analysis, the ultimate goal of which is to develop a computational algorithm that utilizes the raw biological signals to drive an artificial actuator. Here, the computational scheme does not try to be biologically realistic. Any control algorithm that successfully solves the problem is acceptable. Despite the difference in goals, the importance of interplay between the data- and theory-driven approaches should not be underestimated. The ideas and concepts developed in one field drive the other, and vice versa.

This was illustrated by Lukashin, Amirikian, and Georgopoulos (1996a, 1996b), who addressed the question of how neuronal signals might be used to drive an artificial actuator so that its motor output would correspond to the performance of a real limb. They suggested a computational scheme that transformed impulse activity recorded from the monkey motor cortex during performance of the isometric force task to the force produced by a simulated actuator. Although this computational model is not an accurate realization of biological motor control, it is biologically plausible. The model was inspired by, and based on, experimental findings obtained in anatomical, neurophysiological, and psychophysical studies conducted on three different organisms: (1) experiments on microstimulation of the frog's spinal cord (see MOTOR PRIMITIVES); (2) studies of human arm stiffness characteristics (Mussa-Ivaldi, Hogan, and Bizzi, 1985; see also references in EQUILIBRIUM POINT HYPOTHESIS); and (3) single-cell recordings in the motor cortex of monkey (Georgopoulos, 1996).

Transformation of Spiking Activity into Isometric Force Exerted by an Actuator

The motor cortical activity used in Lukashin et al., (1996b) as command signals came from single-cell recording experiments (Georgopoulos et al., 1992), discussed above, when there was no force bias applied ($\mathbf{B} = 0$). In this case, the force \mathbf{S} exerted by the monkey had only the dynamic component \mathbf{I} and was the only force acting on the isometric handle. Therefore, the force \mathbf{S} developed over time had to be in the instructed direction and had to increase in magnitude in order to exceed a required threshold. The bottom left part of Figure 2 shows an example of impulse activity of $N = 15$ different cells. These spike trains were recorded in different trials but for the same instructed direction of force (180° in the 2D workspace, Figure 2, top left). The key idea is that these cortical signals must now drive a simulated actuator (Figure 2, top right) in such a way that it would exert an isometric force in the same direction as the monkey did.

The actuator is a planar two-joint, six-muscle model of the arm. The transformation of cortical signals into motor output of the actuator is performed by an artificial neural network (Figure 2, bottom right) connected to the actuator. The network receives experimentally measured impulse activity as a time-varying input to the input layer and transforms it into a time-varying pattern of activity at the output layer. This results in contraction of actuator "muscles" by means of changing the muscle rest lengths. Finally, a set of the muscle rest lengths unambiguously defines the direction and magnitude of the end-point force exerted by the actuator against an immovable object. Thus the motor control algorithm realized in this model transformed a neural field, i.e., cortical activity, to a force field, i.e., end-point force (see GEOMETRICAL PRINCIPLES IN MOTOR CONTROL).

In the framework of this computational scheme, the performance of the model (i.e., the relation between the input cortical signals

and the force exerted) depends mainly on the network connectivity, which must provide a synergistic activation of all muscles to generate a required motor action. To ensure physiologically normal motor output of the actuator, the network was trained (Lukashin et al., 1996a) on experimental data obtained from studies of human arm stiffness (Mussa-Ivaldi et al., 1985). As a result, the stiffness properties of the model arm were similar to those measured for the human arm. Moreover, the biological relevance of this model was further independently tested (Lukashin et al., 1996a) by simulating experiments on microstimulation of frog spinal cord (see MOTOR PRIMITIVES). The model was fully consistent with experimentally observed vector summation of active force fields, realizing the idea of a linear combination of a small number of force field primitives to produce a large repertoire of motor behaviors (see MOTOR PRIMITIVES).

After training, the underlying set of synaptic weights was fixed and the performance of the model was tested against the whole neurophysiological data set (Georgopoulos et al., 1992), which included both experimentally measured motor cortical commands and resulting motor actions. An important issue of the performance is the robustness of the decoding scheme with respect to (1) the size of the population of cells generating neuronal signals, (2) variations in the composition of cells included in the population of a given size, and, finally, (3) changes in the cell activity from trial to trial.

The uniformity of the distribution of the cells' preferred directions throughout space is of particular importance in reconstructing the directional signal encoded in the population activity (see POPULATION CODES). Therefore, all other factors being equal, the best performance for a fixed-size population is expected for such a composition of cells whose preferred directions form a nearly uniform distribution. Lukashin et al., (1996b) found that when the uniformity requirement is fulfilled, the performance and robustness of the model improve gradually as the size of the population (N) increases, revealing a tendency for saturation as N approaches 15–20 cells.

Typical results for $N = 15$ demonstrating the time-evolving performance of the model are displayed in Figure 3. The forces exerted by the monkey and by the actuator are shown for four instructed directions, together with the corresponding motor cortical activity. It can be seen (Figure 3, templates A1, B1, C1, D1) that directional tuning of cortical impulse activity is barely perceptible, owing to the large variability in cell discharge. However, the computation scheme successfully decodes these signals, and the time-varying forces developed by the actuator (Figure 3, templates A3, B3, C3, D3) are very similar to those developed by the monkey arm (Figure 3, templates A2, B2, C2, D2). Following an initial period of time, which lasts 100–200 ms, the direction of force exerted by the actuator stabilizes and the magnitude of force increases. The stabilized direction of force is close to the instructed direction for which the cortical activity was recorded. This was also observed for the remaining four instructed directions for this particular ensemble of cells and for other ensembles of cells ($N \geq 15$) and trials, thus suggesting a high degree of robustness of the decoding algorithm.

Discussion

One of the major challenges of brain theory is to elucidate the neural basis of behavior. Significant progress has been made over the past decade in determining functional properties of single motor cortical cells with respect to relatively simple yet behaviorally meaningful motor actions. The finding that neurons are broadly tuned to behavioral variables led to the key idea of distributive coding. The population vector algorithm allows a read-out of this code by transforming aggregates of purely temporal spike trains into a spatiotemporal vector. The neuronal population vector has

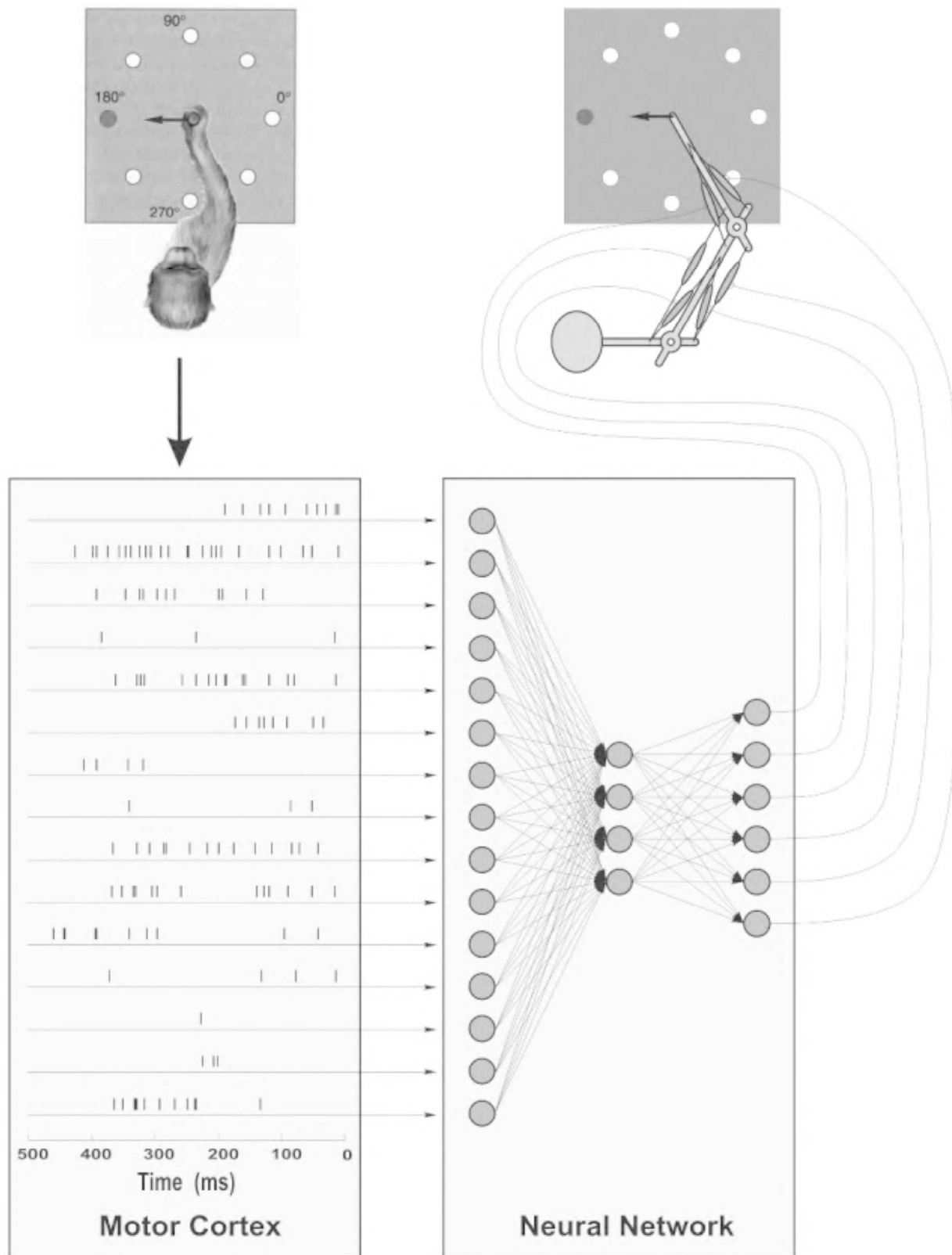


Figure 2. The decoding computation scheme used to transform neuronal commands encoded in a series of action potentials into a force exerted by the simulated actuator. The top left part of the figure illustrates a monkey exerting a force against the immovable handle in one (180°) of eight instructed directions. An example of the motor cortical activity recorded

while the animal performed this task is represented in the bottom left panel. These neuronal signals drive the simulated actuator sketched in the top right part of the figure. A three-layered feedforward neural network (the directed connections are shown by thin arrows) transforms cortical signals into coordinated activation of actuator muscles.

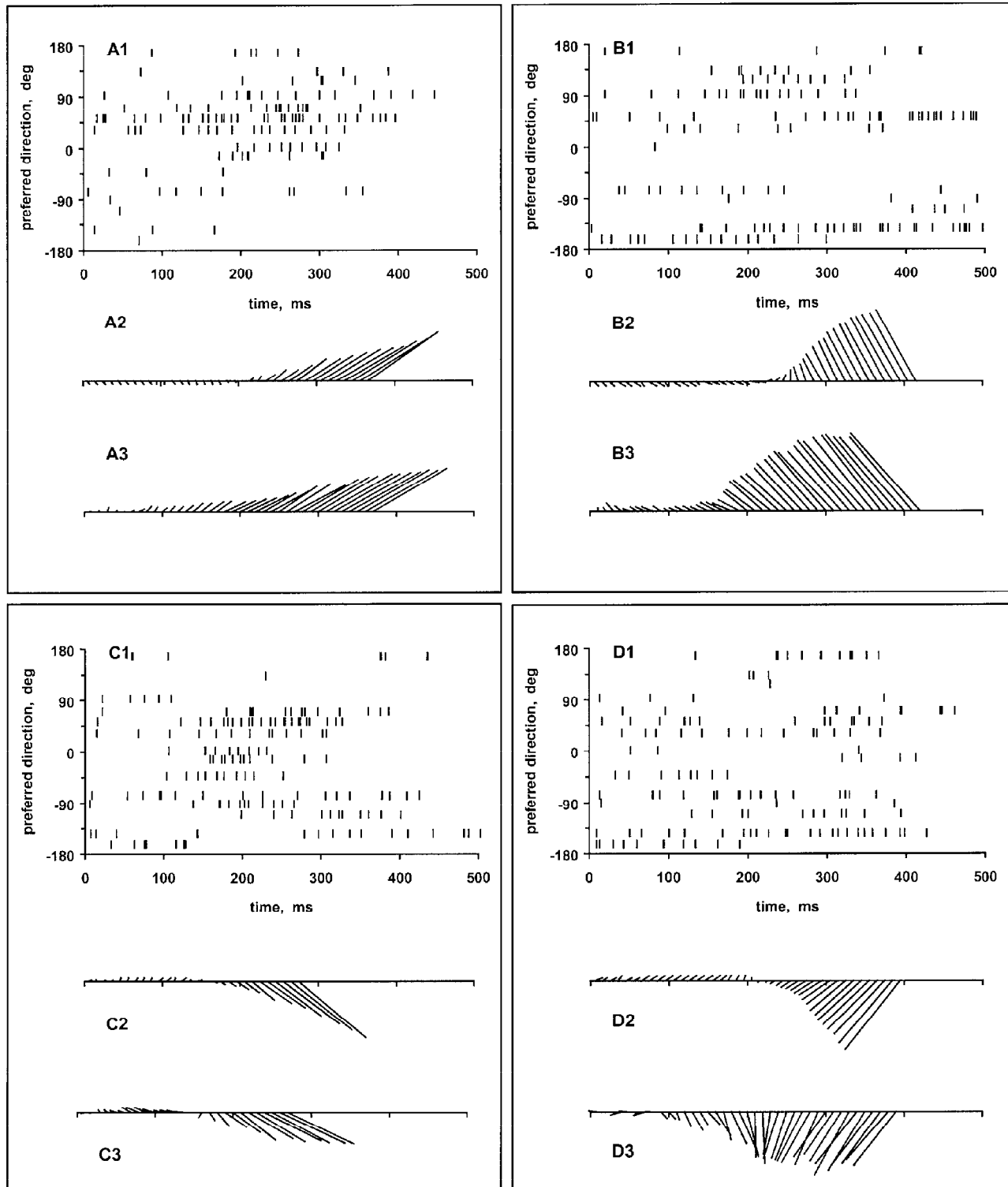


Figure 3. The motor cortical activity and the forces exerted by a monkey and by a simulated actuator. Four rasters, A1, B1, C1, and D1, show spiking activity recorded while the monkey developed isometric forces in four different instructed directions (45° , 135° , -45° , and -135° , respectively). The spike trains are aligned to the time of visual stimulus (zero time) instructing the monkey to begin the motor action. Each raster includes spike trains for the same 15 cells ordered along the vertical axis in accordance with their preferred directions. The time-varying forces developed by the monkey are depicted in A2, B2, C2, and D2. The force vectors measured

every 10 ms are displayed as line segments; the tails of force vectors are aligned along the time axis (the horizontal line). The time scale is the same as shown in the rasters above. Finally, A3, B3, C3, and D3 show forces exerted by the actuator in response to the neuronal activity presented in the rasters. The forces were calculated using the decoding algorithm described (see text and Figure 2) with a 10-ms time step and were arranged along the time axis in the same way as experimentally measured forces. The measured and calculated forces are normalized to the same magnitude, assigned arbitrarily.

proved to be a robust and accurate measure of the directional tendency of a neuronal ensemble in different brain structures and under a variety of conditions. The discovery of cortical representation of motor commands, combined with the elucidation of neural mechanisms by which these commands generate a particular behavioral pattern unfolding in time, should further advance our understanding of neural basis of motor behavior.

The research in this direction also provides an impetus to the field of neuroprosthetics (see PROSTHETICS, NEURAL) and the design of adaptive systems that transform neuronal signals recorded from the brain into physiologically accurate motor output of multi-joint prosthetic limbs (see BRAIN-COMPUTER INTERFACES). The feasibility of this idea was demonstrated by Lukashin et al. (1996b) in a particular case when the required motor output was an exertion of isometric force. The simulated actuator, which mimicked the primate arm, responded to the experimentally recorded motor cortical commands with surprising fidelity, generating forces in quantitative agreement with those exerted by a trained monkey in both the temporal and spatial domains. An important finding was that even a small ensemble of cortical cells can reliably control relatively complex motor output. Recent studies (Schwartz et al., 2001, and references therein) have made further progress toward the cortical control of arm prosthetics.

Road Maps: Mammalian Brain Regions; Mammalian Motor Control; Neural Coding

Related Reading: Arm and Hand Movement Control; Brain-Computer Interfaces; Motor Primitives; Population Codes; Reaching Movements: Implications for Computational Model

References

- Amirikian, B., and Georgopoulos, A. P., 2000, Directional tuning profiles of motor cortical cells, *Neurosci. Res.*, 36:73–79.
- Ashe, J., and Georgopoulos, A. P., 1994, Movement parameters and neuronal activity in motor cortex and area 5, *Cereb. Cortex*, 6:590–600.
- Flash T., and Sejnowski T. J., 2001, Computational approaches to motor control, *Curr. Opin. Neurobiol.*, 11:655–662.
- Fu, Q. G., Suarez, J. I., and Ebner, T. J., 1993, Neuronal specification of direction and distance during reaching movements in the superior precentral premotor area and primary motor cortex of monkeys, *J. Neurophysiol.*, 70:2097–2116.
- Georgopoulos, A. P., 1996, Arm movements in monkeys: Behavior and neurophysiology, *J. Comp. Physiol. A*, 179:603–612. ♦
- Georgopoulos, A. P., Ashe, J., Smyrnis, N., and Taira, M., 1992, The motor cortex and the coding of force, *Science*, 256:1692–1695.
- Johnson, M. T. V., Mason, C. R., and Ebner, T. J., 2001, Central processes for multiparametric control of arm movements in primates, *Curr. Opin. Neurobiol.*, 11:684–688. ♦
- Lukashin, A. V., Amirikian, B. R., and Georgopoulos, A. P., 1996a, Neural computations underlying the exertion of force: A model, *Biol. Cybern.*, 74:469–478.
- Lukashin, A. V., Amirikian, B. R., and Georgopoulos, A. P., 1996b, A simulated actuator driven by motor cortical signals, *NeuroReport*, 7:2597–2601. ♦
- Mussa-Ivaldi, F. A., Hogan, N., and Bizzi, E., 1985, Neural, mechanical, and geometric factors subserving arm posture in humans, *J. Neurosci.*, 5:2732–2743.
- Schwartz, A. B., 1992, Motor cortical activity during drawing movements: Single-unit activity during sinusoid tracing, *J. Neurophysiol.*, 68:528–541.
- Schwartz, A. B., and Moran, D. W., 2000, Arm trajectory and representation of movement processing in motor cortical activity, *Eur. J. Neurosci.*, 12:1851–1856. ♦
- Schwartz, A. B., Taylor, D. M., and Helms Tillery, S., 2001, Extraction algorithms for cortical control of arm prosthetics, *Curr. Opin. Neurobiol.*, 11:701–707.
- Scott, S. H., 2000, Population vectors and motor cortex: neural coding or epiphenomenon? *Nature Neurosci.*, 3:307–308.
- Todorov, E., 2000, Direct cortical control of muscle activation in voluntary arm movements: A model, *Nature Neurosci.*, 3:391–398.

Motor Pattern Generation

Jeffrey Dean and Holk Cruse

Introduction

Movement is central to the survival of animals and the performance of machines. The elegance of animals moving in complex environments has long aroused curiosity and the desire to emulate this ability. Animals move using muscles that produce force, sensory systems that signal the state of the organism and its surroundings, and a nervous system that links the two and contributes its own intrinsic activity. All possess some characteristics unattractive to engineers. Muscles are relatively slow and their force varies with muscle length, its rate of change, and the pattern of activation (see MUSCLE MODELS). Most sense organs conflate information about a parameter's value and its rate of change (see ADAPTIVE SPIKE CODING). Most neurons transmit information slowly using a pulse code of limited bandwidth. Response variability necessitates feedback supervision (see MOTOR CONTROL, BIOLOGICAL AND THEORETICAL), but inherent transmission delays, component variability, and limited coding precision force feedback gains to be low to avoid instability and unwanted oscillations.

The contrast between component quality and exquisite motor performance raises major unsolved puzzles. One is whether component characteristics actually simplify control. For example, the spring-like properties of muscles might reduce movement planning to end-point specification (see EQUILIBRIUM POINT HYPOTHESIS).

Others lie in the distributed, highly parallel organization of biological control systems.

Metaphors and Tools

Since Descartes reduced animals to machines, biologists have applied metaphors from contemporary technology to the brain and motor control. As technology advanced, metaphors progressed from clock mechanisms and pneumatic actuators through card readers and magnetic tape to current computer-based concepts. For example, the orchestration of complex muscle activity was attributed to a central motor score that could be stored and replayed like magnetic tape: Graham Hoyle envisioned both a motor tape for motor neuron activation (*motor output* in Figure 1) and a sensory tape for the sensory signals expected during unperturbed execution (*efference copy* in Figure 1).

Currently dominant is *motor program*, analogous to the instructions for a computer (Keele, Cohen, and Ivry, 1990). Varying usage led to alternative suggestions (e.g., *coordinated control program*: see SCHEMA THEORY; *motor control structure*: Cruse et al., 1990). *Motor program* is still apt if program is understood very generally as code that can generate complex output in the absence of input, vary its output depending on inputs, and modify itself. The important caveat is that the term not imply classical von Neumann ar-

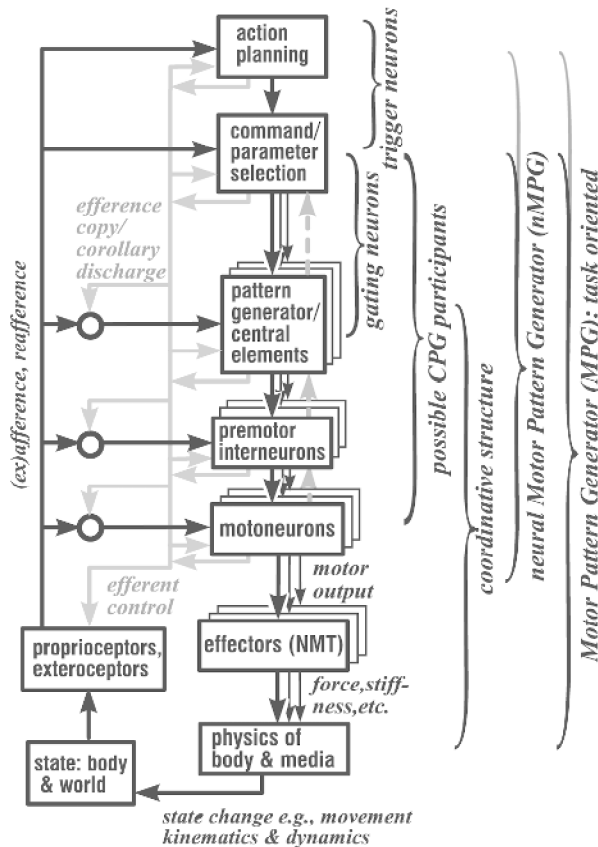


Figure 1. A hierarchical description of motor pattern generation showing potential interactions between efferent (motor), afferent (sensory), and central elements. nMPG is the set of neural elements, together with their states and interactions, that produces patterned motor output for a particular behavior; MPG includes the properties of effectors, body, and substrate. Sensory information can modify activity at all levels, but may itself be modified by central activity (efference copy, corollary discharge). Particularly in vertebrates, recurrent connections or corollary pathways may equal or exceed forward connections. For simplicity, the gray pathway (left) combines ascending (recurrent, internal loop) and descending interactions. Not shown are multi-element recurrent loops that implicitly or explicitly implement internal representations, models, and model-reference control. In complex behavioral sequences, planning instances participate in nMPGs. Even in simple nervous systems, functional hierarchies need not correspond to separate anatomical levels: individual neurons may participate in several functions (three leftmost brackets) and transmit signals forward and backward across several levels. NMT, neuromuscular transform.

chitectures with one processor working through sequences of instructions stored in separate memory. Instead, motor programs represent states established in the nervous system that produce coherent, task-related patterns of intrinsic activity and responses to stimuli. Like schemas, they are high-level shorthand for the function of parallel, distributed physiological systems (e.g., Figure 1).

This article focuses on the biological implementation of motor programs. Biomechanics and neurophysiology, supported by neuroanatomy, modeling, and molecular techniques, provide data. Cybernetics, the formal analysis of a system's input-output characteristics whereby systems can range from single neurons to whole animals (see *MOTOR CONTROL, BIOLOGICAL AND THEORETICAL*), provides systems-level analytical tools. So does the theory of dynamic systems (see *COOPERATIVE PHENOMENA*), which seeks collective variables summarizing emergent properties of complex systems and control parameters influencing their behavior.

Motor Pattern Generation

Like any physical system, animals move subject to the forces acting within and upon them. Animals actively control muscle forces, so generating movement is equivalent to generating appropriate patterns of muscle activity. As Sherrington noted, regardless of the complexity of the central nervous system (CNS), all motor influences must converge on or before the motor neurons, the final common path. *Motor output*—activity in this final common path recorded as muscle electrical activity (EMG), efferent activity in peripheral nerves, or activity in identified motor neurons—is often a convenient measure when experimental conditions preclude movement.

Although typical hierarchical descriptions of motor control (Figure 1) depict effectors, usually muscles, transforming into action a plan represented in motor output, the relationship between motor output and muscle force or movement is not simple (Hooper and Weaver, 2000; see *MUSCLE MODELS*). Nonmuscular forces like gravity or tendon and substrate elasticity strongly influence many movements. The characteristics of the body as a physical plant and the body–environment interaction may already be incorporated into motor output (Chiel and Beer, 1997), further obscuring the relationship between movement and neural activity.

Movement itself and motor output are called motor patterns, but we will focus on active control and use *neural motor pattern generator (nMPG)* for the system producing (neural) motor output that in appropriate contexts causes natural behavior. nMPGs can include simple and complex reflexes as well as intrinsic CNS activity. Many scientists would combine the nMPG, the body realizing its output, and the environment into a single dynamical system, arguing that each can contribute equally to molding the movement (Chiel and Beer, 1997); we will use *motor pattern generator (MPG)* for this dynamical system.

MPGs are conventionally linked to particular behaviors or actions, but defining units of behavior and thus MPGs is a matter of taste and inherently recursive (see *SCHEMA THEORY*). Investigators adopting whole-animal, functional perspectives might choose walking and a walking MPG as convenient initial units, so that refining the analysis to consider single legs or joints leads to models with many subunits in a walking MPG (e.g., Figure 1). Other investigators might begin with joint flexion and a flexion MPG, so that expanding the scope leads to models with many MPGs interacting to produce walking.

Regardless of the choice, similar questions arise. First, how is a particular motor pattern generated? Which system properties are necessary? Which are sufficient? Is the output stable with respect to external disturbances? Next, how is the pattern switched on and off? How are movement speed, form, and amplitude specified and adapted to task demands? How are external and internal state variables reflected within the nMPG? Finally, in ontogeny and evolution, do genes encode precise circuits—specifying each element and its connections—or crude circuits plus general measures of activity that engage plasticity mechanisms to achieve appropriate behavior (see *ACTIVITY-DEPENDENT REGULATION OF NEURONAL CONDUCTANCES*), and how do MPGs for similar behaviors compare in different species?

A Brief History of Theories of Motor Pattern Generation

When physiological study of the nervous system began in earnest, the easily accessible parts—peripheral sensory and motor elements and their combination in elementary reflexes—were investigated first. Because theories are built on what is known, reflexes assumed central roles, making animals into reactive, reflex-driven machines. In biology, motor patterns were explained as chains of reflexes in which each movement creates appropriate sensory stimuli to elicit

the next movement, leading to complex behavioral sequences or, if the chain is closed, rhythmic behaviors. Timing and continuation of leg movements in walking were attributed to reflexes based on changes in leg position, loading, and equilibrium signals during the step, but such conceptual models were never simulated quantitatively. In psychology, the analogous theory was *response chaining*. Such reaction-based formulations were extended in ethology to interactions between individuals and in psychology to behaviorism.

The rigidity of reflex or response chain theories caused their downfall (e.g., Lashley, 1951; Keele et al., 1990). Both had to postulate subtle stimulus differences to account for different behaviors under seemingly identical conditions and were unable to explain rule-based behavior in novel situations. Response chains based on reflex loops cannot explain rapid sequences when event intervals are shorter than minimum response latencies. Chains based on internal loops have difficulty explaining concurrency of preparatory movements and their dependence on preceding *and* succeeding actions.

A new theory was needed, one incorporating generalized representations of behaviors—plans or schemas (Lashley, 1951; see SCHEMA THEORY) or motor programs—which are modifiable by external and internal influences. In psychology, neurological findings implicate specific brain structures in planning (see BASAL GANGLIA; CEREBELLUM AND MOTOR CONTROL), but the details are unclear. In biology, motor programs acquired physiological foundations, beginning with the realization that the CNS alone can produce patterned motor output. T. Graham Brown showed that cats deprived of sensory inputs could still produce coordinated stepping; he envisioned a neural oscillator producing alternating activity in flexor and extensor muscles. His experimental approach—deafferentation—and his conceptual model both remain current (see HALF-CENTER OSCILLATORS UNDERLYING RHYTHMIC MOVEMENTS). Later, von Holst's (see Delcomyn, 1980) quantitative analyses of rhythmic behaviors were best explained by interactions among multiple intrinsic rhythm generators.

As physiological techniques improved, intrinsic motor patterns were traced within the CNS. Individual neurons or networks able, in the absence of patterned sensory inputs, to produce patterned motor output related to natural behaviors are generally called *central pattern generators (CPGs)*. *Neural oscillator* or *central oscillator* are terms also used, particularly for rhythmic patterns, but CPG includes discrete movements. (Naming CPGs again raises questions of system boundaries. CPGs in the lobster stomatogastric system originally distinguished according to anatomical segment and function were later found to interact in complex ways within what in a broader view is an ingestion CPG; see CRUSTACEAN STOMATOAGASTRIC SYSTEM.)

For a time, the importance of CPGs was overemphasized at the expense of peripheral elements. Wilson's demonstration of basic flight rhythms in the efferent motor fibers of deafferented locusts was particularly influential (see LOCUST FLIGHT: COMPONENTS AND MECHANISMS IN THE MOTOR). Sensory influences were relegated to providing tonic excitation rather than cycle-by-cycle patterning. Initial physiological descriptions of neural oscillators reinforced this emphasis, especially when evidence for CPGs was found in virtually all rhythmic behaviors (Delcomyn, 1980). Besides the physiological data, however, a willingness to overlook differences between natural motor patterns and those in deafferented preparations contributed to the emphasis on CPGs (Pearson, 1993).

The necessary correction occurred when it was shown that sensory inputs, besides influencing frequency, timing, and form of rhythmic activity, often fulfill both criteria for CPG inclusion (Pearson, 1993): (1) rhythmic activity correlated with the behavior and (2) the ability to shift or "reset" the phase of ongoing rhythms when appropriately stimulated. For example, actual wing movement and

loading, visual stimuli, and changes in air flow past the head modulate and stabilize locust wing movement on a cycle-by-cycle basis (see LOCUST FLIGHT: COMPONENTS AND MECHANISMS IN THE MOTOR). Similar entrainment by sensory inputs occurs in many systems with well-developed neural oscillators, as does modulation of transitions in sequences of nonrhythmic movements.

Peripheral influences are easily incorporated into motor programs and similar high-level concepts; MPG and nMPG inherently include both central and peripheral elements. Many researchers also expand the CPG concept to incorporate sensory influences when available, but this disrupts the natural congruity of functional and anatomical boundaries.

In summary, most MPGs incorporate both intrinsic CNS activity (CPGs) and peripheral influences reflecting biomechanical characteristics, reflexes, and proprioceptive and exteroceptive afferent activity (Pearson, 1993). As strategies for adaptive behavior, these two elements are analogous to network adaptation via genetic algorithms and via learning or developmental plasticity, respectively. The balance is subject to evolutionary selection, depending on the cost of errors and the predictability of motor command outcomes; it remains a subject of debate and research.

Central Pattern Generators

Understanding of CPGs progressed most rapidly for rhythmic behaviors. Initial conceptual models addressed simple two-phase rhythms, such as alternation between stance and swing. Half-center oscillator models contain two functional units connected by reciprocal inhibition and subject to a common excitatory drive, whereby functional units can be either single neurons or groups of neurons synchronized by reciprocal excitatory synapses. This bistable circuit oscillates if the inhibition decays or if other functionally equivalent changes occur (see Camhi, 1984). George Székely demonstrated that multiphase rhythms can be generated by neurons or functional groups in a ring with recurrent inhibition (e.g., Camhi, 1984).

Early physiological findings distinguished two kinds of neural oscillators. In *cellular oscillators (pacemakers; see OSCILLATORY AND BURSTING PROPERTIES OF NEURONS)*, rhythms depend on complex, nonlinear membrane properties and continue when the cell is isolated from the nervous system, whereas in *network oscillators* they depend on the connectivity of cells incapable of rhythmic activity on their own. Connectivity was emphasized initially because it was easier to characterize and because several vertebrate and invertebrate networks contained reciprocal or recurrent inhibition, providing a satisfying agreement with theoretical simulations using simple model neurons.

Subsequent results blurred the distinction and emphasized the contribution of cellular properties (Selverston et al., 1997). First, time delays necessary for half-center models require nonlinear properties. Second, some neurons within putative network oscillators, and even some motor neurons, possess nonlinear membrane properties, allowing them to oscillate or exhibit *plateau potentials*, switching between inactivity and prolonged high activity (see OSCILLATORY AND BURSTING PROPERTIES OF NEURONS; NEUROMODULATION IN INVERTEBRATE NERVOUS SYSTEMS). Unexpected pacemakers turned up within putative network oscillators. Third, synaptic interactions show similar complexity, changing in strength or even sign, depending on time or on the relative potentials of sender and recipient (see TEMPORAL DYNAMICS OF BIOLOGICAL SYNAPSES). Fourth, besides simple recurrent inhibition, real networks include positive feedback or mixtures of circuits. Even the five-neuron recurrent ring initially identified as the swim CPG of the leech contains reciprocal inhibitory connections within and diagonally across the ring (see Camhi, 1984). Finally, real CPGs

incorporate both cellular and network properties, with redundancy contributing to robust rhythmicity.

Complexity and redundancy impede attribution of function to particular elements even in simple invertebrate networks with few neurons. CPG simulations using realistic synaptic interactions and membrane characteristics are absolutely necessary. The practical difficulty is the ever-increasing list of properties of possible significance. Consideration of neuronal morphology adds further complexity (see DENDRITIC PROCESSING). Morphology influences responses according to both the timing and the location of inputs. Thus, simulation of even a single real neuron, let alone a network, is a formidable problem (see SINGLE-CELL MODELS). Even some well-studied CPGs lack definitive models. Although the optimism following early successful simulations using simplistic neurons dissipated, newer, more realistic models provide grounds for renewed optimism (e.g., Selverston et al., 2000).

Network simulations and physiological data do show that systems can perform appropriately even when some connections appear nonfunctional or even dysfunctional (e.g., excitatory connections to neurons normally silent when the sender is active). Such “rogue” elements (Robinson, 1992) occur even in simple resistance and withdrawal reflexes. Because natural selection works most directly on behavior, eliminating all rogue elements may not be possible or necessary: motor output is an emergent property of the whole CPG.

CPGs may drive normal movements or merely facilitate particular modes of motor output. They avoid peripheral delays at the expense of adaptability to bodily and environmental changes. They represent predictions for suitable motor output, or a kind of *internal model* for adaptive behavior in a given context. Given the incomplete understanding of simple CPGs and their somewhat stereotyped outputs, moving beyond these to understand flexible motor control presents major challenges. Recurrent connections prominent in higher vertebrates presumably contribute to more elaborate internal models supporting flexible nMPGs (see ACTION MONITORING AND FORWARD CONTROL OF MOVEMENTS; REACHING MOVEMENTS: IMPLICATIONS FOR COMPUTATIONAL MODELS).

Switching Patterns On and Off

Functionally, pattern generators are controlled by a switch. Neural elements implementing the switch were labeled *command neurons* (see Figure 1) (Kupfermann and Weiss, 2001). Activity in command neurons should be both necessary and sufficient for producing behavior. If action potentials are required, the neuron's threshold determines the behavioral threshold. In practice, few neurons fulfill both criteria (Camhi, 1984). More often, control is distributed among many neurons, reflecting the redundancy common in biological systems. As a result, the concept has been extended to “command systems” containing multiple “command elements” (see COMMAND NEURONS AND COMMAND SYSTEMS).

Two kinds of switches have been identified. Some behaviors continue only as long as an appropriate stimulus is present; others continue after it ends. A similar distinction applies to command neurons: activity in *trigger neurons* initiates a longer-lasting behavior, while activity in *gating neurons* determines the duration of the behavior. Command neurons turning on one behavior may also turn off or inhibit command neurons for interfering behaviors and excite those for functionally linked behaviors—an architecture replicated in P. Maes's ANN. If not actively turned off, gating neuron activity may simply decay below threshold, terminating the behavior.

Real nervous systems often mix command, coordination, and pattern generation, and behavioral choice involves multiple functional levels. Activity in some gating neurons oscillates with the output rhythm. Some can reset the rhythm and fulfill criteria for

CPG inclusion. Some leech gating neurons are functionally outside the CPG, but the recurrent, excitatory pathways modulating their activity help prolong this activity. In *Tritonia*'s escape system, the command signal reflects membrane properties of CPG neurons: depending on stimulus strength the network produces either a single longitudinal contraction or multicycle dorsoventral flexions. Thus, one anatomical network may be a *polymorphic network* producing qualitatively different motor patterns; in other words, individual neurons participate in multiple behaviors, and command is a population code (Kristan and Shaw, 1997).

Command signals often work via nonlinear effects of neural activity (see ACTIVITY-DEPENDENT REGULATION OF NEURONAL CONDUCTANCES) or modulatory transmitters and hormones (see NEUROMODULATION IN INVERTEBRATE NERVOUS SYSTEMS) that modify membrane properties, producing qualitative changes in outputs or even functional reconfigurations of nMPGs. Thus, biological nMPGs show a protean variability compared to artificial neural networks, where unit properties are usually static and output changes reflect input changes or connectivity changes through learning.

Sensory Influences

nMPGs usually incorporate feedforward and feedback using sensory information about the surroundings and motor performance. Formally, feedback depends on the consequences of motor activity. Negative feedback acts to reduce deviations (“errors”) from a reference value, as in resistance reflexes, whereas positive feedback (with suitable limiting nonlinearities) maintains or accentuates ongoing movements. Feedforward pathways help select actions or set parameters in advance, such as specifying the size of a saccade or an escape turn. However, in natural, continuous streams of behavior, distinguishing error-correcting feedback from parameter-setting feedforward mechanisms is not always clear (Cruse et al., 1990). For example, learning can be described as delayed negative feedback that adjusts future parameter setting.

Sensory signals can be modified at various levels (see Figure 1). Sensory activity can be modulated peripherally through efferent control or centrally at the output synapses of the primary afferent fibers. Reflexes occurring in one situation can be reversed or replaced by wholly new responses (Pearson, 1993). In invertebrates, resistance reflexes—negative feedback opposing leg displacement during posture—are replaced as stance begins by assistance reflexes to boost propulsion. At other times they remain active, opposing deviations from normal trajectories. Phase-dependent reflex modulation may be a natural consequence of spike thresholds in neurons (see LOCUST FLIGHT: COMPONENTS AND MECHANISMS IN THE MOTOR).

The relative importance of central and peripheral components varies considerably. For example, central elements appear weaker in walking than in flying and swimming, where the substrate is more forgiving. In flying, variability in wing movements affecting lift and steering can be corrected over several subsequent cycles, whereas in walking, not finding a foothold or tripping over an obstruction must be corrected immediately. When stick insects walk, movement of the leg itself, as signaled by sensory inputs, determines the state of the step CPG (Bässler, 1986; Dean et al., 1999); isolated CPGs produce slow, fragmentary, irregular motor output. Our work shows that a simple step rhythm can arise from a four-unit, recurrent nMPG in which two units represent the state and two units represent position criteria for changing state. Recurrent connections provide positive feedback to reinforce the current state, causing each movement to continue to its endpoint.

Other Peripheral Influences on Motor Patterns

Physical characteristics of muscles, tendons, skeletal elements, and the environment modify or even create movement. Motor behaviors

are often studied using motor output, but the transformation between this activity and movement is not trivial. Different neural rhythms in the lobster CPG controlling chewing were characterized long before their functional significance was determined (see CRUSTACEAN STOMATO-GASTRIC SYSTEM). Modeling the transformation from neural activity to muscle force is difficult but improving (Hooper and Weaver, 2000). Motor neuron activity is difficult to relate to movement, partly owing to the sophistication with which animals use passive and elastic properties of their muscles and skeletons. According to one hypothesis for limb movements to a target (see EQUILIBRIUM POINT HYPOTHESIS, but also ARM AND HAND MOVEMENT CONTROL), muscle activity is not related to movement as such but to the target, the allowable deviation, and an estimate of possible and allowable errors.

Including the periphery in MPG models is particularly important when it strongly affects movement. Algorithms for step coordination simulated using a simplified representation of leg position will be unable to control a machine with real legs if they try to place legs in unreachable positions. Behavioral data on limb interactions, the basis of many coordination models, actually represent the total effect of internal activity, reflexes, and the physics of animal and environment. Dynamical systems theory, using collective variables like phase, can sometimes provide a succinct encapsulation of this system behavior.

In extreme cases, significant aspects of movements are determined by the physical system. Treating bipedal walkers simply as a system of passive, damped pendulums explains many characteristics, suggesting that muscle activity is only occasionally necessary to maintain the pendular motion or correct disturbances (McGeer, 1990). In some insects, flight muscles require general activation, but individual wing beats result from mechanical oscillations involving the muscles' intrinsic response to stretch.

In summary, biological movement is not always driven rigidly by nMPG output (see CEREBELLUM AND MOTOR CONTROL). Taking advantage of physical properties may enable adequate control using neural approximations of exact algorithms and internal models (e.g., Dean et al., 1999), especially when animals are small or compliances are high, so that impacts or stresses arising from inaccuracies can be absorbed without injury. This may not be true of large, powerful robots in an environment where safety is an essential concern.

Coordinating Multiple Motor Patterns

Many behaviors contain distinct elements that occur concurrently or sequentially, requiring spatial and temporal coordination of body parts (e.g., EYE-HAND COORDINATION IN REACHING MOVEMENTS; LOCOMOTION, INVERTEBRATE) or the completion of one action before another begins. For this purpose, several MPGs or subunits within one MPG can be structured hierarchically or in parallel, depending on task requirements (see Figure 1). For speaking and typing, the kinds of errors and time delays, the concurrence of preparatory and execution phases, and the modulation of elements depending on earlier and later elements indicate a hierarchical arrangement, whereas relative coordination of concurrent rhythms indicates a parallel arrangement.

In these cases, interactions among and within MPGs can occur at many levels, ranging from overall planning through parameter setting and CPG activity to peripheral loops involving mechanical coupling and reafference. Network models offer natural simulations of these interactions. Rhythm coordination has been studied extensively (see CRUSTACEAN STOMATO-GASTRIC SYSTEM; CHAINS OF OSCILLATORS IN MOTOR AND SENSORY SYSTEMS). Coordination mechanisms within MPGs are often highly redundant and thus robust. In insect walking, multiple centrally mediated mechanisms

are augmented by multiple locally mediated (intraleg) mechanisms sensitive to leg position, state, and load (Dean et al., 1999).

Discussion

In the near future, interactions between theoreticians and experimentalists should be fruitful in many but not all respects. Neuroscience clearly needs computational neuroscientists using simulations of biological nMPGs to represent acquired knowledge and test its completeness. Artificial neural networks provide new tools and, more important, reemphasize several concepts that have important implications for interpreting physiological experiments (e.g., Robinson, 1992): (1) the common currency in neural networks is neuronal activity; (2) this activity need not be simply related to physical parameters; (3) processing is distributed and parallel; (4) redundancy contributes robustness; and (5) approximations may suffice.

Neural engineers naturally look to animals for inspiration in improving technical systems, expecting concise lists of biological solutions for different computational problems. Because animals are so diverse, neuroscience provides only rudimentary guidelines. Each species has its own answer to a very complex and specialized problem, an important part of which is reproduction. Genes and development constrain animals to evolve in small steps that are themselves adaptive, or at least not seriously detrimental. In contrast, technological innovation can create radically new machines; prototypes hopelessly outperformed by existing technologies are not doomed to extinction.

Nevertheless, the very diversity of biological solutions enhances what biology has to offer. Better understanding of biological MPGs should help algorithm selection for technical applications, especially as artificial networks approach the complexity of biological networks in size, connectivity, unit properties, and plasticity. Optimization techniques based on biological learning or evolution already help adapt control systems (see LOCOMOTION, INVERTEBRATE). *Animat research*, by studying real robots, shows that implementing MPGs in real robots sometimes simplifies control and representation issues. Purely reactive robots, with integration at the effectors and no explicit world models, can generate astonishingly complex behavior. Optimally balancing reactive control with implicit and explicit representations is a current challenge for animat research; understanding biological implementations is a challenge for neuroscience.

Road Map: Motor Pattern Generators

Related Reading: Command Neurons and Command Systems; Motor Control, Biological and Theoretical; Sensorimotor Learning

References

- Bässler, U., 1986, On the definition of central pattern generator and its sensory control, *Biol. Cybern.*, 54:65–69. ♦
- Camhi, J. M., 1984, *Neuroethology: Nerve Cells and the Natural Behavior of Animals*, Sunderland, MA: Sinauer. ♦
- Chiel, H. J., and Beer, R. D., 1997, The brain has a body: Adaptive behavior emerges from interactions of nervous system, body and environment, *Trends Neurosci.*, 20:553–557.
- Cruse, H., Dean, J., Heuer, H., and Schmidt, R. A., 1990, Utilization of sensory information for motor control, in *Relationships Between Perception and Action: Current Approaches* (O. Neumann and W. Prinz, Eds.), Berlin: Springer-Verlag, pp. 43–79. ♦
- Dean, J., Kindermann, T., Schmitz, J., Schumm, M., and Cruse, H., 1999, Control of walking in the stick insect: From behavior and physiology to modeling, *Auton. Robots*, 7:271–288.
- Delcomyn, F., 1980, Neural basis of rhythmic behavior in animals, *Science*, 210:492–498.

- Hooper, S. L., and Weaver, A. L., 2000, Motor neuron activity is often insufficient to predict motor response, *Curr. Opin. Neurobiol.*, 10:876–882.
- Keele, S. W., Cohen, A., and Ivry, R., 1990, Motor programs: Concepts and issues, in *Attention and Performance: XIII. Motor Representation and Control* (M. Jeannerod, Ed.), Hillsdale, NJ: Erlbaum, pp. 77–110. ♦
- Kristan, W. B., Jr., and Shaw, B. K., 1997, Population coding and behavioral choice, *Curr. Opin. Neurobiol.*, 7:826–831.
- Kupfermann, I., and Weiss, K. R., 2001, Motor program selection in simple model systems, *Curr. Opin. Neurobiol.*, 11:673–677.
- Lashley, K. S., 1951, The problem of serial order in behavior, in *Cerebral Mechanisms in Behavior: The Hixon Symposium* (L. A. Jeffress, Ed.), New York: Hafner, pp. 112–136.
- McGeer, T., 1990, Passive dynamic walking, *Int. J. Robot. Res.*, 9:62–82.
- Pearson, K. G., 1993, Common principles of motor control in vertebrates and invertebrates, *Annu. Rev. Neurosci.*, 16:265–297.
- Robinson, D. A., 1992, Implications of neural networks for how we think about brain function, *Behav. Brain Sci.*, 15:644–655. ♦
- Selverston, A. I., Panchin, Y. V., Arshavsky, Y. I., and Orlovsky, G. N., 1997, Shared features of invertebrate central pattern generators, in *Neurons, Networks, and Motor Behavior* (P. S. G. Stein, S. Grillner, A. I. Selverston, and D. F. Stuart, Eds.), Cambridge, MA: MIT Press, pp. 105–117. ♦
- Selverston, A. I., Rabinovich, M. I., Abarbanel, H. D. E., Elson, R., Szűcs, A., Pinto, R. D., Huerta, R., and Varona, P., 2000, Reliable circuits from irregular neurons: A dynamical approach to understanding central pattern generators, *J. Physiol. (Paris)*, 94:357–374.

Motor Primitives

Simon F. Giszter

Introduction

The concept of a modular organization of spinal motor systems dates back to Sherrington (1910) or earlier (see citation in Giszter et al., 2001a). Recently, a new perspective on motor system modularity (and especially spinal cord) has developed as a result of a series of experiments on the spinal cords of frog, rat, and cat. It has been proposed that a set of motor elements termed “motor primitives” are implemented in the spinal cord. These are recruited by central pattern generators to construct spinal motor acts. Data from the frog have been especially significant in this framework (see SCRATCH REFLEX). The frog wiping behaviors have been utilized in the study of motor control since the nineteenth century. The advantages of these behaviors are the clear adjustments documented, the simplicity of aspects of frog motor organization, and the robustness of the behaviors. The frog’s wiping responses are largely organized in the spinal cord: they persist unaltered following spinal transection (i.e., in the isolated spinal cord). Examination of wiping movements and microstimulation of frog spinal cord (and more recently an examination of descending controls) has generated a framework for describing the basis of spinal construction of motor behavior. Movements are constructed as a sequencing and combination of a collection of force-field motor primitives or fundamental elements. Although controversial, because the biological circuit underpinnings are not yet well established, this framework is proving promising, and to date, it is holding up under careful experimental scrutiny and implementation in robots (see Schaal, 1999).

Properties of Force-Field Motor Primitives

Microstimulation of frog spinal cord in intermediate gray regions using low currents (1–10 μ A, 0.5-ms pulses at 40 Hz for 300 ms) activates a few hundred or thousand cells and elicits specific groups of muscle responses. With the spinal frog held immobile in an isometric apparatus, these muscle responses could be characterized as an endpoint (ankle) force vector and a vector of associated joint torques (see Giszter, Mussa-Ivaldi, and Bizzi, 1993). This measurement was repeated at multiple limb configurations and stimulus strengths. At a single configuration, over time, and at increasing stimulation strength up to 15 μ A, the orientation of force elicited from a single site remained fixed, while the amplitude was modulated. By moving the limb to new positions, the effects of muscle-length-dependent viscoelastic properties, moment arms, and the ef-

fects of linkage kinematics on endpoint force and joint torques could be assessed. These effects were summarized by expressing the endpoint force as a function of endpoint position, or the joint torque vectors as a function of limb configuration (see, e.g., force fields measured in this way for a behavioral response in Figures 1F through 1H). When data were expressed in this way, it was discovered that the magnitude ratios and relative orientations of vectors across the range of positions remained fixed for a single site (Figure 1H). Force measured in isometric conditions could be expressed as a function of stimulus amplitude (s), configuration (r), and time (t) as

$$F(r, s, t) = A(s) \cdot a(t) \cdot \phi(r) \quad (1)$$

where $A(s)$ was a scalar function of stimulus strength and $a(t)$ was a scalar function of time; $\phi(r)$ was a fixed field structure that was simply scaled, was conservative, and was often convergent in both limb joint and limb endpoint coordinates. The function $a(t)$ varied with site of stimulation. There were phasic, tonic, and phasic/tonic sites (Giszter et al., 1993).

When the electrode was moved a small distance in spinal cord a similar field $\phi(r)$ was obtained. However, for larger changes in position of the electrode in spinal cord, the force pattern could alter significantly. Over a sampling of the lumbar spinal cord intermediate gray using microstimulation (see Bizzi et al., 2000; Giszter et al., 2001b), chemical stimulation (Saltiel, cited in Bizzi et al., 2000), or combined skin and microstimulation, a small set of force-field types was found. About six force-field patterns were found in all, separated in middle and deep intermediate zones by relatively silent areas (Giszter et al., 2001b).

Combination of stimuli applied to ipsilateral areas producing different force-field types resulted in one of two effects for costimulation: (1) In 85% of combinations, linear superposition or vector summation occurred (see Equation 2); (2) in 15% of combinations, winner-take-all occurred with one site dominating response (see Equation 3). Graded motion of a limb could be obtained by controlled costimulation. The combinations observed are consistent with modular construction of reflex responses; and see Mussa-Ivaldi (cited in Bizzi et al., 2000) and the section below entitled “Wiping Motor Pattern Construction and Control Using Force-Field Primitives.” In summary, for ipsilateral costimulation, either linear or winner-take-all combinations occur:

$$\begin{aligned} F(r, S_1, S_2) &= B \cdot [\Phi_1(r, S_1) + \Phi_2(r, S_2)] \\ &= B \cdot [A(S_1)\phi_1(r) + a(S_2)\phi_2(r)] \end{aligned} \quad (2)$$

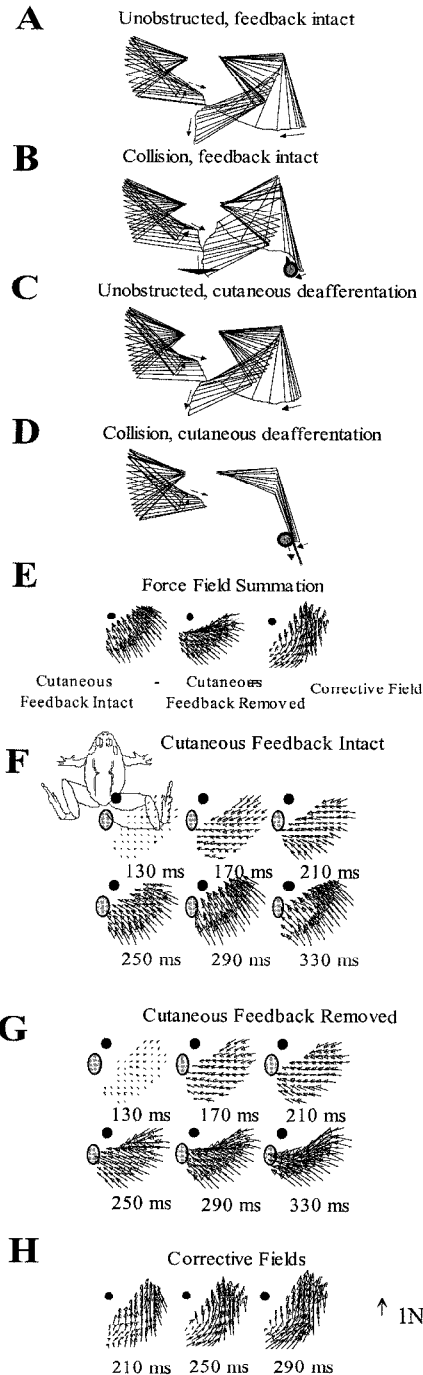


Figure 1. Examples of spinally organized responses and their force-field bases (redrawn from Giszter et al., 2000a; Kargo and Giszter, 2000). Upper panels: free-limb kinematics; lower panels: isometric force-field measurements. *A*, normal wipe trajectory. *B*, corrections to obstacles. *C*, loss of skin feedback from the effector limb (right limb) does not alter free trajectory. *D*, Corrections for collisions are abolished by loss of skin feedback. *E*, At each of a grid of ankle locations, the isometric force production is measured at the ankle. The correction response pattern is hypothesized to be due to vector summation of a corrective force-field primitive with the unperturbed pattern. *F*, field pattern with collision detection. *G*, field pattern without correction. *H*, Corrective response pattern is hypothesized to be due to subtraction (i.e., *G* from *F* as described in panel *E*). The force pattern is a scaled version of a single pattern at each time point shown, consistent with definitions used for a force-field primitive (see text). Data collected and published in Kargo and Giszter (2000), with support of NIH NS34640 and NS40412.

or

$$F(r, S_1, S_2) = B \cdot [\Phi_1(r, S_1)] \quad (3)$$

where B is a scalar scaling parameter and A and a are scalar functions of stimulus strength.

More recently, costimulation between contralateral sites has also been tested (Giszter et al., 2000a). For bilateral costimulation, some spinal cord regions do show nonlinear responses to costimulation while the remainder obey the combination rules above (Giszter et al., 2000a). Taken in total, these effects are consistent with microstimulation providing access to primitives and parts of the CPGs that can be combined in various ways.

Localization and Circuit Underpinnings of Motor Primitives

It is worth considering how microstimulation recruits cells and what the underpinnings of the functional relations observed above might be. The next sections demonstrate the functional uses of primitives in theory and real behavior, but the circuit underpinnings of these primitives are poorly understood, and the initial microstimulation that was used remains extremely controversial. Microstimulation as a technique of mapping has often been used. However, the laminar, nucleated, and segmentally organized spinal cord, such maps are harder to interpret and understand than, for example, in cortical columns. Spinal cord's recursive or looped projections, as opposed to simple output cascades, are confounding factors. Thus, although mappings with microstimulation or chemical stimuli reveal topographies, they may not reveal the localization (if any) that can be ascribed to primitives (Bizzi et al., 2000, and Giszter et al., 2001b). Rather, they are hot spots for more or less pure access to (and control of) primitives, which is entirely different. The circuit underpinning of the primitive may be discretely localized, broadly distributed, or an emergent pattern or mode of the spinal circuitry. Nonetheless, the fact that the circuitry supports these computational and functional primitives and that they may be accessed and combined in different ways through different localized sites is an important contribution. It may affect understanding of both motor control and neuroprostheses. Important future tasks are discovering how the circuitry supports primitives and how to link this meta-organization of primitives to the enormous body of exquisite physiology and anatomy of the spinal cord elaborated in the last 50 years by groups in Sweden, Denmark, Russia, Britain, Canada, and the United States. Some of the recent findings provide a series of strong hypotheses for this future work.

Capabilities of a System of Primitives: Theory

Important theoretical issues for motor primitives are (1) how to build time-varying field structures from motor primitives to support movement, (2) discovering the motor primitive basis sets that best provide general approximations and stability guarantees, and (3) discovering how appropriate basis sets might be constructed and modified by developmental or learning mechanisms.

Initially, Mussa-Ivaldi (1992, cited in Giszter et al., 2000b) showed that arbitrary fields were readily approximated by using a linear combination of conservative and circulating (or rotational) radial vector field primitives. In biological testing, circulating fields accounted for less than 5% of the variance of field structures observed. This concept has now been applied in the frog and in human biomechanics (see EQUILIBRIUM POINT HYPOTHESIS). The coefficients of these basis-field elements could be found by standard methods.

Mathematically, the force field $F(x)$ relating force to position x can be approximated by using k basis fields $q_i(x)$ and control parameters c_i :

$$F(x) = \sum_{i=1}^k c_i q_i(x) \quad (4)$$

The k basis fields $q_i(x)$ are subdivided into two groupings. These two groupings consist of $k/2$ conservative or irrotational fields and $k/2$ solenoidal or circulating fields. Thus,

$$F(x) = C(x) + R(x) = \sum_{i=1}^{k/2} q_i(x) + \sum_{i=k/2+1}^k q_i(x) \quad (5)$$

where $C(x)$ is a pure conservative field and $R(x)$ a pure solenoidal field. Circulating fields have curl and can be used to generate energy in closed cycles. In biological measures to date, $R(x)$ is generally found to be sufficiently small that it can be neglected.

Given a set of j sampled (or planned/desired) force vectors P_i at locations x_i , the task of a planner is to find the control vector c , which minimizes the error e given by

$$e = \sum_{i=1}^j [F(x_i) - P_i]^2 \quad (6)$$

Depending on the number of samples (j), a minimum norm, an exact, or a least squares approximate solution can be found.

Applying the Mussa-Ivaldi vector field work to the motor system in the nonredundant limb allows approximation of arbitrary smooth vector fields. However, it follows that the planning and choice of the control fields to be approximated must be obtained in some other manner.

Serially redundant manipulators might pose difficulty for the use of this class of models in many biological systems. A study of how far the mechanism of basis field summation might apply to serially redundant planar linkages Mussa-Ivaldi and Gandolfo suggested that in serially redundant linkages a close approximation to vector summation may occur among a large fraction of randomly chosen control primitives. Mussa-Ivaldi (1997) extended these analyses to human-like movement trajectory construction. Use of force-field construction for motion generation in a nonconvex work space can be used for obstacle avoidance and navigation (see Khatib et al., 1999, and POTENTIAL FIELDS AND NEURAL NETWORKS).

More recently, Kargo and Giszter (2000) have proposed on experimental grounds that the primitives should ideally be considered as viscoelastic fields of specific durations (therefore wavelet-like in behavior) and that the approximation problem for biological motor control can be formulated as

$$F(r, \dot{r}, t) = \sum_i A_i \cdot a(b_i \dot{r} + \tau_i) \cdot \Phi_i(r, \dot{r}) \quad (7)$$

where A_i are scalar amplitude parameters, parameters b_i control frequency of a fixed waveform given by function a , τ_i is a phase parameter, and $\Phi_i(r, \dot{r})$ are the viscoelastic force-field primitives.

Collections of primitives might be used in several ways. Planning and calculating how to combine primitives to approximate a desired field structure is one task that the CNS could perform for each behavior and perhaps each individual instance. In principle, arbitrary field structures and field time courses can be generated by the techniques of basis field approximation. However, as discussed, this flexibility implies that planning the details of these processes must be deferred to other mechanisms. In biological terms, collections of primitives could also be used in low-level motor behaviors driven by CPG oscillators, reflex stimuli, and simpler reinforcement and decision systems to elaborate simple protective actions and perhaps to bootstrap higher motor organization in mammals. Alternative models can be developed in this framework (Giszter et al., 2000b; see also Eliasmith and Anderson, 2000). Clearly, motor primitives could be used in each of these ways in biological systems.

Ideal primitives should provide specific stability guarantees in movement construction and execution. The biologically identified motor primitives simulate passive systems. When they are examined in viscoelastic terms, it is speculated that they may also represent contracting systems, which provide specific stability guarantees when combined in different arrangements (see Slotine and Lohmiller, 2001). The relationship of primitives to contracting systems awaits experimental testing.

It is also important to relate the motor primitives and elements discussed above to rhythmicity and CPGs (see Kiehn, Hounsgaard, and Sillar, 1997). Recently, Sternad et al. (2001) have strongly argued the need for use of limit cycle oscillating “primitives” in human and robot movement construction to synthesize timing. CPGs and limit cycle systems (see MOTOR PATTERN GENERATION) are a huge area that can be mentioned only briefly in this context. The relationship of timing or phasing systems to the force-field primitives discussed above is unclear in the biological data. The role of oscillation and timing in motor learning is clearly significant (Sternad et al., 2001). Kargo and Giszter (2000) have argued for a separation of rhythm and sequence generation from the force-field motor primitives used in execution. Several investigators are actively examining oscillating or “limit cycle primitives” and stability guarantees for these (e.g., Slotine and Lohmiller, 2001; Eliasmith and Anderson, 2000). It is possible that a hierarchy of different types of modules may form the basis of rapid motor development, flexible movement construction, and motor learning.

How force-field primitives are initially constructed and established in a motor system is an important issue that has also recently been considered. Todorov and Ghahramani have advanced the notion that the biological primitives that have been observed can be predicted as the result of a developmental process of system identification of a compact basis. The basis set that is obtained is able to jointly represent the limb dynamics and sensorimotor relationships in the neural control (unpublished work in progress; see also Mataric, Zordan, and Williamson, 1999, and Fod, Mataric, and Chadwicke Jenkins, 2002).

Motor Primitives in Real Behaviors: Biological Motor Control

Wiping Motor Pattern Construction and Control Using Force-Field Primitives

The task of wiping movements consists of locating a stimulus on the body surface and executing an action that removes it. The task involves many elements of sensorimotor behavior seen in more complex and voluntary acts. The animal subdivides the task as follows: (1) It moves together the body segment on which the target stimulus is located and the effector limb, closing the kinematic chain, and (2) it executes a movement that removes the irritant. These subtasks require the frog to transform skin location, limb configuration, and body scheme to (a) select an appropriate effector, (b) select a set of postures and limb trajectories for the effector, and (c) generate a set of appropriate muscle activations to control and move the effector through these postures and trajectories. Each stage involves ill-posed problems; that is, there is a set of many possible solutions to the problem, from which one must be selected.

A wiping strategy normally consists of a sequence of trajectories and postures. As an example, the kinematic postures and transitions (note, for example, placing and aiming) that are observed in wipes to the opposite hindlimb are summarized in Figure 1A. For some types of wipes, particular phases are optional and need not be executed during each cycle. Thus, in wiping to the back, extension may be omitted, flexion may be omitted, and whisk and flexion may be blended in both intact and spinal frogs (see Berkinblit, Feldman, and Fookson, 1986). It is clear that body scheme infor-

mation is used to control wiping. This frequently involves active posturing of target limbs. Initial experiments on force-field primitives suggested a linkage of force-field primitives and elements of reflex behaviors such as wiping (Giszter et al., 1993). This linkage is now much more firmly established. Both flexion withdrawal and wiping (Giszter et al., 2000b) have been shown to be composed of sequences and combinations of force-field primitives. In wiping, hip and knee extensor deletions in the motor pattern (which were first observed by Stein's group in turtle scratch; see SCRATCH REFLEX) represent deletions of specific force-field primitives (Kargo and Giszter, 2000). An example of kinematic corrections and the isometrically measured correction force-fields taking this form are redrawn from Giszter et al. (2000b) (based on original work in Kargo and Giszter, 2000) and shown in Figure 1.

Rapid online corrections of wiping trajectories (Figure 1B) can be shown to occur by inserting specific force-field primitives into the ongoing behavior as a result of obstacle collisions (Figure 1F; Kargo and Giszter, 2000). The fields without correction (Figure 1G) can be subtracted from those in Figure 1F (e.g., as in Figure 1E) and lead to a fixed structure field (Figure 1H). Setup and adjustment of the motor pattern and trajectory and response to muscle vibration can all be described as amplitude or phase modulation of force-field primitives that are combined by the spinal cord of spinalized frogs according to the rule in Equation 7 above with parameter b fixed for all primitives, that is,

$$F(r, \dot{r}, t) = \sum_i A_i \cdot a(t + \tau_i) \cdot \Phi_i(r, \dot{r}) \quad (8)$$

Descending Control and Force-Field Primitives

d'Avella and Bizzi have extended the spinal results to less reduced or intact preparations and have shown that a small basis set of muscle patterns and force patterns observed in the spinal frog represent a major fraction of the basis for vestibular correction (Bizzi et al., 2000) and intact behaviors.

Force-Field Primitives in Mammals

Force-field primitives of properties closely resembling the frog data are found in rat (Tresch and Bizzi, 1999) and cat, respectively. This framework in mammals suggests a new generation of functional electrical stimulation neuroprostheses stimulating either motor pools or interneuron pools to recruit single muscles or ensemble primitives (see Giszter et al., 2000b, for review). The relationship identified interneuron systems such as the C3–C4 and L3–L4 interneurons (found in the cervical [C3–C4] and lumbar [L3–L4] segments of mammalian spinal cord) remains to be explored.

Primitives in Voluntary Movement and in Motor Adaptation

Decomposition of human motor acts into a few summed elements has now been achieved by various laboratories (see publications of Sanger et al., cited in Kargo and Giszter, 2000). The use of the idea of motor primitives in describing effects in motor learning has gained support from the work of Thoroughman and Shadmehr and of Matsuoka and Bizzi (see Giszter et al., 2001a). How these elements used in learning and adaptation relate to spinal structures remains to be seen. The relationship of the primitives used to approximate and generalize the control of a trajectory in a novel environment and the primitives that are established as a basis set in reflexes in spinal cord of “lower” animals is clearly an important area. The higher-level elements of motor learning may represent more spatially localized patterns of recruitment, activation, and combination of the whole limb spinal primitives. However, it is also likely that completely novel primitives are elaborated in development and learning, for example, in motor cortex and cerebellum, to augment the spinal “bootstrap” or reflex behaviors and ex-

pand the domain of movement possibilities through life in a variety of ways (see Giszter et al., 2001a). How the construction and plasticity of primitives are organized in mental development and then later during the learning of novel motor acts or after trauma in adults is a fascinating and important area of investigation.

Road Maps: Motor Pattern Generators; Neuroethology and Evolution

Related Reading: Command Neurons and Command Systems; Equilibrium Point Hypothesis; Geometrical Principles in Motor Control; Locomotion, Vertebrate; Motor Pattern Generation; Potential Fields and Neural Networks; Radial Basis Function Networks; Scratch Reflex; Visuomotor Coordination in Frog and Toad; Visuomotor Coordination in Salamander

References

- Berkinblit, M. B., Feldman, A. G., and Fookson, O. I., 1986, Adaptability of innate motor patterns and motor control mechanisms, *Behav. Brain Sci.*, 9:585–638. ♦
- Bizzi, E., Tresch, M., Saltiel, P., and d'Avella, A., 2000, New perspectives on spinal motor systems, *Nature Rev. Neurosci.*, 1:101–108. ♦
- Eliasmith, C., and Anderson, C. H., 2000, Rethinking central pattern generators: A general approach, *Neurocomputing*, 32–33:735–740.
- Fod, A., Mataric, M. J., and Chadwicke Jenkins, O., 2002, Automated derivation of primitives for movement classification, *Autonomous Robots*, 12(1):39–54.
- Giszter, S. F., Grill, W., Lemay, M., Mushahwar, V., and Prochazka, A., 2000a, Intraspinal microstimulation: Techniques, perspectives and prospects for FES, in *Neural Prostheses for Restoration of Sensory and Motor Function* (K. A. Moxon and J. K. Chapin, Eds.), Boca Raton, FL: CRC Press, pp. 101–138. ♦
- Giszter, S. F., Moxon, K. A., Rybak, I., and Chapin, J. K., 2000b, A neurobiological perspective on design of humanoid robots and their components, *IEEE Intelligent Systems*, 15(4):64–69. ♦
- Giszter, S. F., Moxon, K. A., Rybak, I., and Chapin, J. K., 2001a, Neurobiological and neurobotic approaches to design of a controller for a humanoid motor system, *Robotics and Autonomous Systems*, 37:219–235. ♦
- Giszter, S. F., Mussa-Ivaldi, F. A., and Bizzi, E., 1993, Convergent force field organized in the frog's spinal cord, *J. Neurosci.*, 13:467–491.
- Giszter, S. F., Loeb, E., Mussa-Ivaldi, F. A., and Bizzi, E., 2001b, Repeatable spatial maps of a few force and joint torque patterns elicited by microstimulation applied throughout the lumbar spinal cord of the spinal frog, *Human Movement Sci.*, 19:597–626.
- Kargo, W. J., and Giszter, S. F., 2000, Rapid corrections of aimed movements by combination of force-field primitives, *J. Neurosci.*, 20:409–426.
- Khatib, O., Yokoi, K., Brock, O., Chang, K., Casal, A., 1999, Robots in human environments: Basic autonomous capabilities, *Int. J. Robotics Res.*, 18(7):684–696.
- Kiehn, O., Hounsgaard, J., and Sillar, K., 1997, Basic building blocks of vertebrate spinal central pattern generators, in *Neurons, Networks, and Motor Behavior* (P. Stein, S. Grillner, A. Selverston, and D. Stuart, Eds.), Cambridge, MA: MIT Press, pp. 47–60. ♦
- Mataric, M. J., Zordan, V. B., and Williamson, M. M., 1999, Making complex articulated agents dance: An analysis of control methods drawn from robotics, animation, and biology, *Autonomous Agents and Multi-Agent Systems*, 2(1):23–44.
- Mussa-Ivaldi, F. A., 1997, Nonlinear force fields: A distributed system of control primitives for representing and learning movements, *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 84–90.
- Schaal, S., 1999, Is imitation learning the route to humanoid robots?, *Trends Cogn. Sci.*, 3(6):233–242. ♦
- Slotine, J. J., and Lohmiller, W., 2001, Modularity, evolution, and the binding problem: A view from stability theory, *Neural Networks*, 14(2):137–145.
- Sternad, D., Duarte, M., Katsumata, H., and Schaal, S., 2001, Dynamics of a bouncing ball in human performance, *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 63(1, Pt 1):011902.
- Tresch, M. C., and Bizzi, E., 1999, Responses to spinal microstimulation in the chronically spinalized rat and their relationship to spinal systems activated by low threshold cutaneous stimulation, *Exp. Brain Res.*, 129:401–416.

Motor Theories of Perception

Carol A. Fowler, Bruno Galantucci, and Elliot Saltzman

Introduction

Motor theories of perception propose that there is recruitment of the motor system or of motor competence (i.e., knowledge) in perception. Perhaps the best known motor theory of perception is Liberman's motor theory of speech perception (see Liberman, 1996, for a history and overview of the motor theory). Within speech science, despite its prominence, the theory has been judged implausible on several grounds. However, in the larger field encompassing studies of perception, action, and their coupling, it is given more credence. It is instructive to consider why the judgments differ between speech experts and experts in the broader domain.

In the following, we outline the motor theory of speech perception and describe some of the findings underlying its development. Next we offer reasons why speech scientists have doubted especially one of its two central claims, namely, that the speech motor system participates in speech perception. Then we suggest why the reasons are not sufficient to refute the claim, and we show that it acquires credibility when it is set in the larger context of investigations of perception, action, and their coupling. In addition, we summarize research that suggests a neural system consistent with Liberman's largely undeveloped ideas about neural support for speech perception. The discovery of mirror neurons in primates (Rizzolatti and Arbib, 1998) provides an existence proof of neuronal perceptuomotor couplings.

The Motor Theory of Speech Perception

Although in alphabetic script, consonants and vowels are discrete, their expression in acoustic speech signals is not. This is because speakers coarticulate speech gestures; that is, they produce the articulatory gestures of successive consonants and vowels in a temporally and spatially overlapping manner. Gestures are linguistically significant actions of the vocal tract. More specifically, they are equivalence classes of articulatory patterns controlled with respect to linguistically significant goals defined in an abstract task space. Consequences of coarticulation are evident, for example, during production of the word *to* when activation of the vowel /u/ 's lip protrusion gesture overlaps the activation of the consonant /t/'s lingual gesture (compare *tea*). Due to coarticulation, acoustic speech signals are highly context sensitive, and they lack a discrete segmental structure.

Liberman developed a motor theory of speech perception when he and his colleagues found that speech percepts track articulation more closely than the acoustic signals to which articulation gives rise. Two experimental findings were especially telling. One was that, in the synthesized syllables /di/ and /du/, the critical acoustic cues for /d/ were quite different, owing to the effects of coarticulation by the different vowels. Indeed, the cues were audibly distinct when presented in isolation to listeners. However, the gestures for /d/ are the same in natural productions of the two syllables, and the consonants sound alike. A complementary finding was that the same acoustic cue was identified as /p/ before /i/ and /u/, but as /k/ before /a/. Because of coarticulation, to generate the cue before /i/ or /u/ requires production of /p/, whereas to generate it before /a/ requires production of /k/.

Both findings suggested to Liberman that when articulation and acoustic patterns diverge due to coarticulation, perception tracks articulation, a central claim of the motor theory. Subsequently, many other findings (see Liberman, 1996, for a review) converged

on the same conclusion. A notable one is the McGurk effect (McGurk and MacDonald, 1976), in which a video of a speaker mouthing one syllable, say, /da/, is dubbed with a different acoustic syllable, say, /ma/. Listeners hear a syllable (/na/ in the example) that reflects integration of gestural information from both modalities.

In Liberman's view, recovery of articulation in speech perception implies recruitment of the motor system. Such motor recruitment is required because of coarticulation in speech production. Speech information must be transmitted rapidly, and the gestural overlap provided by coarticulation permits efficient packaging of consonants and vowels. However, coarticulation has other consequences, including context sensitivity in acoustic information for phonetic segments. Therefore, two specializations, one for coarticulating and one for perceiving coarticulated speech, are needed, and, because neither specialization is useful without the other, they had to coevolve. Moreover, given the motor character of the percept, and Liberman's view that this reflects recruitment of the motor system in perception, the inference was plausible that the specializations were one and the same: a phonetic module. By using gestures as a common currency for talkers and listeners, the module helps guarantee achievement of parity between them—that is, sufficient equivalence between phonological messages sent and received, a necessity for successful communication.

Speech Science: The Implausibility of the Motor Theory

Following are grounds on which the motor theory of speech perception has been judged implausible, and then some reasons why we reject each argument.

1. Many speech scientists (e.g., Ohala, 1996) deny that speech percepts have a motor character, and they have no other reasons to suppose that the speech motor system is involved in perceiving speech.
2. Liberman and colleagues wrote very little about how the speech motor system might participate in speech perception, and the mechanism that they typically alluded to (analysis by synthesis) is not obviously workable at the rates at which consonants and vowels are perceived.
3. Listeners' perception of speech gestures need not imply that the speech motor system is recruited in speech perception. This is because the acoustic signal, having been caused by the gestures, and taking distinctive forms for distinct gestures, provides information about them. Listeners perceive gestures because that is what the information in acoustic speech signals is about.

We will address the first objection here only by remarking that, in our opinion as in Liberman's, evidence in favor of perceiving motor gestures is substantial and unrefuted. For example, we know of no studies that refute the evidence we cited in the previous section in favor of the claim that, when articulation and acoustic patterns diverge, perception tracks articulation. As for the other two objections, however accurate they may be, neither refutes the motor theory's claim of motor system recruitment in speech perception. As for the second objection, even if the particular mechanisms proposed by Liberman and colleagues are not the ones that support speech perception, it does not follow that no mechanism involving

a production-perception link does the job. As for the last objection, even though acoustic speech signals provide information about speech gestures, that does not preclude a perceptual mechanism in which the speech motor system or motor competence participates in decoding the acoustic signal.

The Broader Scientific Field: The Necessity for Motor Theories?

In the broader scientific field, central theoretical ideas of Liberman's motor theory recur (e.g., Viviani and Stucchi, 1992), and there are research findings suggesting motor involvement in perception. We review one example of a theoretical view that shares critical ideas with those of Liberman and then summarize a few of the research findings.

A Related Idea

Prinz (e.g., 1997) addresses an issue that arises in the study of perceptually guided action and that is very much like the one we have labeled parity. In speech, *parity* refers to the relation between messages sent and perceived. The messages must characteristically be the same; otherwise communication fails. Prinz has raised the same issue in asking how perception can guide action under the common assumption that percepts are representations of sensory information, and planned actions are coded in purely motor terms; that is, they lack a common currency. He proposes instead that percepts and actions share a common code. Further, consistent with the motor theory's identification of gestures as the common currency of talkers and listeners, and with the hypothesis that gestures are represented in the task spaces of talkers and listeners (SPEECH PRODUCTION), Prinz's *action effect principle* invokes a common code that represents not the proximal stimulus, but the relevant distal event properties. (*Proximal* refers to the signals that stimulate the sense organs, whereas *distal* refers to the environmental events that causally structure the proximal stimuli). Prinz's research shows, for example, that when stimuli that guide responses in some tasks share distal features with responses, response times are affected.

Research Findings

The larger context of evidence, to which we alluded earlier, in which the motor theory gains plausibility includes evidence from communication systems of other animals, evidence of motor recruitment in perception of motion, and findings of mirror neurons. In each domain, we provide illustrative examples.

Communication systems of other animals. Male crickets produce mating calls to attract females. Females respond to the calls by moving toward the male, but they do not produce calls themselves. However, males and females show a remarkable symmetry. Different varieties of crickets produce different calls, and females prefer the calls of their own type. When crickets are hybridized by mating the male of one type to the female of another, the male's call exhibits components from the calls of both parental types. Remarkably, female hybrids prefer the hybrid call to the call of either parental type (Hoy, Hahn, and Paul, 1977). This suggests a genetic correspondence between neural systems supporting call production in males and call perception in females.

Evidence of perception-action coupling can be found within individuals as well as between them. In zebra finches, the neural

system supporting call production also responds to components of auditorily presented songs (Williams and Nottebohm, 1985). A major path for song production in the zebra finch brain begins at a "higher vocal center" (HVC), which projects to the robustus archistriatum (RA) and from there to the tracheosyringeal portion of the hypoglossal nerve (nXIIts). nXIIts innervates the muscles of the syrinx. The HVC and nXIIts both respond to tone bursts, and motor neurons in nXIIts are differentially responsive to different components of perceived songs. Hauser (1996) concludes that "in order for birds to perceive the proper acoustic features of a song syllable, the percept must be converted into a series of motor actions required to produce the sound" (pp. 148–149).

Evidence in humans for motor recruitment in perception outside the speech domain. The tangential velocity of curved movements made by humans is proportional to curvature according to a two-thirds power law, decreasing with increases in curvature (e.g., Viviani and Stucchi, 1989). Viviani and Stucchi have shown that observers' judgments of the shapes of ellipses being drawn on a computer screen (judgments as to whether the major axis is oriented vertically or horizontally) are affected not only by the form's shape, but also by its velocity profile. When ellipses were drawn with constant velocity—a profile characteristic, in natural drawing, of a circular form—perceivers' judgments were poor. Tracings of ellipses that adhered to the two-thirds power law were judged accurately. An implicit proprioceptive-motor, rather than visual, task (Viviani, Baud-Bovy, and Redolfi, 1997) provided similar results. Blindfolded participants' right arms were moved in elliptical trajectories that did or did not preserve the two-thirds power law. With the left arm, participants tried to reproduce the movement of the right arm. Shapes of reproduced trajectories were more accurate when ellipses traced by the right arm conformed to the two-thirds power law than when they did not. Together, these data show that motor competence, here knowledge about velocity constraints on biological movements, is brought to bear on perception of motion.

Other evidence for linkages between the motor and perceptual systems comes from experiments that manipulate the similarity between a stimulus-response pair and measure its facilitatory or inhibitory influence on motor performance (see Prinz, 1997, for a review). For example, Stürmer, Aschersleben, and Prinz (2000) had participants produce a grasping gesture (first close the hand, then open it) or a spreading gesture (first open then close). The task-relevant stimuli for the movements were color changes on a hand that was displayed on a computer monitor, with different colors signaling each task. The visible hand also produced a task-irrelevant gesture on each trial, starting and ending from a neutral half-open position. In one case, it closed and then opened; in the other, it opened and then closed. Although participants were told to ignore the irrelevant information, selecting their responses only on the basis of the color change, their response latencies were faster when their movements matched the irrelevant ones. That perception of a hand gesture interacts with the execution of a similar or dissimilar hand gesture provides strong evidence that the perceptual and the motor systems share a common currency.

Mirror neurons. The foregoing evidence, like the evidence underlying the motor theory of speech perception, suggests access to the motor system or to motor knowledge in perception. Recent findings of mirror neurons may reveal part of a neural mechanism that permits and promotes such access.

Rizzolatti and colleagues (see Rizzolatti and Arbib's, 1998 review) have found neurons in the premotor cortex of the monkey (area F5) that respond both when the monkey performs a given action and when it perceives a similar action performed by another monkey or by a human. Many mirror neurons are quite specific in

firing during the performance of, say, one manual grasping movement but not another. Many of them exhibit the same specificity in the observed actions that stimulate them to fire.

There is evidence for mirror neurons in humans. Fadiga et al. (1995) used transcranial magnetic stimulation (TMS) of the motor area, in which stimulation provoked muscle activity in the fingers. During TMS, participants observed several events or situations: someone grasping an object, the stationary object itself, someone tracing shapes in the air with the arm, or the dimming of a light. The investigators found more TMS-induced muscle activity in the fingers when participants were observing grasping than when they were observing any other of the events. The modulation of muscle activity was specific to the actions observed. Fadiga et al. concluded that "in humans there is a neural system matching action observation and action execution" (p. 2609).

The finding of mirror neurons reveals neural systems that underlie perception-action coupling in monkeys and perhaps in humans as well. From the perspective of the motor theory of speech perception, it is intriguing that the neurons were found in an area of the monkey brain that includes the homologue of Broca's area in humans, which is involved in language use. The findings, therefore, lend credence to the motor theory's claim of a production-perception coupling in speech.

Discussion

The motor theory of speech perception has inspired analogous theories in other domains. Yet the theory was motivated by requirements of speaking and listening that Liberman considered special to speech. Is speech special in respects that should discourage efforts to generalize some of its proposals to other domains? We suspect not.

As we have noted, talkers and listeners must characteristically achieve parity to communicate successfully (Liberman, 1996), and parity achievement requires use of a common currency by talkers and listeners. That the speech percept has a motor character suggests that this common currency is defined in gestural task space: listeners *detect* the proximal acoustic signal, but they *perceive* the distal gestural activities of talkers. According to the motor theory, gesture perception is fostered by motor recruitment in perception.

This is not very different from what is required for successful nonlinguistic transactions with the environment, including those with other actors. Although proximal energy patterns stimulate the sense organs, animals must perceive the distal possibilities afforded for action (e.g., Gibson, 1979; see also GRASPING MOVEMENTS: VISUOMOTOR TRANSFORMATIONS). For actions to be felicitous, parity is required among perceived possibilities for action, real possibilities for action, and action itself. This is real-world, functional perception-action coupling. Plausibly, neural-motor recruitment in

perception fosters achievement of these parities as well as those of linguistic communication.

In much cognitive science research, perception and action are assumed to be sufficiently distinct and autonomous that they can be studied independently. However, consideration of the relations between animals and their environments uncovers no principled way to draw such a sharp distinction. Perception-action couplings are central to the design of animals. Understanding the real-world settings in which cognitive activity occurs reveals that it could not be otherwise.

Road Map: Linguistics and Speech Processing

Related Reading: Language Evolution: The Mirror System Hypothesis; Language Processing; Optimality Theory in Linguistics; Speech Production

References

- Fadiga, L., Fogassi, L., Pavesi, G., and Rizzolatti, G., 1995, Motor facilitation during action observation: A magnetic stimulation study, *J. Neurophysiol.*, 7:2608–2611.
- Gibson, J. J., 1979, *The Ecological Approach to Visual Perception*, Boston: Houghton Mifflin. ♦
- Hauser, M., 1996, *The Evolution of Communication*, Cambridge, MA: MIT Press. ♦
- Hoy, R., Hahn, J., and Paul, R., 1977, Hybrid cricket auditory behavior: Evidence for genetic coupling in animal communication, *Science*, 195:82–84.
- Liberman, A. M., 1996, *Speech: A Special Code*, Cambridge, MA: MIT Press. ♦
- McGurk, H., and MacDonald, J., 1976, Hearing lips and seeing voices, *Nature*, 264:746–748.
- Ohala, J., 1996, Listeners hear sounds not tongues, *J. Acoust. Soc. Am.*, 99:1718–1728.
- Prinz, W., 1997, Perception and action planning, *Eur. J. Cogn. Psychol.*, 9:129–154. ♦
- Rizzolatti, G., and Arbib, M., 1998, Language in our grasp, *Trends Neurosci.*, 21:188–194. ♦
- Stürmer, B., Aschersleben, G., and Prinz, W., 2000, Correspondence effect with manual gestures and postures: A study of imitation, *J. Exp. Psychol. Hum. Percept. Perform.*, 26:1746–1759.
- Viviani, P., Baud-Bovy, G., and Redolfi, M., 1997, Perceiving and tracking kinesthetic stimuli: Further evidence of motor-perceptual interactions, *J. Exp. Psychol. Hum. Percept. Perform.*, 23:1232–1252.
- Viviani, P., and Stucchi, N., 1989, The effect of movement velocity on form perception: Geometrical illusions in dynamic display, *Percept. Psychophys.*, 46:266–274.
- Viviani, P. and Stucchi, N., 1992, Motor-perceptual interactions, in *Tutorials in Motor Behavior II* (G. Stelmach and J. Requin, Eds.), Amsterdam: North Holland.
- Williams, H., and Nottebohm, F., 1985, Auditory responses in avian vocal motor neurons: A motor theory for song perception in birds, *Science*, 229:279–282.

Multiagent Systems

José M. Vidal and Edmund H. Durfee

Introduction

Natural systems are based on parallel, distributed processing at many different levels. At the neural level, interconnected and concurrently acting neurons propagate signals among themselves such that coherent behavior emerges from their joint activity. Within the

brain, different neural subsystems combine and exchange signals to work together to yield an overall intelligent nervous system. Beyond the boundaries of a single intelligent entity are other such entities, which together make up societies that achieve more than the individual entities can. Thus, what constitutes an "individual" can be highly subjective: what is an individual to one researcher

may, to another, be a complex distributed system comprised of finer-grained agents. In this context, an agent's "granularity" corresponds to the amount of processing it does between interactions with others.

Research in artificial neural networks (ANNs) has concentrated on the finer-grained levels of intelligence evidenced in the brain, drawing on neurophysiology and psychology as sources of inspiration when trying to build computational models of such natural systems. Research in brain theory has dealt with different levels, from neurons to brain regions to humans. Finally, research in multiagent systems has focused on coarse-grained levels of individuality and interaction, where the goal is to draw on sociological, political, and economic insights to develop multiagent systems composed of autonomous interacting agents (Weiss, 1999). As such, there is ample room for comparison between the ANN, brain theoretic, and multiagent systems approaches. Much can be gained by comparing the different techniques used in these fields of study. Specifically, agents in multiagent systems must interact with each other. This interaction is often facilitated by the use of agent models. That is, agents either have or learn models of the agents with which they interact. Agents that learn models of other agents can do so by observing their past behavior. These models allow agents to avoid dealing with malicious or broken agents. In a system with many learning agents, we can expect agents to build nested models of the other agents—that is, models that include an agent's models of other agents, and so on. We might expect agents to build deeper and deeper nested models of each other in an effort to outguess each other. However, while there is some amount of deep-model escalation—agents building nested models of each other to greater and greater depths—some research (Vidal and Durfee, 1998) shows that these deeper models exhibit decreasing returns. By using their models of each other, the agents loosely organize themselves into self-reinforcing communities of trust.

Learning agents can use any type of learning, including ANNs. Although there has been very little research into this topic, ANNs seem like a promising learning technique because of their flexibility and ability to generalize. An agent that uses ANNs could learn about the "tags" that identify a cheating agent by extrapolating from a few examples. This would allow the agent to avoid unproductive future interactions with other agents that it has never met. Such behavior, when enacted by all the agents in the system, would speed up system convergence and discourage cheating agents more rapidly. Furthermore, the dynamics generated by a system composed of many learning agents are not unlike the dynamics of an ANN. Some of the same problems of credit assignment and propagation of rewards arise in both of these systems, even if the protocol details are different.

Brain theory can be related to multiagent systems at many levels. At one level brain theory considers neurons as agents. That is, it considers them as a basic behavior unit. Unlike agents in a system, neurons do not engage in complex interactions with other neurons. However, neurons have predefined rules of encounter. For example, a neuron can "count on" the signals of a particular input line carrying specific information. These simple rules enable coordination of the system of neurons. Similar rules are often used by multiagent systems to achieve coordination. At a higher level, brain theory considers brain regions as agents. These agents are more complex and might start to derive a benefit by considering other brain regions as agents and trying to build models of them or trying to interact with them using multiagent techniques. At an even higher level, brain theory considers people as agents. At this level we might wonder if the theories developed for the construction and control of artificial multiagent systems might be useful in either explaining or guiding collective human behaviors.

We can also contrast the various abstraction levels in brain theory to the two organizational varieties in multiagent systems: top-

down and bottom-up. In a top-down approach, one or a few agents monitor the global performance of the organization and detect when a particular organizational structure is needed. Most typically, roles in the new organization are assigned (often through a contracting style of protocol) and the new organization is adopted (Corkill and Lesser, 1983). Alternatively, in a bottom-up approach, an agent monitors its own performance, decides when a change (such as to the tasks it itself performs) is in order, and unilaterally makes that change. This in turn could cause other agents to change what they are doing, and reorganization propagates throughout the network (Ishida, Gasser, and Yokoo, 1992). These techniques are similar to the brain theory studies of bottom-up organizational emergence as seen in neurons and the top-down organizational rules as developed and imposed by humans in their organizations.

In this article we will begin by categorizing the different types of agent interaction protocols that are being studied in multiagent systems, from cooperative problem solving among cooperative agents to coordination protocols for autonomous selfish agents. These will be explained in detail in later sections of this article, including brief overviews of the latest research in these areas. We conclude this article by summarizing some of the similarities and differences between multiagent systems and the fields of brain theory and neural networks, highlighting some potentially important areas for cross-fertilization between them.

Agent Coordination

A central concern in distributed artificial intelligence (DAI) is the development of interaction protocols for the coordination of agents. System designers build protocols that determine how, when, and what the agents will communicate to each other. A good protocol should achieve the expected global goals while maintaining some independence between agents. The degree of independence depends on the particular application. For example, if the agents are robots in a large field, then we can expect them not to interfere with each other as long as they keep their distance. However, if these same robots are charged with the task of lifting a large object, then much tighter coordination will be needed. The amount of independence depends on the type of tasks the agents must achieve and on their interaction opportunities within the environment.

Agents often use commitments and conventions in order to coordinate (Jennings, 1994). *Commitments* are pledges from an agent that it will try to achieve a specified task. Commitments help other agents plan their actions without interfering with each other. *Conventions* specify when commitments can be broken and what the agents should do when this happens. In a typical system we can expect that unforeseen circumstances might force an agent to drop certain commitments, especially if the world the agent inhabits is changing. Conventions tell the agent how it can renege on its commitments without causing undue distress to the system.

DAI systems can be roughly separated into two categories: systems in which all agents share a common goal, and systems in which each agent has its own goals. When all agents have the same goal, the interaction protocol enables balanced task decomposition and allocation. That is, the interaction protocol divides and distributes the tasks among the agents so as to avoid overloading any resource or agent, handing agents overlapping or conflicting tasks, and leaving tasks unhandled. When all the agents have different goals, then a coordination protocol must be designed to align their interests with the designer's global system goal. We discuss these types of systems, and their protocols, in the next sections.

Cooperative Problem-Solving Systems

In a cooperative distributed problem-solving system (Bond and Gasser, 1988), agents work together to solve problems that require them to cooperate. Research in this area takes two major forms.

One of these forms is represented by the functionally accurate/cooperative (FAC) paradigm (Corkill and Lesser, 1983), where problem solvers individually solve their local subproblems and share their partial results so that, over time, increasingly complete results get formed and the system solves the entire problem. Success is achieved when one or more agents construct and share a hypothesis that satisfies the solution criteria. Of course, the wholesale exchange of all partial results can bog down a system quickly, and so substantial effort has gone into the development of techniques by which cooperating agents could model each other to be smart about which results to share. These models have included organizational structures (Corkill and Lesser, 1983), partial global plans (Durfee and Lesser, 1991), and the TAEMS framework (Decker and Lesser, 1995). Another approach is to include team considerations from the beginning, which has led to research into team-oriented programming (Tambe and Zhang, 1998).

A complementary perspective on cooperative problem solving has viewed the process as sharing tasks rather than results. As embodied in the Contract-Net protocol (Smith, 1980), the task-sharing perspective sees the coordination problem as associating tasks to be done with the right agents to do them. In Contract-Net, for example, an agent with a large task to do would decompose it into smaller tasks, and then attempt to contract these out to the most suitable agents by announcing each subtask, collecting bids, and awarding the subtask. Because agents choose to bid on the tasks they receive, the assignment of tasks to agents involves mutual selection. A similar specialization occurs in the evolutionary structuring of the brain into specialized regions. Task sharing can also be coupled with result sharing, such that organizational roles could be contracted out, FAC would follow, and then tasks to implement the solution could be contracted out.

Continuing research in cooperative distributed problem solving has served to extend and refine the basic paradigms of task and result sharing, involving, among other things, the introduction of planning to the process, and the formulation of negotiation strategies for reaching a compromise between what different agents want that maximizes the overall system performance. Note, however, that in this discussion the emphasis has been on the overall system rather than an individual. The assumption is that agents in these systems are built with an implicit goal of doing whatever they need in order to improve the performance of the entire system. Although this assumption is often reasonable when building real systems, since we can design such agents, it does neglect the issue of how to get individual agents, each with its own selfish goals, to solve problems cooperatively. This issue has been the focus of a variety of research activities more recently, as the next section explains.

Selfish Agents

Cooperative problem solving has been shown to be generally beneficial from a systemwide perspective, but many DAI researchers are concerned with how such cooperation might emerge when an agent can only see benefits to itself. In other words, what is the knowledge and reasoning that agents employ in making smart decisions about when to work together? Once they have adopted the goal of working together, then they can employ cooperative problem solving techniques to actually accomplish their joint activities.

One of the most fruitful fields for insights when investigating how selfish agents can still work together for their mutual benefit is economics. In fact, the use of economic techniques for coordinating multiagent systems is a very active area of research. The techniques used include voting, bargaining, and auctions (Weiss, 1999, chap. 9). Each technique has different strengths and weaknesses. They are all concerned with the making of rational decisions by a group of distributed agents, each with its own goals. These techniques usually involve several communication steps as

well as rational decision making by the agents. They are very useful techniques for structuring the interactions between agents because they aggregate the individual agents' desires to form a global decision or allocation.

Auctions, for example, are very efficient methods for matching buyer and seller agents that are interested in the same good or service. The use of a single exchange currency allows an agent to externalize its utility with a single number that is understood by all other agents. If there are enough agents participating in the auction, then both buyers and sellers are generally guaranteed a fair price. Finally, the auction acts as a location service where all agents that are interested in the item can find each other and interact using the auction's bidding protocol, thereby eliminating the need for agents to find and try to talk directly to each other.

There are, however, some problems with auctions that echo the problems studied by economists. For example, just like their human counterparts, intelligent agents might be able to collude with each other and manipulate the auction's price to their benefit. This problem can be circumvented by changing the type of auction or by increasing the number of participants. A lying auctioneer who does not assign the winnings to the correct agent might also break an auction. Finally, auctions assume that all agents are trading exact instances of the same item. That is, they do not consider the possibility that one agent's products are of a higher quality than another agent's products. Economists often ignore minor differences in the quality of a good or, if the difference is great enough, they state that the different quality goods should simply be considered different goods altogether.

Even in the presence of agents that cheat or offer different quality goods, a stable system can often be achieved by allowing agents to use models of other agents. For example, if the agents see each other as rational, then they can make a deal allowing them to cooperate (Rosenschein and Genesereth, 1985). Alternatively, they can evaluate the performance of alternative strategies to select the cooperative strategy from a purely selfish view (Gmytrasiewicz, Durfee, and Wehe, 1991), leading to self-organization for their mutual benefit. Finally, they can use learning techniques to build models of other agents (Vidal and Durfee, 1998) and use these models to guide interactions.

Organization theory (Malone, 1987) has also looked at the problem of how organizations form and why individuals are willing to become parts of an organization. In joining an organization, an individual forgoes some of his freedom because he has now committed to performing some set of actions for others in the organization. He has also accepted dependencies on others. The positive side of the resulting web of commitments (Gasser, 1991) is that, when the agents abide by their responsibilities and while the circumstances under which the organization was formed hold, the payoff to members of the successful organization is higher than those members could have gained individually. So organization, under the right circumstances, is the rational thing to do. The remaining challenge, then, is in developing computational mechanisms by which individuals can detect that the circumstances are right for organizing and for performing the organizational self-design.

Discussion

Given the information above, and the larger context in which this article is written, we now wish to delve deeper into the relation between multiagent systems, brain theory, and ANNs. They differ in many respects. One respect is clearly the granularity of computation relative to communication. Multiagent systems generally assume that communication is time consuming, error prone, and costly. Thus, communication decisions are made judiciously, and

messages among entities in multiagent systems are at the symbol rather than signal level, encoding much richer semantic content.

Because communication is at such a premium, and because envisioning the impact of a message requires a model of the hearer of that message, agents in multiagent systems usually have explicit models of other agents, including their interests, abilities, and expectations. Decision making in agents is thus a complex process of mapping potential actions (including communication acts) into explicit models of the anticipated activities of others, leading to a wide range of behaviors. Often, to anticipate the actions of another, an agent will execute its inferencing processes on its model of that other agent, drawing conclusions about what the other agent could be thinking or doing by "putting itself in the other's shoes."

In brain theory, this implies that an agent should be able to ignore its current state and instead to "think" as if it were another agent. Because such projection requires an agent to be able to disconnect from its own reality and superimpose that assumed of another, this implies that models of the mind must provide a higher-level override capability to control the processing of the parallel brain regions and schemas. In this way, the same machinery that an agent uses to control its own actions can be used to predict those of others.

In ANNs the individual units are generally simpler, signal-processing elements, and the complexities arise from the sheer number of interconnections and the ways that those interconnections evolve over time. In multiagent systems, the primary concern has been to endow agents with knowledge about protocols, conventions, common goals, and so on right from the start, so that they can immediately interact efficiently; interaction is usually too expensive and slow to depend on "learning" how to interact. In many approaches to ANNs, the focus is precisely on learning, because interaction is cheap and fast in the tightly coupled, signal-propagating architecture assumed. Thus, in ANNs, important performance criteria include trainability, convergence, and robustness. Multiagent systems share the robustness criterion, but rather than the capability to learn, they focus on performance in terms of correctly and quickly coordinating activity to work as an effective team from the outset, making maximum use of resources at all times.

For example, having mobile robots learn to coordinate their movements by allowing them to learn from colliding is generally infeasible, since collisions can lead to disablement of the robots. In a multiagent approach (Montgomery and Durfee, 1993), the robots each plan their behaviors and then engage in a dialogue to efficiently isolate and resolve conflicting actions. Following a predominant paradigm in AI, the robots essentially engage in a distributed search through the space of joint behaviors, using a hierarchical representation to focus their search. Through the distributed hierarchical search, the agents can balance the costs of coordination with its benefits, to attain an appropriate level of coordination. As problems scale up, moreover, the agents can employ abstractions to represent teams of agents as single entities. Mobile robots performing deliveries, for example, are clustered into geographic teams such that a robot models only the other members of its team, and coordinates with other teams through the decisions of team leaders. Properties of the task and of the agents, known ahead of time by the agents, dictate appropriate decomposition and task-abstraction strategies that, in the best case, can reduce the time to solution from exponential in the single-agent case to logarithmic in the multiagent case.

However, even thought agents are usually endowed with the needed interaction protocols, it is often the case that these protocols leave the agent with many possible choices. For example, the fact that agents must interact using a Contract-Net protocol says nothing about which tasks the agents should bid on, or how much they should bid. In fact, the correct choice might depend on the current state of the system and of the other agents. These problems have

led to the study of learning agents within multiagent systems. As mentioned earlier, this is an area where multiagent systems and ANNs share some similarities. Both of them try to understand the behavior of systems composed of adaptive agents. In the multiagent case the agents are more complex and loosely connected, whereas in ANNs the units are simpler and have fixed connections but can interact much more often. Still, many of the same mathematical tools used to understand the emergent behaviors in ANNs could be used to understand and engineer multiagent systems.

Another critical similarity between multiagent systems and ANNs is an emphasis on emerging intelligence—on the whole being more than the sum of its parts. From relatively simple neural units, complex patterns of activity can arise in neural networks; from cooperation and competition among schemas, intelligent behavior results; from rational choices among individuals, societies and civilizations (to anthropomorphize) emerge. As was alluded to at the outset of this article, all systems are distributed if you look closely enough, and it is the fact that the collection is more than the sum of its parts that allows us to call it a system rather than a collection of component parts. Whether simple or complex, communicating signals or symbols, distributed systems are a ubiquitous framework encompassing all of these studies. As such, these studies have common ground for sharing ideas and insights.

As a specific example of the opportunities in a cross-disciplinary study, a key concern in multiagent systems, as in ANNs, is in how to propagate global feedback such that individual entities can modify their behavior correctly. Multiagent systems suffer from the same credit/blame assignment problems as neural networks and schema systems. If the system as a whole performs well or poorly, which entities, and which interactions among entities, were responsible?

Propagation algorithms that have been developed for neural networks, multiagent systems composed of learning agents, and feedback loops that support reorganization in multiagent systems raise many common concerns. So far, the similarities and potential overlaps between the multiagent systems and ANN algorithms have not been widely studied.

Similarly, multiagent systems, brain theory, and ANNs are all concerned with timely and appropriate communication among computational units. The thresholding computations used in neural networks and the cooperative/competitive links among brain regions have analogues in multiagent systems that decide when a result is good enough to share, and how much to believe results received from others. Such analogies are, to date, not firmly understood, and represent opportunities for cross-fertilization.

Road Map: Artificial Intelligence

Related Reading: Artificial Intelligence and Neural Networks; Competitive Queuing for Planning and Serial Performance; Decision Support Systems and Expert Systems; Hybrid Connectionist/Symbolic Systems; Schema Theory; Speech Recognition Technology

References

- Bond, A. H., and Gasser, L., Eds., 1988, *Readings in Distributed Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann. ♦
- Corkill, D. D., and Lesser, V. R., 1983, The use of meta-level control for coordination in a distributed problem solving network, in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, San Francisco: Morgan Kaufmann, pp. 748–756.
- Decker, K., and Lesser, V. R., 1995, Designing a family of coordination mechanism, in *Proceedings of the First International Conference on Multi-Agent Systems*, Cambridge, MA: AAAI Press, pp. 73–80.
- Durfee, E. H., and Lesser, V. R., 1991, Partial global planning: A coordination framework for distributed hypothesis formation, *IEEE Trans. Syst. Man. Cybern.*, 21:1167–1183.
- Gasser, L., 1991, Social conceptions of knowledge and action: DAI foundations and open systems semantics, *Artif. Intell.*, 47:107–138.

- Gmytrasiewicz, P. J., Durfee, E. H., and Wehe, D. K., 1991, A decision-theoretic approach to coordinating multiagent interactions, in *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, San Francisco: Morgan Kaufmann, pp. 62–68.
- Ishida, T., Gasser, L., and Yokoo, M., 1992, Organization self-design of distributed production systems, *IEEE Trans. Knowledge Data Eng.*, 4:123–134.
- Jennings, N. U., 1994, Commitments and conventions: The foundation of coordination in multi-agent systems, *Knowledge Eng. Rev.*, 8:223–250.
- Malone, T. W., 1987, Modeling coordination in organizations and markets, *Manage. Sci.*, 33:1317–1332.
- Montgomery, T. A., and Durfee, E. H., 1993, Search reduction in hierarchical distributed problem solving, *Group Decis. Negotiat.*, 2:301–317.
- Rosenschein, J. S., and Genesereth, M. R., 1985, Deals among rational agents, in *Readings in Distributed Artificial Intelligence* (A. H. Bond and L. Gasser, Eds.), San Mateo, CA: Morgan Kaufmann, pp. 227–234.
- Smith, R. G., 1980, The contract net protocol: High-level communication and control in a distributed problem solver, *IEEE Trans. Comput.*, C-29:1104–1113.
- Tambe M., and Zhang W., 1998, Towards flexible teamwork in persistent teams, in *Proceedings of the Third International Conference on Multi-Agent Systems*, Cambridge, MA: AAAI Press.
- Vidal, J. M., and Durfee, E. H., 1998, Learning nested models in an information economy, *J. Exp. Theoret. Artif. Intell.*, 10:291–308.
- Weiss, G., Ed., 1999, *Multiagent Systems*, Cambridge, MA: MIT Press. ♦

Muscle Models

Thomas G. Sandercock, David C. Lin, and W. Zev Rymer

Introduction

Muscle is a remarkable mechanical actuator that transduces chemical energy into force and motion, thereby providing power to move the skeleton. Because of the complexity of this transduction and the intricacies of muscle microstructure and architecture, no comprehensive models have yet been able to predict muscle performance completely. For this reason, muscle models are widely used to fulfill a variety of more narrowly defined objectives, ranging from attempts to promote understanding at the molecular level to more practical simulations of whole-muscle behavior. These practical simulations are typically used as part of a broader study of basic musculoskeletal biomechanics or for issues of understanding neural control mechanisms.

Muscle models can be usefully classified in order of increasing complexity. The more elaborate models generally have a wider range of application and can give more accurate results. However, this increase in fidelity is achieved at the cost of an increase in the mathematical complexity, involving parameters that are often not known initially and are not readily measured. Aside from the mechanisms of force generation, many other processes, such as activation, potentiation, and fatigue, also play an important role. In the absence of a single model that captures all these features, models are usually simplified to include only those behaviors that are deemed of interest in a particular application. This article outlines three major model classes and provides guidelines for their application:

1. *Input-output models.* The simplest of models are “black box” input-output models that attempt to capture very specific behavior over a restricted range of operation. Such models commonly use linear transfer function descriptors to transform neural excitation into force.
2. *Lumped parameter mechanical models.* The next level in complexity is typified by lumped parameter mechanical models. These are often composed of combinations of linear mechanical elements such as springs and dashpots to create fairly simple viscoelastic analogs of muscle. Nonlinear relations representing hyperbolic force-velocity behavior and tendon properties can also be incorporated. Such models are usually termed “Hill models.” The parameters characterizing the elements of these models are usually directly measurable by experiments. Model inputs may be neural excitation or length and force perturbations, while outputs may include muscle force, stiffness, and the time course of muscle length changes.

3. *Cross-bridge models.* More sophisticated “cross-bridge” models attempt to reproduce the dynamics of molecular processes that are responsible for force generation in muscle. These models incorporate mathematical descriptions of the dynamics of cross-bridge populations, their driving chemical reactions, and the resulting mechanical consequences. Such models usually require knowledge of numerous parameters and rate functions for the underlying reactions. Most of these parameters are not directly measurable, making the fitting of the model to a specific muscle or experimental situation difficult. Again, inputs can consist of neural excitation pulses or mechanical perturbations, while various outputs can be obtained from such a model, ranging from mechanical variables to thermodynamic information.

We next summarize briefly the relevant basic physiology of muscle. See McMahon (1984) for more details.

Muscle Architecture

A muscle is composed of many long, thin cells, or fibers, arranged parallel to each other. Most fibers terminate in microtendons, which merge to form a common tendon that connects to the skeleton. Because of this parallel organization, the total force a muscle can produce is proportional to the summed cross-sectional area of all the fibers. The fibers are, in turn, composed of several thousand parallel myofibrils. Each myofibril is composed of repeating microscopic units (2–3 μm in length) called sarcomeres, which are the basic contractile units of muscle. Since sarcomeres within a fiber are linked in series and contract together, many key muscle properties, such as the maximum speed at which a muscle can shorten, are proportional to the length of the fiber. For this reason, muscle contractile properties are often normalized by both the muscle cross-sectional area and the fiber length (Zajac, 1989).

Muscles come in an array of sizes and shapes, reflecting differences in fiber length, fiber number, and fiber orientation. There are also systematic differences in biochemistry and metabolic properties. See Alexander (1981) for a discussion of muscle and tendon architecture and Burke (1981) for a discussion of muscle fiber and motor unit specialization.

Input-Output (System) Models

At the simplest level, muscle may be treated as a linear system, usually of second order, with a single input and single output (SISO). In fact, since muscle has many simultaneous inputs—pri-

marily neural activation, but also length, force, temperature, and so on—additional constraints on the system are required to apply linear systems approaches.

For example, Mannard and Stein (1973) modeled isometric cat soleus muscle as a linear system, with a neural pulse train input and a force output. Here, motor axons were excited by a random electrical pulse train, and the resulting force data were well fitted by a critically damped, linear, second-order system. Essentially, this muscle acted as a low pass filter, with a cutoff of 5 Hz. Difficulties with the linear systems approach became apparent when slight changes were made in the experiment. By changing the amplitude of the input (mean stimulus rate to the ventral roots), system gain changed by more than a factor of 4, and the cutoff frequency ranged from 8 to 2 Hz. Changing muscle length also changed the system parameters. Thus, the linear approximation assumed by this model holds true only for closely specified conditions.

There are several other nonlinearities in muscle that limit the usefulness of the linear systems approach. First, active muscle behaves quite differently when it is shortening than when it is lengthening—behavior that is inconsistent with linear system properties. For example, if active muscle is stretched rapidly, force may drop precipitously after an initial region of high stiffness, giving rise to muscle “yield.” Second, muscle force shows marked hysteresis when measured during increasing neural activation compared with decreasing activation.

Because a SISO linear system model is at best an approximation, the model must be identified for the specific application for which it is used. Attempts to identify a more broadly applicable model by using nonlinear system techniques generally fail because muscle is quite nonstationary and the system changes before it can be fully characterized. Advances in nonlinear techniques may make this a viable approach. For now, the linear systems approach has the advantage of its well-developed theoretical background, which allows relatively simple system identification techniques to be implemented.

Lumped Parameter Models

The earliest experimentally based descriptions of muscle resulted in muscle models that were composed of viscoelastic elements. The most widely applicable of these models is that of A.V. Hill and can be described as follows: Muscle is composed operationally of three elements: (1) a contractile element (CE) that acts as an active force generator, (2) an elastic element (SE) that represents the combined stiffness of tendon and cross-bridges in series with the force generator, and (3) a second elasticity in parallel with the previous two elements (PE) that represents the passive tissue contributions to muscle force (Figure 1A).

Hill (1938) characterized the CE by applying a series of constant force inputs (i.e., an “isotonic load”) to active muscle. Muscle responds to such an input by shortening with an initially constant velocity. Smaller loads result in larger velocities. Plotting a number of such force-velocity pairs demonstrates this trade-off (Figure 1B), which can be fitted well with a hyperbola of the form

$$V_{CE} = \frac{b(P_o - F)}{F + a}, \quad F \leq P_o \quad (1)$$

where F is the applied isotonic force, V_{CE} is the resulting initial velocity of shortening, P_o is the maximum isometric (velocity equal to zero) force, and a and b are empirical constants. Equation 1 has been found to describe the steady-state force-velocity behavior of a wide variety of skeletal muscles during shortening. The constitutive relation for the CE embodies this hyperbolic force-velocity trade-off.

The series elasticity is usually modeled as a purely linear spring of stiffness k , and the passive elasticity often takes the form of an

exponential function that increases with extension. The total muscle length is the sum of the lengths of the CE and SE. The contribution of the PE depends on the precise muscle geometry but is often important only at long muscle lengths and is thus frequently neglected in practice. When passive tension is neglected, a single, first-order, ordinary differential equation expresses the dynamics of this model with force, F , as an output:

$$\frac{dF}{dt} = k_{SE}(V_M - V_{CE}) \quad (2)$$

where V_M is the velocity of the end of the whole muscle, V_{CE} is the contractile element velocity from Equation 1, and k_{SE} is the series elastic stiffness. The V_{CE} property described by Equation 1 applies only to shortening muscle and only when the muscle is operating in a length range at which isometric force does not vary. Modifications to this standard model extend its applicability to situations in which large length excursions occur. For example, length dependence (Figure 1C) can be incorporated into the force-velocity relation (Equation 1) by changing P_o to reflect the isometric force available at the current muscle length and scaling the parameters a and b by this factor as well. A length-dependent nonlinear SE stiffness can also be included.

In contrast, it is necessary to define an entirely new force-velocity relation to describe behavior during lengthening contraction, that is, when the muscle is forced to lengthen by an external load that exceeds its active force-generating capacity. One such relation (Mashima et al., 1972) is given by

$$V_{EC} = \frac{b'(P_o - F)}{2P_o - F + a'}, \quad F > P_o \quad (3)$$

where a' and b' are empirical constants. Such an extended model has been shown to perform well for complex motions involving both eccentric and concentric contractions at full activation (Krylow and Sandercock, 1997).

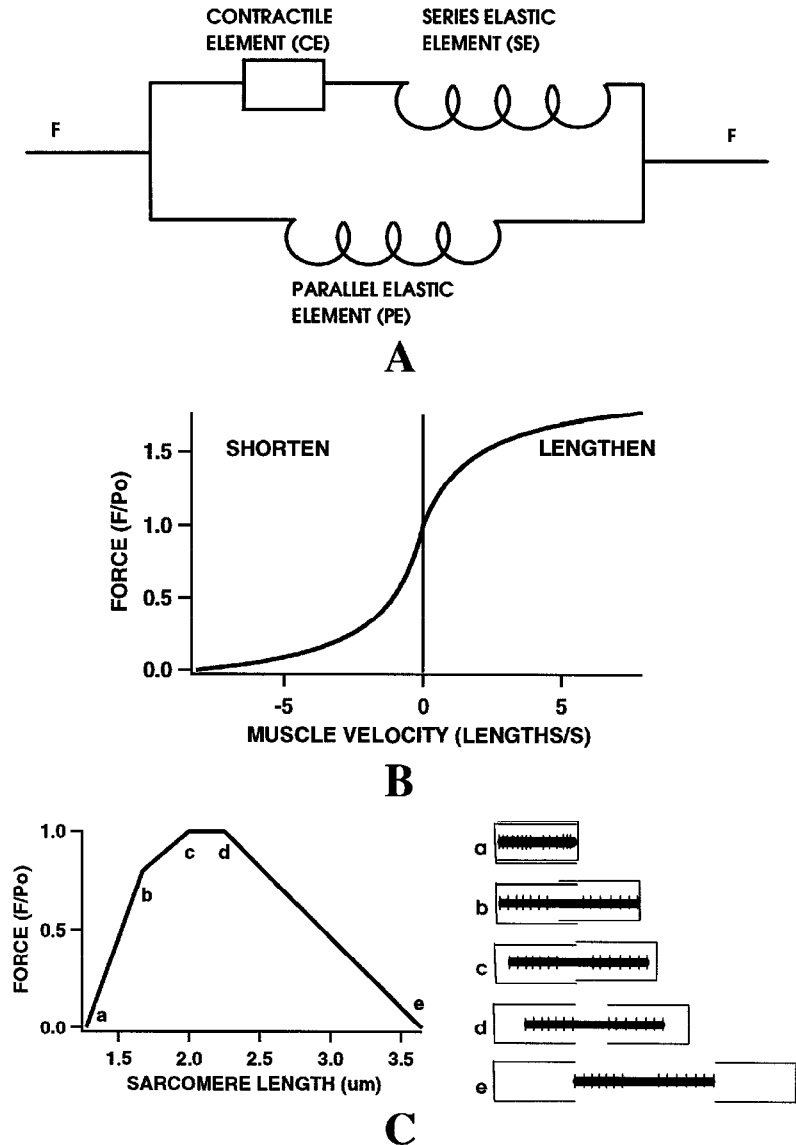
Cross-Bridge Models

The main components of the sarcomere are two sets of interdigitating protein filaments called thin filaments (made up partly of actin) and thick filaments (made up largely of myosin) (Squire, 1981). When suitably activated by calcium ions and in the presence of adenosine triphosphate (ATP), large populations of molecular projections (cross-bridges) on the thick filaments interact with receptor sites on the actin to produce force and relative motion between the two sets of filaments. Each cross-bridge is believed to act independently, interacting cyclically with successive actin sites to produce a ratchet-like action. The forces thus produced between the two filaments are in a direction to cause each sarcomere to shorten. The actin and myosin filaments are approximately inextensible, and sarcomere shortening occurs because of the relative sliding of the filaments past each other (sliding filament theory).

Muscle force exhibits a pronounced length dependence that can be explained by the sliding filament theory. As muscle length changes, the relative overlap of the actin and myosin filaments in each sarcomere changes because of telescoping of the sarcomere structure, and this overlap determines the maximum number of available cross-bridges at any given muscle length. Figure 1C shows the idealized length-tension curve measured during steady-state isometric contraction when the muscle is fully active.

In contrast to lumped parameter models, which try to reproduce macroscopic behavior with discrete mechanical elements, cross-bridge models strive to incorporate the known microstructure of the sarcomere together with biochemical kinetics of muscle protein to predict macroscopic variables such as whole muscle force, stiffness, shortening velocity, energy consumption, heat liberation, and

Figure 1. Schematic representations of (A) Hill model structure, (B) the force-velocity relation for both concentric and eccentric regions, and (C) isometric force-length relation and the corresponding sarcomere geometry responsible for this effect.



so on. The prototypical scheme for this type of model (Huxley, 1957) idealizes the interaction of actin and myosin as consisting of two possible cross-bridge states—either bound or unbound—and derives equations that describe the evolution of the distribution of bond lengths for the population of bound cross-bridges. The cross-bridges are assumed to act independently as linear springs, producing force in proportion to their extension, when bound to actin. It is necessary to define simplified chemical reactions that describe the transitions between the states and the form of the rate functions that determine the extent and directions of these reactions. Since the chosen rate functions depend explicitly on the cross-bridge length, the reactions are coupled directly to mechanical events of the cross-bridge cycle as well as to external perturbations imposed on the muscle by loading conditions. See McMahon (1984) and Zahalak (1981, 1992) for more details.

Tendon Properties

An idealized structure is often assumed for the tendon, in which muscle is connected to a linear series elastic element with high

stiffness. In fact, the tendon is far from being simply an inextensible link, and its mechanics modify muscle output significantly. For example, under some conditions, the muscle fibers can shorten while the complete muscle-tendon structure lengthens. The tendon can be used to store energy, as was demonstrated in the Achilles tendon of the wallaby (Alexander, 1981), where its energy storage plays an integral role in efficient jumping locomotion. Although tendon is often simply modeled as an ideal spring, the mechanical properties of tendon are more complex. Its force-length curve depicts an initial compliant region, followed by a reduced compliance, and is often described as exponential, exponential-linear, or a quadratic function of length. When a tendon is stretched and released, the measured force shows a pronounced hysteresis that is a complex function of its history. Nonlinear tendons have been incorporated as the series elastic elements in Hill-type models, providing some improvement in model accuracy.

The cross-bridges that generate force in a muscle are themselves often modeled as ideal springs. They can be lumped together and modeled as an elasticity in series with the tendon to define a global muscle stiffness. For example, in cat soleus muscle, which is con-

sidered to have a short, stiff tendon, half of its compliance (the reciprocal of stiffness) is attributed to the tendon when the muscle is fully activated. In muscles with longer tendons, the compliance of the tendon even predominates at high activation. At lesser activation levels, fewer attached cross-bridges result in the sarcomeres becoming the primary source of compliance in the muscle. Experiments show muscle stiffness increases approximately linearly with activation. This important mechanical property of muscle is often incorrectly represented in lumped parameter models.

Models of Activation

Muscle receives neural input in the form of discrete action potentials. Through a complex sequence of events, each action potential results in a release of calcium from stores in the sarcoplasmic reticulum. This calcium binds to regulatory proteins and allows cross-bridge cycling to proceed. The calcium is quickly sequestered, deactivating the actin-binding sites and allowing the muscle to relax. The frequency of action potential arrival at the muscle determines the degree of muscle activation. At low frequencies, muscle responds with discrete force transients (twitches). At high frequencies, isometric force fuses into a smooth contraction (tetanus) that rises to maximal levels (Figure 2).

The muscle models presented earlier need to be coupled with a model of activation to be fully comprehensive. These models fall into two basic categories: (1) models based on an estimated mean level of neural excitation to the muscle and (2) models that translate a sequence of discrete action potentials into muscle activation. When simulating voluntary movement, models based on estimated mean excitation are preferable because the true excitation is never known, and little is accomplished by trying to estimate the action potential sequence to all motor units in the muscle. However, when muscle is stimulated by a known action potential train, modeling of the discrete action potentials is necessary to reproduce the ripple occurring in an unfused tetanus.

The most widely used activation model uses the rectified and filtered electromyogram (REMG) to estimate the input to a muscle (DeLuca, 1979). The electromyogram signal is recorded by using either surface or intramuscular electrodes and results from the complex summation of electric fields produced by each muscle fiber action potential. The REMG provides a measure of the total number

of action potentials to the muscle. It often has a linear or exponential relationship to isometric force, and the output of this relationship can serve as the input to a Hill-type or cross-bridge model. Unfortunately, the relationship between the REMG and force varies in different muscles, for different recording electrodes, or even with the placement of the electrodes. Furthermore, because of its stochastic properties, an ensemble average of the REMG is needed for rapidly changing levels of excitation. At best, REMG is a crude approximation of neural drive.

The second approach to modeling activation is to approximate the physiological events after the arrival of an action potential at the muscle. Unfortunately, the complexity of the events precludes an accurate and simple model. In addition, activation is strongly influenced by muscle length. Well-documented activation-related phenomena can more than double or halve the force measured from a muscle stimulated by identical pulse trains. See Burke (1981) for discussions of potentiation, doublets, sag, and fatigue.

A simple model to transform a time sequence of action potentials into activation for a Hill-type model is described by

$$\begin{aligned} x(t) &= \sum_n \delta(t - t_n) \\ \dot{r}(t) &= -C_1 r(t) + C_2 x(t) \\ \dot{y}(t) &= \begin{cases} -C_3 y(t) + r(t), & y(t) \leq 1 \\ -C_3(t), & y(t) > 1 \end{cases} \end{aligned} \quad (4)$$

where $x(t)$ is the input, $\delta(t)$ are unit impulses representing action potentials at times t_n , and C_1 , C_2 , and C_3 are constants. Activation, $y(t)$, is used to scale the force-velocity relationship: P_o , a , b , a' , and b' in Equations 1 and 3 are multiplied by $y(t)$. The results of the model applied to cat soleus (Figure 2) show good agreement with the experimental data measured during isometric conditions. However, the model shows substantial errors (up to 50% of maximal muscle force) with low firing rates and high velocity length changes. The largest errors can be attributed to the separation of activation and contractile properties inherent in a Hill-type model (Sandercock and Heckman, 1997). More sophisticated activation models coupled with a cross-bridge model (Zahalak, 1992) address this problem but have not yet been shown to reduce the overall error for widely varying muscle conditions.

Furthermore, during natural contraction of a muscle, slow-fatigue-resistant motor units are active at low forces, and fast-fatigable motor units are recruited only for stronger contractions (Burke, 1981). Because the mechanical properties of these motor units are strikingly different, the overall mechanical properties of a muscle probably change substantially with activation. A muscle model might address this by changing the muscle parameters with increasing activation or by treating the fast and slow motor unit populations separately and combining the results. This problem has received little attention, yet is likely to be a significant source of error in existing models.

Discussion

As we outlined in the Introduction, the choice of a particular type of model is determined by the intended use of the resulting information. Linear system-type muscle models can provide intuitive insights in the frequency domain that are not easily obtained from the other methods, but muscle's inherent nonlinearities make such models locally applicable at best. Cross-bridge-type models are essentially the only choice to study molecular mechanisms. To study whole muscle or multiple muscle systems, both cross-bridge and Hill-type models offer possibilities, although Hill-type models are much more accessible. Cross-bridge models are capable of a wider range of behaviors, but they pay the price by needing more parameters. Because of the difficulty in estimating these parameters, cross-bridge models are rarely used to study control of mul-

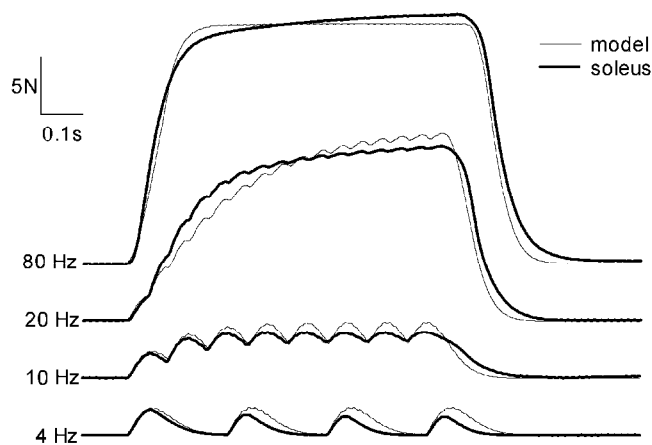


Figure 2. Cat soleus force responses for increasing rates of synchronous stimulation. At low stimulus rates, force pulses are responses to individual stimulus pulses. Force pulses fuse to produce smooth traces as the rate approaches 40 Hz. Simple Hill model predictions using the activation scheme given by Equation 4 show similar behavior.

timuscle movement. Hill-type models are by far the most widely used for this application.

A Hill model that is extended to include eccentric contraction and large muscle length changes does a fair job predicting muscle behavior and is the most practical solution for many requirements but has at least two major weaknesses. First, the extensions to Hill-type models for lengthening contractions fit well only under limited conditions and cannot predict muscle "yield." Second, when activation is coupled with the Hill model, the model has no mechanism to handle varying cross-bridge persistence observed with different movement histories. Systematic methods to identify the parameters in a simplified cross-bridge-type model could make it the method of choice.

It is not known how accurate a muscle model must be to effectively study control of movement. Since muscle is a nonstationary system, with crucial time- and history-dependent properties, it is difficult, even under carefully controlled laboratory conditions, to get the identical response to the same input. Perhaps neural control systems make such differences unimportant. Conversely, Lehman (1990) has shown that predicted neural control signals are very sensitive to the muscle model structure. Here, activations necessary to reproduce experimental wrist motions were calculated by using three different muscle models: a linear viscoelastic model, a Hill model with constant SE stiffness, and a Hill model with activation-dependent stiffness. The most complex muscle model predicted control signals that most closely resembled the actual EMG signals. Other nonlinear properties of muscle, the very ones that make modeling difficult, may be advantageous for function and thus may be required to help us understand the control of movement.

In Memoriam: The authors gratefully acknowledge the major contributions of Andrew Krylow and wish to dedicate this chapter in his memory.

Road Map: Mammalian Motor Control

Related Reading: Motoneuron Recruitment; Prosthetics, Motor Control

References

- Alexander, R. M., 1981, Mechanics of skeleton and tendons, in *Handbook of Physiology, Section L: The Nervous System*, vol. 2, *Motor Control*, Part I (V. B. Brooks, Ed.), Bethesda, MD: American Physiological Society, pp. 17–42.
- Burke, R. E., 1981, Motor units: Anatomy, physiology, and functional organization, in *Handbook of Physiology, Section I: The Nervous System*, vol. 2, *Motor Control, Part I* (V. B. Brooks, Ed.), Bethesda, MD: American Physiological Society, pp. 345–422. ♦
- Deluca, C. J., 1979, Physiology and mathematics of myoelectric signals, *IEEE Trans. Biomed. Eng.*, BME-26:313–326.
- Hill, A. V., 1938, The heat of shortening and the dynamic constants of muscle, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 126:136–195.
- Huxley, A. F., 1957, Muscle structure and theories of contraction, *Prog. Biophys.*, 7:255–318.
- Krylow, A. M., and Sandercock, T. G., 1997, Test of a modified Hill model in reproducing force responses involving eccentric contraction, *J. Biomech.*, 30, 27–33.
- Lehman, S. L., 1990, Input identification depends on model complexity, in *Multiple Muscle Systems: Biomechanics and Movement Organization* (J. M. Winters and S. L.-Y. Woo, Eds.), New York: Springer-Verlag, pp. 94–100.
- Mannard, A., and Stein, R. B., 1973, Determination of the frequency response of isometric soleus muscle in the cat using random nerve stimulation, *J. Physiol.*, 229:275–296.
- Mashima, H., Akazawa, K., Kushima, H., and Fujii, K., 1972, The force-load-velocity relation and the viscous-like force in the frog skeletal muscle, *Jpn. J. Physiol.*, 22:103–120.
- McMahon, T. A., 1984, *Muscles, Reflexes, and Locomotion*, Princeton, NJ: Princeton University Press.
- Sandercock, T. G., and Heckman, C. J., 1997, Force from cat soleus muscle during locomotor-like movements: Experimental data versus Hill-type model predictions, *J. Neurophysiol.*, 77, 1538–1552. ♦
- Squire, J., 1981, *The Structural Basis of Muscular Contraction*, New York: Plenum.
- Zahalak, G. I., 1981, A distribution-moment approximation for kinetic theories of muscular contraction, *Math. Biosci.*, 55:89–114.
- Zahalak, G. I., 1992, An overview of muscle modeling, in *Neural Prostheses: Replacing Motor Function After Disease or Disability* (R. B. Stein, P. Hunter Peckham, and D. B. Popovic, Eds.), New York: Oxford University Press, pp. 17–57. ♦
- Zajac, F. E., 1989, Muscle and tendon: Properties: Models, scaling, and application to bio-mechanics and motor control, *CRC Crit. Rev. Biomed. Eng.*, 112:52–62.

Neocognitron: A Model for Visual Pattern Recognition

Kunihiko Fukushima

Introduction

The *neocognitron* (Fukushima, 1980, 1988b, 1991) is a neural network model for deformation-resistant visual pattern recognition.

In primary visual cortex, neurons respond selectively to local features of a visual pattern, such as lines or edges in particular orientations. In the inferotemporal cortex, cells exist that respond selectively to certain figures such as circles, triangles, or squares, or even human faces. Thus, the visual system seems to have a hierarchical architecture in which simple features are first extracted from a stimulus pattern, then integrated into more complicated ones. In this hierarchy, a cell in a higher stage generally has a larger receptive field and is more insensitive to the position of the stimulus. This kind of physiological evidence suggested the network architecture for the neocognitron.

The neocognitron is a hierarchical network consisting of many layers of neuron-like cells. There are forward connections between

cells in adjoining layers. Some of these connections are variable and can be modified by learning. The neocognitron can acquire the ability to recognize patterns by learning. Since it has a large power of generalization, presentation of only a few typical examples of deformed patterns (or features) is enough for the learning process to be successful. It is not necessary to present all of the deformed versions of the patterns that might appear in the future. After learning, the neocognitron can recognize input patterns robustly, with little effect from deformation, changes in size, or shifts in position. It is even able to correctly recognize a pattern that has not been presented before, provided the pattern resembles one of the training patterns.

The principle of the neocognitron can be used in various kinds of pattern recognition systems, such as systems recognizing handwritten characters (Fukushima, 1988b, 2002; Fukushima and Wake, 1991).

The Network Architecture

The neocognitron has a multilayered architecture, as shown in Figure 1, in which each rectangle represents a two-dimensional array of cells. Each cell receives its input connections from only a limited number of cells situated in a small area on the preceding layer. The density of cells in each layer is designed to decrease with the order of the stage.

The lowest stage of the hierarchical network is an input layer U_0 , consisting of a two-dimensional array of receptor cells. Each succeeding stage has a layer U_S consisting of "S-cells," followed by another layer U_C consisting of "C-cells." Thus, in the whole network, layers of S-cells and C-cells are arranged alternately.

Each layer of S-cells or C-cells is divided into subgroups, called "cell-planes," according to the features to which they respond. The cells in each cell-plane are arranged in a two-dimensional array. Each rectangle drawn with heavy lines in Figure 1 represents a cell-plane. The connections converging to the cells in a cell-plane are homogeneous and topographically ordered. In other words, the connections have a translational symmetry such that each of the cells of a cell-plane shares the same set of input connections. This condition of translational symmetry holds for both fixed and variable connections. The modification of variable connections is always done under this condition.

S-cells are feature-extracting cells. They resemble simple cells in the visual cortex in their response. Connections converging to these cells may be modified by learning. After learning, S-cells are able to extract features from input patterns. In other words, an S-cell is activated only when a particular feature is presented in its receptive field. The features extracted by the S-cells are determined during the learning process. Generally speaking, local features, such as lines in particular orientations, are extracted in the lower stages. More "global" features, such as parts of a training pattern, are extracted in higher stages.

C-cells, which resemble complex cells in the visual cortex, are inserted in the network to allow for positional errors in the features of the stimulus. The connections from S-cells to C-cells are fixed and invariable. Each C-cell receives signals from a group of S-cells that extract the same feature, but from slightly different positions (Figure 2). The C-cell is activated if at least one of these S-cells is

active. Even if the stimulus feature is shifted in position and another S-cell is activated instead of the first one, the same C-cell keeps responding. Hence, the C-cell's response is less sensitive to shifts in the position of the input pattern.

The layer of C-cells at the highest stage is the recognition layer: the response of the cells in this layer is the final result of pattern recognition by the neocognitron.

Principles of Deformation-Resistant Recognition

In the whole network, with its alternate layers of S-cells and C-cells, the process of feature extraction by the S-cells and toleration of positional shift by the C-cells is repeated. During this process, local features extracted in lower stages are gradually integrated into more global features. Finally, each C-cell of the recognition layer at the highest stage integrates all the information of the input pattern and responds only to one specific pattern. Figure 2 illustrates this situation schematically.

Tolerating positional error a little at a time at each stage, rather than all in one step, plays an important role in endowing the network with the ability to recognize even distorted patterns. Figure 3 illustrates this situation. Let an S-cell in an intermediate stage of the network have already been trained to extract a global feature consisting of three local features of a training pattern "A," as shown in Figure 3A. The cell tolerates a positional error of each local feature if the deviation falls within the dotted circle. Hence, the S-cell responds to any of the deformed patterns shown in Figure 3B. The toleration of positional errors should not be too large at this stage. If large errors are tolerated at any one step, the network may come to respond erroneously, such as by recognizing a stimulus like Figure 3C as an "A" pattern.

Since errors in the relative position of local features are thus tolerated in the process of extracting and integrating features, the same C-cell responds in the recognition layer at the highest stage, even if the input pattern is deformed, changed in size, or shifted in position.

Self-Organization of the Neocognitron

The neocognitron can be trained to recognize patterns through either unsupervised or supervised learning. Various training methods have been proposed, and this section introduces two of them.

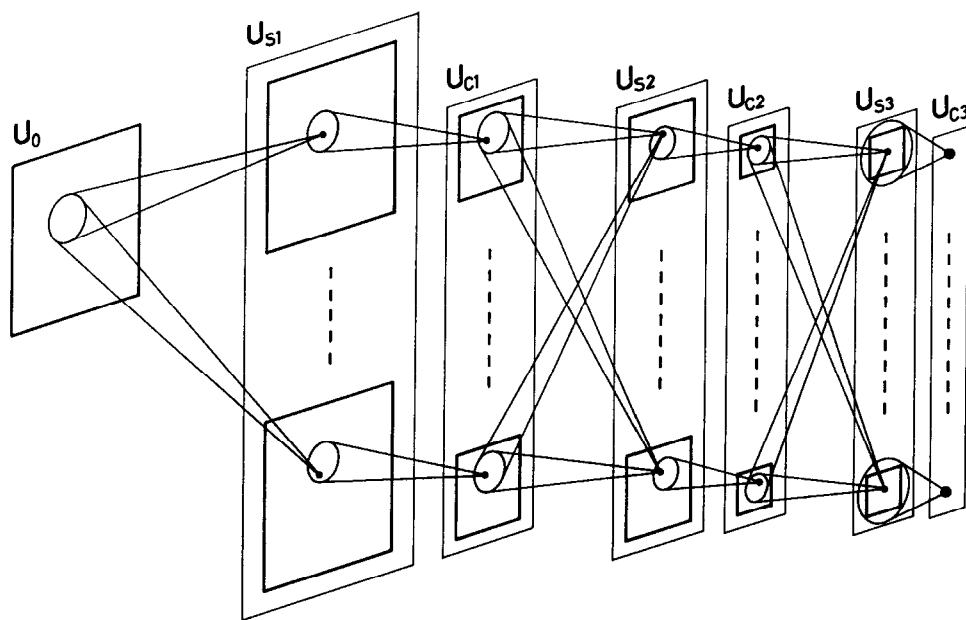
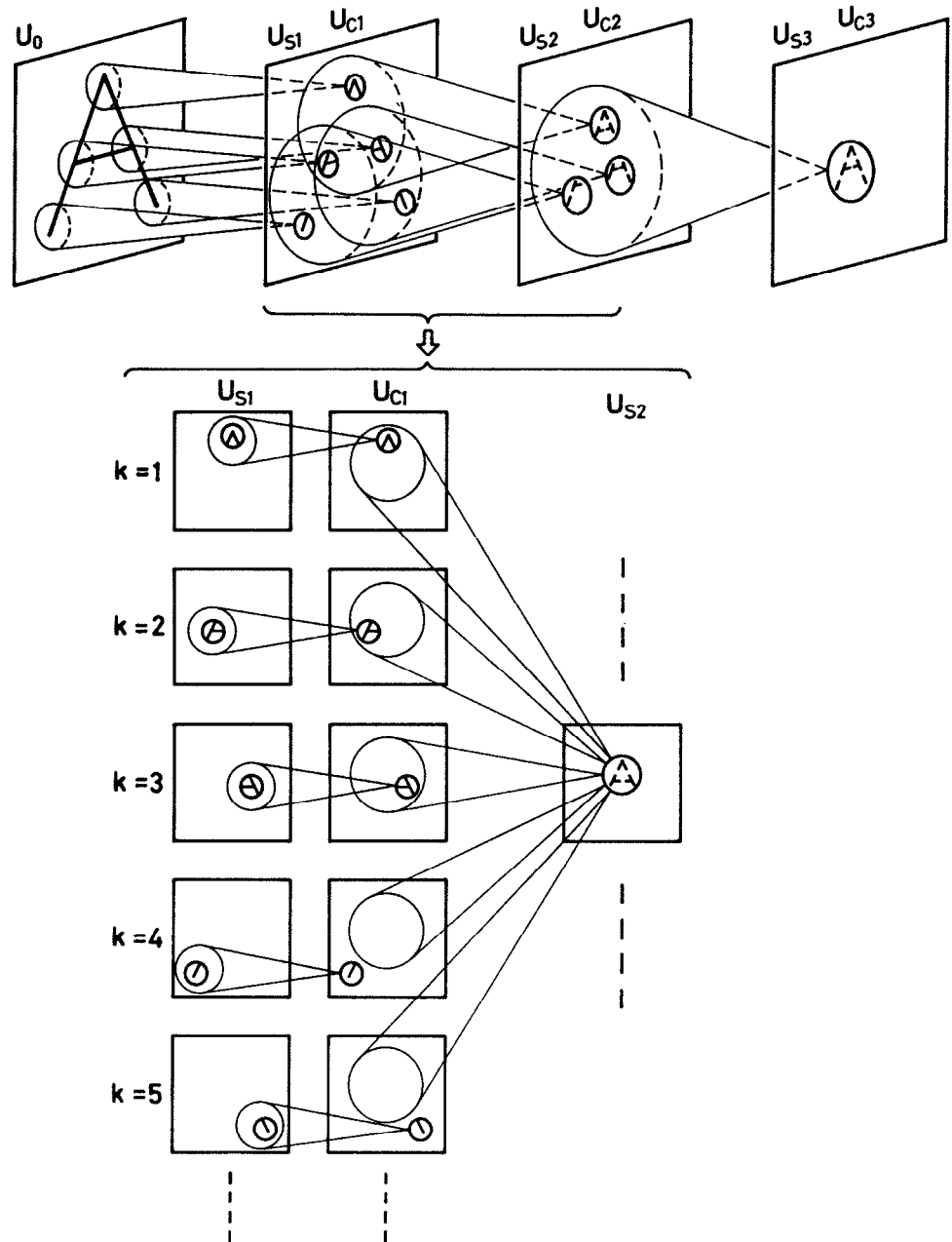


Figure 1. The network architecture of the neocognitron. Each rectangle drawn with heavy lines represents a "cell-plane." The cells in each cell-plane are arranged in a two-dimensional array.

Figure 2. Illustration of the process of pattern recognition in the neocognitron (Fukushima, 1980). As shown in the upper half of the figure, local features extracted in lower stages are gradually integrated into more “global” features. The lower half of the figure is an enlarged illustration of a part of the network. The cell-plane with $k = 1$ in layer U_{S1} consists of S-cells that extract \wedge -shaped features. Since the stimulus pattern “A” contains the \wedge -shaped feature at the top, an S-cell near the top of this cell-plane is active. A C-cell in the succeeding cell-plane ($k = 1$) in U_{C1} has excitatory input connections from S-cells situated in the circle and is activated if one of these S-cells is active. Only one cell-plane is shown in U_{S2} in this enlarged illustration. Each S-cell in this cell-plane detects the existence of features $k = 1, 2, 3$ in U_{C1} , and at the same time the absence of features $k = 4, 5$.



In the case of unsupervised learning, the self-organization of the network is performed using two principles. The first principle is a kind of “winner-take-all” rule (see WINNER-TAKE-ALL NETWORKS): among the cells situated in a certain small area, only the one responding most strongly has its input connections reinforced. The change of each input connection to this maximum-output cell is proportional to the intensity of the response of the cell from which the relevant connection leads.

Figure 4 illustrates this process, showing only the connections converging to an S-cell. The S-cell receives variable excitatory connections from a group of C-cells of the preceding stage. The cell also receives a variable inhibitory connection from an inhibitory cell, called a V-cell. The V-cell receives fixed excitatory connections from the same group of C-cells as does the S-cell, and always responds with the average intensity of the output of the C-cells.

The initial strength of the variable connections is very weak and nearly zero (Figure 4A). Suppose the S-cell responds most strongly of the S-cells in its vicinity when a training stimulus is presented (Figure 4B). According to the winner-take-all rule just described, variable connections leading from activated C- and V-cells are reinforced, as shown in Figure 4C. The variable excitatory connections to the S-cell grow into a “template” that exactly matches the spatial distribution of the response of the cells in the preceding layer. The inhibitory variable connection from the V-cell is also increased at the same time, but not strongly, because the output of the V-cell is not as large.

After the learning, the S-cell acquires the ability to extract a feature of the stimulus presented during the learning period. Through the excitatory connections, the S-cell receives signals indicating the existence of the relevant feature to be extracted. If an

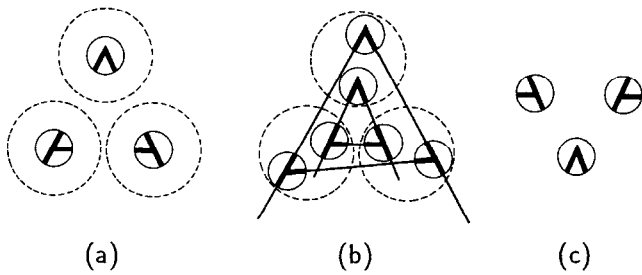


Figure 3. Illustration of the principle for recognizing deformed patterns (Fukushima, 1988a). An S-cell, which has already been trained to extract a global feature consisting of three local features as shown in part *a*, tolerates a positional error of each local feature if the deviation falls within the dotted circle. Hence, the S-cell responds to any of the deformed patterns shown in part *b*. The toleration of positional errors should not be too large at this stage. If large errors are tolerated at any one step, the network may come to respond erroneously, such as by recognizing a stimulus like the one in part *c* as an “A” pattern.

irrelevant feature is presented, the inhibitory signal from the V-cell becomes stronger than the direct excitatory signals from the C-cells, and the response of the S-cell is suppressed (Fukushima, 1989).

Once an S-cell is thus selected and reinforced to respond to a feature, the cell usually loses its responsiveness to other features. When a different feature is presented, a different cell usually yields the maximum output and has its input connections reinforced. Thus, a “division of labor” among the cells occurs automatically.

The second principle for learning is introduced in order that the connections being modified always preserve translational symmetry. The maximum-output cell not only grows by itself, it also controls the growth of neighboring cells, working, so to speak, like a seed in crystal growth. To be more specific, all of the other S-cells in the cell-plane, from which the “seed cell” is selected, follow the seed cell, and have their input connections reinforced by having the same spatial distribution as that of the seed cell.

Although the neocognitron can thus be trained by unsupervised learning, supervised learning is still useful when we want to train a system to recognize, for instance, handwritten characters, which should be classified not only on the basis of similarity in shape but also on the basis of certain conventions. In the case of supervised learning, the “teacher” presents training patterns to the network and points out the positions of the features that should be extracted.

The cells whose receptive field centers coincide with the positions of the features take the place of the “maximum-output cells” and become seed cells. The other process of reinforcement is identical to that of the unsupervised learning and occurs automatically.

It is another advantage of the neocognitron that these learning methods, both supervised and unsupervised, require extremely short training times compared with other learning algorithms such as backpropagation. In an extreme case of unsupervised learning, for example, three presentations of a training set consisting of one training pattern from each category was sufficient to train the network to recognize 10 numeric characters robustly (Fukushima and Wake, 1992).

The optimal scale of the neocognitron changes depending on the set of patterns to be recognized. If the complexity of the patterns is high, the total number of stages in the hierarchical network needs to be large. Conversely, the necessary number of cell-planes in each stage of the network increases with the number of categories of patterns to be recognized. However, the increase in scale is not proportional. For example, if we compare a system recognizing 35 alphanumeric characters with a system recognizing 10 numerals, the number of characters to be recognized increases 3.5 times, but the number of cells increases only 1.9 times (Fukushima and Wake, 1991). This results from the fact that the local features extracted in the lower stages are common, and they usually are contained in many patterns of different categories. Although the number of cells is large in these systems, the number of parameters required to describe the network is quite small, because all of the cells in each cell-plane share the same set of input connections.

Selective Attention Model (SAM)

Although the neocognitron has considerable ability to recognize deformed patterns, it does not always recognize patterns correctly when two or more patterns are presented simultaneously. The *selective attention model* (SAM) has been proposed to eliminate these defects (Fukushima, 1986, 1987, 1988a). In the SAM, backward (i.e., top-down) connections were added to the conventional neocognitron-type network, which had only forward (i.e., bottom-up) connections.

When a composite stimulus consisting of two patterns or more is presented, the SAM focuses its attention selectively on one of the patterns, segments it from the rest, and recognizes it. After the identification of the first segment, the SAM switches its attention to recognize another pattern. The SAM also has the function of associative recall. Even if noise or defects affect the stimulus pattern, the SAM can recognize it and recall the complete pattern from

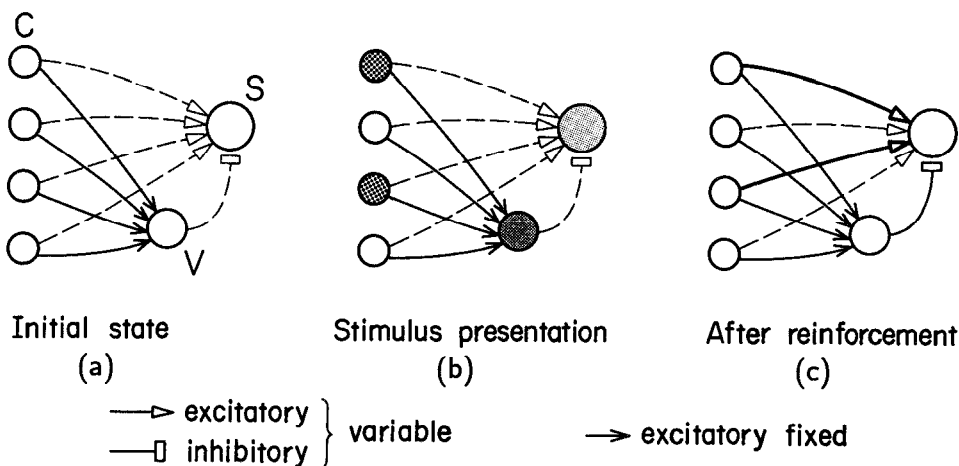


Figure 4. The process of reinforcement of the forward connections converging to a feature-extracting S-cell (Fukushima, 1988a). The density of the shadow in the circle represents the intensity of the response of the cell. *a*, The initial state before training. *b*, Stimulus presentation during the training. *c*, The connections after reinforcement.

which the noise has been eliminated and defects corrected. These functions can be successfully performed even for deformed versions of training patterns that have not been presented during learning.

The SAM has some similarity to the ADAPTIVE RESONANCE THEORY model (q.v.; see also Carpenter and Grossberg, 1987), but the most important difference between the two is the fact that the SAM has the ability to accept patterns deformed in shape and shifted in position, while the adaptive resonance theory does not, in principle, have such functions. With the SAM, not only the recognition of the patterns but also the filling-in process for defective parts of imperfect input patterns work on the deformed and shifted patterns themselves. The SAM can repair the deformed pattern without changing the basic shape and location of the deformed input pattern. The deformed patterns themselves can be repaired at their original locations, thus preserving their deformation.

The principles of the SAM can be extended to be used for several applications: for example, the recognition and segmentation of connected characters in cursive handwriting of English words (Fukushima and Imagawa, 1993), and the recognition of Chinese characters (Fukushima, Imagawa, and Ashida, 1991).

[Reprinted from the First Edition]

Road Maps: Learning in Artificial Networks; Vision

Background: Dynamics and Bifurcation in Neural Nets

Related Reading: Convolutional Networks for Images, Speech, and Time Series; Object Recognition; Visual Scene Perception

References

- Carpenter, G. A., and Grossberg, S., 1987, ART 2: Self-organization of stable category recognition codes for analog input patterns, *Appl. Opt.*, 26:4919–4930.
- Fukushima, K., 1980, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biol. Cybern.*, 36:193–202.
- Fukushima, K., 1986, A neural network model for selective attention in visual pattern recognition, *Biol. Cybern.*, 55:5–15.
- Fukushima, K., 1987, A neural network model for selective attention in visual pattern recognition and associative recall, *Appl. Opt.*, 26:4985–4992.
- Fukushima, K., 1988a, A neural network for visual pattern recognition, *IEEE Computer*, 21(3):65–75. ♦
- Fukushima, K., 1988b, Neocognitron: A hierarchical neural network capable of visual pattern recognition, *Neural Netw.*, 1:119–130.
- Fukushima, K., 1989, Analysis of the process of visual pattern recognition by the neocognitron, *Neural Netw.*, 2:413–420.
- Fukushima, K., 1991, Neural networks for visual pattern recognition, *IEICE Trans.*, E74:179–190. ♦
- Fukushima, K., 2002, Neocognitron for handwritten digit recognition, *Neurocomputing*, in press.
- Fukushima, K., and Imagawa, T., 1993, Recognition and segmentation of connected characters with selective attention, *Neural Netw.*, 6:33–41.
- Fukushima, K., Imagawa, T., and Ashida, E., 1991, Character recognition with selective attention, in *Proceedings of the International Joint Conference on Neural Networks, 1991*, vol. 1, New York: IEEE, pp. 593–598.
- Fukushima, K., and Wake, N., 1991, Handwritten alphanumeric character recognition by the neocognitron, *IEEE Trans. Neural Netw.*, 2:355–365.
- Fukushima, K., and Wake, N., 1992, Improved neocognitron with bend-detecting cells, in *Proceedings of the International Joint Conference on Neural Networks, 1992*, vol. 4, New York: IEEE, pp. 190–195.

Neocortex: Basic Neuron Types

Maria Toledo-Rodriguez, Anirudh Gupta, Yun Wang, Cai Zhi Wu, and Henry Markram

Introduction

The neocortex is functionally parcellated into vertical columns (~0.5 mm in diameter) traversing all layers (layers I–VI). These columns have no obvious anatomical boundaries, and the topographic mapping of afferent and efferent pathways probably determines their locations and dimensions as well as their functions (Peters and Jones, 1984; White, 1989). Multiple columns overlap, suggesting that the underlying neural microcircuits are designed to enable universal computation. These apparently omnipotent and stereotypical microcircuits are composed of a daunting variety of precisely and intricately interconnected neurons (Douglas and Martin, 1998; Somogyi, 1998; White, 1989), that differ in terms of their anatomical, electrophysiological, and molecular properties (Cauli et al., 1997; DeFelipe, 1993; Gupta, Wang, and Markram, 2000; Kawaguchi and Kubota, 1997; Peters and Jones, 1984; Thomson and Deuchars, 1997). This neuronal diversification may provide a foundation for maximizing the computational abilities of the neocortex.

Basic Neuron Types—Anatomy

Excitatory Neurons

Excitatory neurons constitute by far the majority of neocortical cells (70–80%) and consist mainly of two types of neurons (Peters and Jones, 1984; Somogyi, 1989; White, 1989):

- *Pyramidal cells* (PC; Figure 1A1), the most commonly occurring neocortical neuron (located in layers II–VI), are characterized by a single, prominent, vertically oriented dendrite emerging from the apex of their mainly pyramidal-shaped somata (apical dendrite), several (~4–6) more or less horizontally radiating basal dendrites, and long descending axons that project to other cortical and subcortical areas. The apical dendrite traverses through several layers, allowing PCs to sample multiple layer-specific inputs, before fanning out into a terminal tuft (often reaching layer I). The basal dendrites mainly remain within the layer of the cell body (sometimes entering adjacent layers), spanning the full diameter of the cortical column. PC axons give rise to a local columnar cluster that may spill over into neighboring columns before continuing to the white matter. Additionally, long vertical and horizontal collaterals project across layers and columns, sometimes forming secondary axonal clusters. PCs are therefore “local circuit neurons” as well as “projection neurons.”
- *Spiny stellate cells* (SSC; Figure 1A1), found almost exclusively in layer IV of the primary sensory areas, are characterized by multiple short dendrites (contained within a layer and column) radiating from spherical somata (stellate appearance). Their axons produce a local axonal cluster of columnar extent within layer IV, before projecting either loosely or in tight bundles to arborize extensively in layers II/III. Some collaterals descend toward layers V/VI. SSCs are mainly “local circuit neurons,” al-

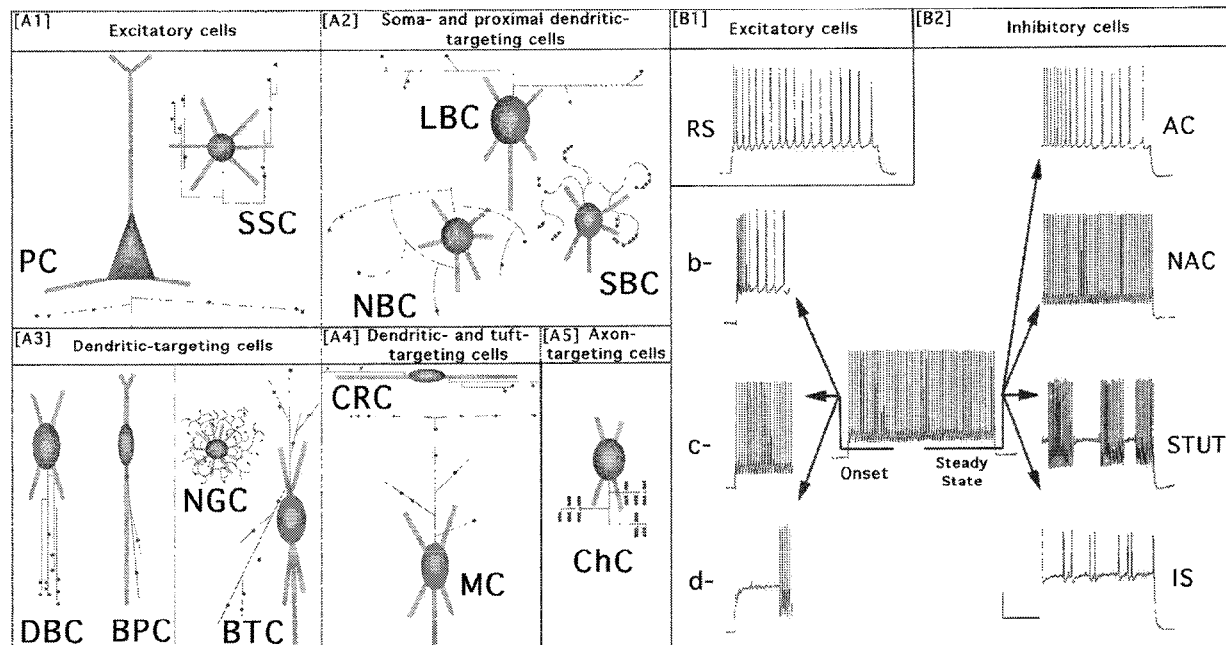


Figure 1. Anatomical and Electrophysiological Diversity of Neocortical Neurons. **A**, Schema, summarizing the main anatomical properties of neocortical excitatory (A1) and inhibitory (A2–5) neurons. Each neuron type is labeled by three-letter abbreviation (for explanation, see text). Dendrites: thick, light gray; axon: thin, black lines; black dots: axonal boutons. Spines omitted for clarity. Neurons oriented with pia facing upward and white matter (WM) downward. Note the presence of a prominent, vertical dendrite directed toward WM on some interneurons (A2–4). Inhibitory interneurons (A2–5) are mainly distinguished by the structure of their axonal arbor (see text) and typically innervate selective domains [A2: (peri-) so-

matic; A3,4 dendritic; A5: axonal] of their target cells. **B**, Representative samples of the most common discharge responses of neocortical excitatory (B1) and inhibitory (B2) neurons to standardized intrasomatic step-current injections. B1: Excitatory cells typically display regular-spiking (RS) discharge behavior. B2: Inhibitory interneurons display a vast repertoire of discharge responses, displaying either bursts (b-), delays (d-), or neither burst/delay (classical, c-) at step-onset, and accommodation (AC), non-accommodation (NAC), stuttering (STUT), or irregular spiking (IS) at steady-state. Scale bar (20 mV; 500 ms) applies to all traces.

though in few cases they have been shown to project to other cortical areas (Douglas and Martin, 1998).

Whereas PCs and SSCs are easily distinguished by multiple morphological features, both types of excitatory neurons share several *functional* properties, that have been amply used to distinguish them from inhibitory neurons (White, 1989): (1) their dendrites are typically densely studded with small membranous protuberances known as spines (hence they are also known as spiny neurons; see DENDRITIC SPINES); (2) they release glutamate from their presynaptic terminals (boutons), which form asymmetric (excitatory) synapses mainly onto the spines of other excitatory neurons (see NEOCORTEX: CHEMICAL AND ELECTRICAL SYNAPSES); (3) their somata invariably receive *only* symmetrical (inhibitory) synapses.

Inhibitory Neurons

Inhibitory neurons constitute 20–30% of the neocortical cells and are highly heterogeneous (Peters and Jones, 1984; Somogyi, 1989; White, 1989; Figure 1A2–5). They are easily distinguished from excitatory neurons by their lack of an apical dendrite, low spine densities (hence they are also known as smooth and/or sparsely spiny neurons), beaded dendrites, and axonal arbors that remain almost exclusively within a column (hence they are also known as local circuit neurons or interneurons; but see exceptions discussed later in this article). Instead of an apical dendrite projecting toward the pia, many interneurons have a prominent dendrite (with more branches) extending toward the white matter (WM). Moreover, the initial course of their axons, which either originate from the soma

or a primary dendrite, is often toward the pia (instead of toward the WM, which characterizes the axon trajectory of excitatory neurons). Inhibitory neurons release GABA at their symmetric synapses, and their cell bodies invariably receive *both* excitatory and inhibitory synapses. Most types of interneurons may display various soma shapes (ovoid, spindle-shaped, triangular, inverted pyramidal) and dendritic morphologies (bipolar, bitufted, and multipolar), but each type characteristically displays unique features in its axonal structure. Details of the axonal arborization (White, 1989), as well as the preferential placement of synapses onto different target-cell domains (Somogyi, 1989; Somogyi et al., 1998), have therefore provided the foundation for classifying interneurons. This selective innervation allows each type of interneuron to effect its target cells in a compartment-specific and potentially independent manner (see PERSPECTIVE ON NEURON MODEL COMPLEXITY).

Inhibitory neurons that selectively innervate:

- the (peri-) somatic region of their target cells, may affect the strength and gain of summated synaptic potentials (see SINGLE-CELL MODELS), the timing of action potential (AP)-generation and hence the concerted action of populations of target cells (see SYNCHRONIZATION, BINDING AND EXPECTANCY)
- the dendrites of their target cells, may influence dendritic processing and integration of synaptic inputs (see DENDRITIC PROCESSING), generation and propagation of dendritic APs, and synaptic plasticity (see HEBBIAN SYNAPTIC PLASTICITY)
- the axon initial segment of their target cells, may affect both the generation and the “gating” of APs (see AXONAL MODELING)

Most interneuron types, although mainly studied in layers II–V, are also found in layer VI, whereas layer I is characterized by its own distinct set of interneurons (see discussion later in this article). Moreover, it is currently not known whether additional subtypes, specific to layer VI, exist, although this lamina is characterized by a multitude of ill-defined local circuit neurons (~8–12 types) that still await precise description (Peters and Jones, 1984). The following describes the most common types of interneurons located in rat somatosensory cortex (layers II–V), based on a very large data set with considerable emphasis on quantitative morphometric analysis (see, for example, Wang et al., 2002). These basic interneuron types are found across different neocortical areas and species. Minor structural variations of these interneuron types (depending on neocortical layers, regions, age, and species) will not be considered here.

Interneurons that preferentially target somata and proximal dendrites. Basket cells (BCs), probably the most frequently encountered neocortical interneurons, are distinguished by their preferential innervation of somata (20–40%) and proximal dendrites (onto shafts and spines) (Kisvarday, 1992). BCs in general give rise to several beaded, mainly aspiny, dendrites. They are composed of three main subclasses, that differ in the structure of their axonal arborizations (Wang et al., 2002), each of which appears to be differentially distributed throughout layers II–VI.

- *Large basket cells* (LBC, Figure 1A2) produce a *sparse* local, mainly intralaminar and intracolumnar, axonal cluster composed of few, long and straight branches of low bouton density (BD) before generating their characteristic conspicuous long-range horizontal collaterals, that traverse multiple columns and some vertically projecting collaterals that may cross all layers (Somogyi, 1989). LBCs are therefore “local circuit” as well as inhibitory “projection” neurons.
- *Small basket cell* (SBC, Figure 1A2) give rise to a characteristic *dense* local, intralaminar and intracolumnar, axonal cluster composed of frequent, short, and curvy axonal branches with high BD. Occasionally SBCs may generate a few far-reaching collaterals projecting across layers and columns. A special subtype of SBC, termed *Clutch Cell*, has been observed in layer IV of the visual cortices of cat/monkey (Kisvarday, 1992). These cells are medium-sized, multipolar cells that typically produce large bulbous terminals, which often “clutch” somata of their target cells.
- *Nest basket cell* (NBC, Figure 1A2) give rise to a *sparse to dense* local, mainly intralaminar and intracolumnar, axonal cluster composed of infrequent, long, and smoothly bending axonal branches of low BD. They may occasionally produce a few far-reaching collaterals projecting across layers and columns. In addition, NBCs exhibit a characteristically simple dendritic arbor with few short and infrequently branching dendrites (Gupta et al., 2000; Wang et al., 2002).
- *Bipolar cells* (BPC, Figure 1A3) typically produce the simplest dendritic and axonal arborization of all interneurons, as both dendrites and axons branch very infrequently at shallow angles. The two long, vertically oriented, primary dendrites of BPCs are emitted from the opposite poles of their small spindle-shaped somata, and may span all cortical layers occasionally forming a dendritic tuft in layer I (Peters and Jones, 1984). The axon of BPCs typically emerges from a primary dendrite (usually the lower dendrite) before ramifying vertically across multiple or all layers. It is characterized by a very low number of boutons that are typically placed onto the dendritic shafts of rather restricted population of target neurons. Some BPCs in layers II–V have been shown to form asymmetrical synapses, preferentially onto spines, suggesting that they are excitatory (eBPCs, not shown; White, 1989).
- *Double bouquet cells* (DBC, Figure 1A3) are interneurons that like BPCs appear to consist of two classes. Inhibitory DBCs, that appear to be preferentially located in layers II/III, display bitufted or multipolar dendritic morphologies and typically produce a thin axon that bifurcates to give rise to a characteristic, mainly descending, “horsetail-like,” tight fascicular axonal bundle. The collaterals forming these narrow columnar bundles of high BD are typically much thicker than the main stem, and may extend across all layers. A local axonal ramification of different densities may occasionally be formed. Some double bouquet cells in layers II–V that generate *both* ascending and descending axonal collateral bundles have been shown to form asymmetrical synapses onto target cells, suggesting that they are excitatory (eDBC, not shown; White, 1989).
- *Neurogliaform cells* (NGC, Figure 1A3) are very small cells that produce dense, spherical dendritic, and axonal fields confined within a single layer and column (densest fields of all interneuron types). They typically produce a large number of thin, radiating dendrites that are short, aspiny, finely beaded, and rarely branched. Their very thin axons, branches intricately to produce a very dense and highly intertwined arborization (spiderweb-like appearance) that is studded with tiny boutons. NGCs target mainly dendritic shafts (Somogyi, 1989) and are also found in layer I.

Interneurons that preferentially target dendrites and dendritic tufts

Interneurons that preferentially target dendrites. Interneurons that preferentially target dendrites, usually give rise to beaded, aspiny, or sparsely spiny dendrites. Importantly, their overall “axonal fields” are preferentially vertically oriented (except NGCs, see the following discussion).

- *Bitufted cells* (BTC, Figure 1A3) display ovoid somata that emit two dendritic tufts from opposite poles that are preferentially vertically oriented and may emit an additional oblique dendrite (Somogyi, 1989). Their axonal arborizations are characterized by long, vertically oriented collaterals of low BD that may extend through all layers and mainly branch in a bifurcating manner. Their axonal ramification is mostly intracolumnar, although in some cases they may extend into neighboring columns.
- *Martinotti cells* (MC, Figure 1A4) display a more elaborate dendritic arbor than most interneurons, which is formed by beaded and sparsely- to medium-spiny dendrites. Their local and quite dense axonal cluster (mainly intralaminar and intracolumnar) is formed by collaterals that branch at wide angles before projecting up to layer I, where they spread across many columns, forming *spiny boutons*. MCs are “similar” to LBCs in that they are “local circuit” as well as inhibitory “projection” neurons. However, due to their innervation of distal dendrites and tufts, the form of their inhibitory impact is expected to differ substantially from that of LBCs.
- *Neurons exclusive to layer I* are believed to mainly innervate the dendritic tufts of target neurons and encompass several interneuron types. *Cajal-Retzius cells* (CRC, Figure 1A4) display large somata, long horizontal dendrites, and horizontally projecting axons, which characteristically give rise to numerous short ascending and some descending terminal fibrils. *Small layer I cells* (not shown) are neurons with short processes that constitute a heterogeneous group of multipolar interneurons with varying axonal arborizations. These have been subdivided into small neurons with poor or rich axonal plexus, respectively (Peters and Jones, 1984).

Interneurons that preferentially target axons

- **Chandelier cells** (ChC, Figure 1A5) are characterized by a local axonal cluster with a “chandelier-like” appearance resulting from the terminal axonal portions forming short vertical bouton arrays (“candlesticks”) onto the axon initial segments of target neurons (mainly PCs; Somogyi et al., 1998). Their local axonal clusters—mainly confined within a single layer and column—are formed by collaterals of high BD that frequently branch at shallow angles. ChCs give rise to mostly aspiny, beaded, infrequently branching dendrites that may span one or several layers.

Basic Neuron Types—Electrophysiology

Neocortical neurons display diverse intrinsic electrophysiological properties that result mainly from differences in their ion channel composition and constellation. Ion channels are state dependent and therefore a neuron’s passive and active properties may change according to different conditions (see ION CHANNELS: KEYS TO NEURONAL SPECIALIZATION). However, for standardized stimulation and recording conditions, neuronal discharge responses are stable and can serve as a reliable “marker” of their biophysical identity. Electrophysiological diversity indicates that identical spatiotemporal patterns of synaptic inputs will be differentially integrated and transformed into fundamentally different AP-patterns (and hence different synaptic outputs) and may therefore profoundly increase the computational repertoire of neural circuits (see SPIKING NEURONS: COMPUTATION WITH).

Excitatory Neurons

Excitatory neurons have been shown to display limited diversity in their discharge responses. Differences in their discharge properties have been described by three distinct features: (1) kinetic properties of single APs, (2) discharge response to intrasomatic threshold, and (3) supra-threshold current injections (Amitai and Connors, 1995; Connors and Gutnick, 1990; see Table 1). Discharge responses to supra-threshold current injections have proven to be the most useful parameter in distinguishing subclasses of both excitatory as well as inhibitory neurons (see the following discussion).

By far the most common discharge response observed for both PCs and SSCs has been described as *regular-spiking* (RS, see Figure 1B1). Sustained supra-threshold currents cause these cells to fire repetitively with a progressive decrease in firing frequency [progressive increase in inter-spike intervals (ISIs)], generally referred to as *spike train adaptation* or *accommodation*. Differences in the degree of accommodation have led to subclassification into RS1 (weak accommodation; *most common behavior*; see Figure 1B1) and RS2 cells (strong accommodation; behavior of PC- and SSC-subpopulations; see Table 1) (Connors and Gutnick, 1990). Some PCs and SSCs have been shown to display *intrinsic bursting* behavior (IB; not shown; see Connors and Gutnick, 1990). These neurons discharge with a cluster of three to five APs riding on a slow depolarizing wave (referred to as a burst), followed by an after-hyperpolarization, and then by either single spikes or bursts at more or less regular intervals (referred to as regular spiking and repetitive bursting, respectively). Other much less common discharge behaviors have been observed for subpopulations of PCs (see Table 1), including *chattering* (CHTs; not shown; Gray and McCormick, 1996) and *rhythmic firing* (RF; not shown; Amitai and Connors, 1995). CHT-cells usually display repetitive long clusters of APs to sustained supra-threshold current injections that, when made audible, sound like chattering. RF-cells discharge continually without accommodation.

Inhibitory Neurons

Inhibitory neurons display a much larger repertoire of discharge behaviors compared to excitatory neurons (see Figure 1B2; see Table 1), and their electrophysiological (sub-) classification has been gradually refined over the last decade. Initially only a single discharge behavior, known as *fast-spiking* (FS), was described for *smooth or sparsely spiny neurons* throughout layers II–VI. FS-cells generate *single* APs with characteristics distinct from that of spiny (excitatory) neurons (faster rise rates (RR) and fall rates (FR), distinct fast afterhyperpolarizing potentials (fAHP); Connors and Gutnick, 1990) and discharge *repetitively* at high frequencies with little or negligible accommodation to sustained supra-threshold currents. Since other discharge behaviors were not observed initially for smooth cells, it was believed that interneurons represent a homogeneous population of characteristically fast-spiking (referring to both the brevity of single APs and the resulting high discharge rate) neurons. Subsequent studies, mainly carried out in layers II/III and V, however, gradually demonstrated, that interneurons could display several other discharge patterns: (1) *burst spiking nonpyramidal* cells (BSNP) originally described as *low-threshold spiking* cells (LTS), typically display burst-like discharges after a hyperpolarizing pre-pulse (see Kawaguchi and Kubota, 1997); (2) *late-spiking* cells (LS) respond with a slow ramp depolarization and a late onset of discharge after a step current pulse (Kawaguchi and Kubota, 1997); (3) *regular spiking nonpyramidal* cells (RSNP) displaying discharge patterns similar to the RS response of PCs (Kawaguchi and Kubota, 1997); and (4) *irregular spiking* cells (IS) typically discharge with an initial burst of APs followed by an irregular spiking response (Cauli et al., 1997). IS cells have been further divided into two subclasses (IS1 and IS2) according to the duration of the initial burst.

Attempts to assign distinct electrophysiological discharge patterns to specific anatomical interneuron types have been made (Kawaguchi and Kubota, 1997; Thomson and Deuchars, 1997). Unfortunately, in many cases, the precise morphological identities of the electrophysiologically classified neurons could only be determined in *fractions* of the recorded cells: some MCs and DBCs were shown to display BSNP behavior, whereas LS behavior was observed for some NGCs and IS behavior for some interneurons with bipolar morphology (Cauli et al., 1997). Finally, DBCs, MCs, and BPCs may also display RSNP behavior, indicating that the same anatomical type may display more than one discharge pattern (see Table 1).

Interneuron discharge patterns, however, display an even richer diversity of behaviors. Recent studies—aimed at understanding the *functional* position of a large number of morphologically identified interneurons within the neocortical microcircuitry—adopted a simple classification scheme that encompasses previous schemes (Gupta et al., 2000; see Table 1) and considers both the *steady-state* and the *onset* response to sustained somatic current injections (Figure 1B2). According to this scheme, neocortical interneurons are categorized into five main classes with three subclasses each, according to the discharge response at steady-state and onset phase, respectively. These interneuronal discharge patterns are stable for (1) different baseline membrane potentials and (2) durations and amplitudes (several times threshold) of step current injections (Gupta et al., 2000; Wang et al., 2002):

- **Nonaccommodating** cells (NAC, Figure 1B2) fire repetitively without frequency accommodation (no or minimal change in ISIs). The steady-state discharge frequency increases steeply as a function of the injected current amplitude, allowing NACs to reach very high firing frequencies. Their APs are very brief and characteristically display a deep fAHP. NACs are the most frequently encountered cells in all layers

Table 1. Electrophysiological Classes of Neocortical Excitatory and Inhibitory Neurons

Electrophysiological Classes of Excitatory Neurons			
Supra-Threshold Responses	AP Characteristics	Threshold Responses	Morphological Type of Excitatory Neuron (*)
RS1	fast RR and fast FR	single AP	PC (layer II–VI); SSC
RS2	fast RR and fast FR	single AP	subpopulations of PC (layer IV–VI) and SSC
IB	fast RR and fast FR	burst	subpopulations of PC (layer V)
CHAT	nd	nd	subpopulations of PC (layer II/III)
RF	nd	bistable: no or nonaccommodating response	subpopulations of PC (layer V)
Electrophysiological Classes of Inhibitory Neurons			
Main Classes	Subclasses	Other Classification Schemes	Morphological Type of Interneuron (*)
NAC (layers I–VI)	b-NAC	FS	LBC, NBC
	d-NAC	FS; LS	LBC, NBC, SBC, BTC, NGC, ChC
	c-NAC	FS	LBC, NBC, SBC, BTC, MC
AC (layer II–VI)	b-AC	BSNP	NBC, BTC, MC, ChC,
	d-AC	FS(**); LS	LBC, NBC, ChC,
	c-AC	RSNP	LBC, NBC, SBC, BTC, BPC, DBC, MC, ChC
STUT (layers II–VI)	b-STUT	BSNP	NBC, BTC, MC
	d-STUT	FS; LS	LBC, NBC
	c-STUT	FS	LBC, NBC, BPC
IS (layers II–V)	b-IS	IS-1, IS-2	BPC
	c-IS	—	BPC, MC
	d-IS (nd)	—	—
BST (layers II–V)	i-BST	—	ChC
	s-BST	BSNP	BPC, DBC
	l-BST	BSNP	BPC, DBC

Interneuron classification: main classes and subclasses defined according to discharge responses at steady-state and onset-phase to intrasomatic current injections, respectively (see text; see Figure 1B2). Abbreviations explained in text; nd: not determined/detected so far; (*) morphological types listed according to sequence of description in text and not according to frequency of occurrence; (**) some authors have suggested to distinguish between accommodating FS-cells (FS-cells) and non-accommodating FS-cells (classical FS-cells or CFS; see Thomson and Deuchars, 1997).

- *Accommodating* cells (AC, see Figure 1B2) fire repetitively with a decrease in discharge frequency (the gradual increase in ISIs preventing high firing rates) and are the second most frequently observed electrophysiological class.
- *Stuttering* cells (STUT, see Figure 1B2) fire high frequency AP-clusters (with no or minimal accommodation) intermingled with unpredictable periods of silence (“morse-code”-like discharges). Cells displaying stuttering near threshold and fast spiking at slightly higher depolarizations are not considered STUTs.
- *Irregular spiking* cells (IS, see Figure 1B2) discharge single APs in a random manner throughout a depolarizing pulse, but do not form distinct clusters of APs.

Each of these main classes displays an array of stereotypical onset responses, which have been used for subclassification. They either discharge with:

- a *burst* (a high-frequency cluster of three or more APs), that *seamlessly* merges into the steady-state response (b-subclass, see Figure 1B2),
- a distinct *delay* before discharging to a current pulse (d-subclass, see Figure 1B2). The duration of the delay decreases progressively as the amplitude of current injection increases. Delayed discharging cells characteristically show significantly higher action potential thresholds than the b- and c-subclasses, or
- neither a burst nor a delay (referred to a *classical* response). The “onset” phase of these cells (c-subclass, see Figure 1B2) is therefore indistinguishable from the steady-state phase.

A fifth main class of *bursting* cells (BST; less frequent than the above main classes), with three subclasses, was recently observed (not shown). The onset response of BST cells is characterized by a high-frequency cluster of three to five APs riding on a slow *de-*

polarizing wave followed by a strong *slow* AHP, that causes a *clear separation* of the onset burst response from the consecutive steady-state responses, even at high current injections. The peak amplitudes of these APs decrease during the bursts in most cells. These burst properties differ fundamentally from those that define the b-subclasses of NAC-, AC-, STUT-, and IS-cells. BST-cells may be subclassified according to their steady-state discharge response into

- *r-BST* cells, which characteristically discharge *repetitive* burst (r = repetitive),
- *s-BST* cells, which characteristically fail to discharge after their initial burst due to a more pronounced, complex of powerful AHP (s = single), or
- *i-BST* cells, which characteristically discharge an accommodating train of APs (i = initial).

Recent computational studies have addressed the mechanisms of bursting behavior in neocortical PCs (see OSCILLATORY AND BURSTING PROPERTIES OF NEURONS). Similar studies for the other types of discharge behaviors in both excitatory and inhibitory neurons are eagerly awaited, especially in light of the profound effects, that different discharge properties may have on the behavior of neural circuits (i.e., van Vreeswijk and Hansel, 2001).

Basic Neuron Types—Molecular

Neocortical neurons express a variety of intracellular molecules including classical neurotransmitters (glutamate, GABA, acetylcholine, and catecholamines), neuropeptides, and calcium-binding proteins (CaBPs), as well as a multitude of different cell-surface molecules (neurotransmitter receptors, etc.). While these molecular species are found throughout the neocortex, each neuron only expresses some of these molecules and in specific combinations

Table 2. Molecular Classes of Excitatory and Inhibitory Neocortical Neurons

Molecular Classes of Excitatory Neurons (Glutamate +)			
Calcium Binding Proteins	Neuropeptides	Anatomical Identity	Electrophysiological Identity
CB	—	PC	RS
—	SOM	PC	RS
—	CCK	PC	RS
CB	CCK	PC	RS
Molecular Classes of Inhibitory Neurons (GABA +)			
Calcium Binding Proteins	Neuropeptides	Anatomical Identity	Electrophysiological Identity
CB	—	LBC, NBC, BTC, MC, DBC	c-AC, c-NAC, d-NAC
PV	—	ChC, LBC, NBC	c-NAC, d-NAC, c-STUT, d-STUT
CR	—	BPC, DBC, CRC	c-AC
CB + PV	—	LBC, NBC	d-NAC
—	NPY	LBC, MC, NBC	c-AC, c-NAC, d-NAC, c-STUT
—	VIP	DBC, BPC, SBC	c-AC, b-NAC, c-IS, b-IS
—	SOM	MC, BTC, NBC	c-AC, c-NAC, b-AC
—	CCK	LBC, BTC, MC, NBC, SBC	b-NAC, c-AC, c-NAC, c-STUT
CB	SOM	MC	c-AC
PV	SOM (*)	NBC	c-AC
CR	VIP	BPC, BTC	b-IS
CB	NPY + SOM	MC	c-AC

Excitatory neurons (shown to express glutamate) and inhibitory neurons (shown to express GABA or GABA producing enzymes GAD 65 and/or GAD 67), located throughout layers II–VI, have been sorted according to the detection of CaBPs, neuropeptides, and their co-expression. Abbreviations explained in text. In all cases listed, consistent expression profiles have been determined at both the protein and mRNA level, except (*), in which co-expression of SOM and PV was only detected at the mRNA level (compare Cauli et al., 1997, Wang et al., 2002, with Kawaguchi and Kubota, 1997). Note that the expression profiles are listed separately for *either* the anatomical *or* electrophysiological identities of the inhibitory neurons. Detailed information regarding expression profiles of SSCs not available to date.

(DeFelipe, 1993), allowing the use of expression and co-expression patterns for neuronal classification. Of these “molecular markers,” the (co-) expression of the most common CaBPs (calbindin, CB, parvalbumin, PV and calretinin, CR) and neuropeptides (neuropeptide Y, NPY, vasoactive intestinal peptide, VIP, somatostatin, SOM, cholecystokinin, CCK) has been most extensively studied (Cauli et al., 1997; DeFelipe, 1997; Kawaguchi and Kubota, 1997; Wang et al., 2002). The functional significance of this molecular diversity is currently not fully understood, although some of the above-mentioned “markers” (i.e., synaptically released neuropeptides) have been implicated in modulating synaptic transmission and/or neuronal excitability.

Table 2 summarizes molecular expression profiles of neocortical neurons (mainly layers II–VI; except CRC, see previous discussion) for the most commonly investigated CaBPs and neuropeptides, based on studies of protein- or mRNA-expression (mainly rodent neocortex). Whereas, *every* molecular expression profile detected at the protein level has been confirmed at the mRNA level, some mRNA-expression profiles have not been detected at the protein level (compare Cauli et al., 1997 and Wang et al., 2002, with Kawaguchi and Kubota, 1997). In general, neocortical excitatory neurons (mainly PCs) have been shown to differ from inhibitory neurons, (1) in that the percentage of PCs expressing CaBPs and/or neuropeptides is considerably lower and (2) in that they typically display a much more restricted set of expression profiles.

Discussion

This article outlines the main properties defining the basic cell types in the neocortex. The most striking feature of neocortical neurons is their immense anatomical, electrophysiological, and molecular diversity. All anatomical cell types can display multiple discharge patterns and molecular expression profiles. Different cell types are synaptically interconnected according to complex organizational principles to form intricate stereotypical microcircuits.

It is still unknown how afferent and efferent pathways determine the locations, dimensions, and functions of these seemingly omnipotent microcircuits that underlie the formation of functional columns. The major challenge for neural network models is to incorporate and account for the cellular diversity, which may explain the universal computational capability of these stereotypical microcircuits.

Road Map: Biological Neurons and Synapses

Related Reading: Biophysical Mosaic of the Neuron; Dendritic Processing; Ion Channels: Keys to Neuronal Specialization; Neocortex: Chemical and Electrical Synapses; Single-Cell Models; Temporal Integration in Recurrent Microcircuits; Visual Cortex: Anatomical Structure and Models of Function

References

- Amitai, Y., and Connors, B. W., 1995, Intrinsic physiology and morphology of single neurons in neocortex, in *Cerebral Cortex, Vol 11: The Barrel Cortex of Rodents* (E. G. Jones and I. T. Diamond, Eds.), New York: Plenum Press, pp. 299–331.
- Cauli, B., Audinat, E., Lambolez, B., Angulo, M. C., Ropert, N., Tsuzuki, K., Hestrin, S., and Rossier, J., 1997, Molecular and physiological diversity of cortical nonpyramidal cells, *J. Neurosci.*, 17:3894–3906.
- Connors, B. W., and Gutnick, M. J., 1990, Intrinsic firing patterns of diverse neocortical neurons, *Trends Neurosci.*, 13:99–104. ♦
- DeFelipe, J., 1993, Neocortical neuronal diversity: Chemical heterogeneity revealed by colocalization studies of classic neurotransmitters, neuropeptides, calcium binding proteins, and cell surface molecules, *Cereb. Cortex*, 3:273–289.
- DeFelipe, J., 1997, Types of neurons, synaptic connections and chemical characteristics of cells immunoreactive for calbindin-D28K, parvalbumin and calretinin in the neocortex, *J. Chem. Neuroanat.*, 14:1–19.
- Douglas, R., and Martin, K. A. C., 1998, Neocortex, in *The Synaptic Organization of the Brain* (G. M. Shepherd, Ed.), New York: Oxford University Press, pp. 459–509.

- Gray, C. M., and McCormick, D. A., 1996, Chattering cells: Superficial pyramidal neurons contributing to the generation of synchronous oscillations in the visual cortex, *Science*, 274:109–113.
- Gupta, A., Wang, Y., and Markram, H., 2000, Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex, *Science*, 287:273–278. ♦
- Kawaguchi, Y., and Kubota, Y., 1997, GABAergic cell subtypes and their synaptic connections in rat frontal cortex, *Cereb. Cortex*, 7:476–486. ♦
- Kisvárdy, Z. F., 1992, GABAergic networks of basket cells in the visual cortex, *Prog. Brain Res.*, 90:385–405.
- Peters, A., and Jones, E. G., 1984, *Cerebral Cortex, Vol. 1: Cellular Components of the Cerebral Cortex*, New York: Plenum Press.
- Somogyi, P., 1989, Synaptic organization of GABAergic neurons and GABA-A receptors in the lateral geniculate nucleus and visual cortex, in *Neural Mechanisms of Visual Perception. Proceedings of the Retina*

- Research Foundation Symposia* (D. K.-T. Lam and C. D. Gilbert, Eds.), The Woodlands: Portfolio Publications, pp. 35–63.
- Somogyi, P., Tamas, G., Lujan, R., and Buhl, E. H., 1998, Salient features of synaptic organisation in the cerebral cortex, *Brain Res. Rev.*, 26:113–135. ♦
- Thomson, A. M., and Deuchars, J., 1997, Synaptic interactions in neocortical local circuits: Dual intracellular recordings in vitro, *Cereb. Cortex*, 7:510–522. ♦
- van Vreeswijk, C., and Hansel, D., 2001, Patterns of synchrony in neural networks with spike adaptation, *Neural Comput.*, 13:959–992.
- Wang, Y., Gupta, A., Toledo-Rodriguez, M., Wu, C. Z., and Markram, H., 2002, Anatomical, physiological, molecular and circuit properties of nest basket cells in the developing somatosensory cortex, *Cereb. Cortex*, 12:395–410.
- White, E., 1989, *Cortical Circuits: Synaptic Organization of the Cerebral Cortex; Structure, Function, and Theory*, Berlin: Birkhauser Verlag. ♦

Neocortex: Chemical and Electrical Synapses

Jay R. Gibson and Barry W. Connors

Introduction

Synapses are specialized sites of communication between neurons. Our goal here is to summarize the diverse functional properties of synapses in neocortex. Synapses in the neocortex tend to be small, but their structure and biochemistry are complex. This intricacy befits their rich and highly dynamic functions. Short-term dynamics allow synapses to serve as temporal filters of neural activity; long-term synaptic plasticity provides specific, localized substrates for various forms of memory; modulation of synaptic function by neurotransmitters provides a mechanism for globally altering the properties of a neural circuit during changes of behavioral state. An important point is that each of these functions—short- and long-term plasticity and modulation—has diverse forms that vary between synapses, depending on their site within the cortical circuit.

Synapses come in two distinctly different types, chemical and electrical, both of which exist in neocortex. Chemical synapses are by far the most abundant. They use a chemical neurotransmitter that is packaged presynaptically into vesicles, released in quantized (vesicle-multiple) amounts, and binds to postsynaptic receptors that either open an ion channel directly or activate a G protein-coupled receptor. Electrical synapses are simpler in both structure and function. Their essential element is a protein called a *connexin*; 12 connexins form a single intercytoplasmic ion channel, and a cluster of such channels constitutes a *gap junction*. Electrical synapses provide a direct pathway that allows ionic current or small organic molecules to flow from the cytoplasm of one cell to that of another.

Our description of the neurons and synapses in the neocortex will follow the very simplified diagram shown in Figure 1. This represents only the input stage of the circuit, but it serves to outline the range of synapse functions seen throughout neocortex; we neglect synaptic connections between cortical layers and areas, as well as the output pathways. Most of the data discussed here originated in studies of sensory and motor neocortices of adolescent and adult rats, mice, guinea pigs, and cats.

Excitatory and Inhibitory Cells

To make sense of the diversity of neocortical synapses, it is essential to understand the nature of the neurons that they interconnect. Neocortical neurons are either excitatory or inhibitory; i.e., their axons form presynaptic terminals that release the excitatory transmitter glutamate or the inhibitory transmitter γ -aminobutyric acid

(GABA). Excitatory neurons include virtually all pyramidal cells and the spiny stellate cells of layer 4 (Figure 1, center). The somata of pyramidal cells exist in all layers below layer 1, and they extend an apical dendrite toward, and often into, layer 1. Pyramidal cells are the output neurons of the neocortex, and their axons often project to other brain areas. Spiny stellate cells are small, locally interacting cells confined to layer 4. Excitatory cells usually produce action potentials that have “regular spiking” characteristics, although subsets may have intrinsic bursting properties (Connors and Gutnick, 1990; see also OSCILLATORY AND BURSTING PROPERTIES OF NEURONS).

The inhibitory neurons tend to have relatively few spines on their dendrites (Thomson and Deuchars, 1997). Many classification schemes have been proposed for inhibitory interneurons based on

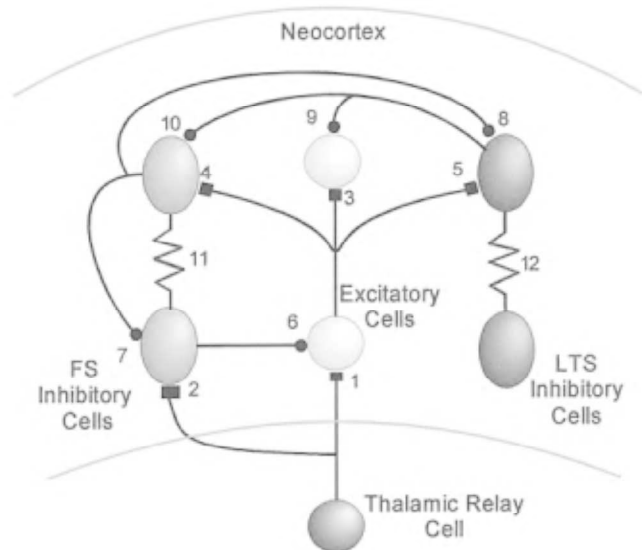


Figure 1. A simplified diagram of the neuronal circuitry in the primary input layer of sensory neocortex. Chemical synapses are represented by rectangles (for excitatory synapses) or dots (for inhibitory synapses), and electrical synapses are represented by zig-zags. The numbers refer to specific synaptic connections that are described in the text.

their dendritic and axonal morphology, the genes they express, their electrophysiological properties, and their synaptic physiology (e.g., Kawaguchi and Kubota, 1997; Gupta, Wang, and Markram, 2000; see also NEOCORTEX: BASIC NEURON TYPES). The number of inhibitory cell types and the criteria for distinguishing them are not universally agreed upon.

The majority of inhibitory neurons can be divided into three basic classes. The first expresses the Ca^{2+} -binding protein parvalbumin, and constitutes about 50% of all GABAergic neurons. Physiologically these cells are often called “fast spiking” (FS) because of their exceptionally short-duration action potentials, which can fire at high rates with little or no adaptation (Figure 1, left). Morphologically, many FS cells are either classical basket cells, with axons that make synaptic contacts onto somata or proximal dendrites, or chandelier (axo-axonic) cells, which synapse exclusively onto the initial axon segments of pyramidal cells. The second largest class of inhibitory neurons expresses the Ca^{2+} -binding protein calbindin, as well as the neuroactive peptide somatostatin; these neurons constitute about 17% of GABAergic neurons. Physiologically, at least some are “low-threshold spiking” (LTS) cells; their tonic firing shows adaptation, and they can fire rebound spikes in response to hyperpolarization (Figure 1, right). LTS cells often have vertically oriented dendritic patterns, and probably include many of the sparsely spiny, bitufted cells and Martinotti cells. A third class of cells stains for the Ca^{2+} -binding protein calretinin and for vasoactive intestinal polypeptide (VIP), and they also constitute about 17% of GABAergic neurons. Their physiological properties include irregular or intermittent firing patterns in response to steady current stimuli, and morphologically they are usually vertically oriented bipolar cells.

Neurotransmitters and Their Receptors

The excitatory neurotransmitter glutamate binds to two types of ionotropic receptors: non-NMDA and NMDA (Ozawa, Kamiya, and Tsuzuki, 1998). The binding of glutamate induces a conformational change in the receptor, which biases an ion channel toward an open state, allowing the passage of ions. Currents passing through these channels usually have a reversal potential around 0 mV. Because the transmembrane potential of a cell is usually much more negative than this, channel opening results in net inward current flow and transient membrane depolarization: the excitatory postsynaptic potential (EPSP). Usually, both types of glutamate receptors exist at the same excitatory synapse. Non-NMDA receptors can be further subdivided into AMPA receptors and kainate receptors. AMPA receptors mediate the vast majority of fast glutamatergic EPSPs, while the functions of kainate receptors are poorly understood. Most AMPA receptor channels are permeable to Na^+ and K^+ and have linear current-voltage relationships, but many excitatory synapses on inhibitory cells also include AMPA receptor subtypes that are permeable to Ca^{2+} and have a more nonlinear current-voltage relationship. In response to synaptically released glutamate, AMPA receptor-mediated currents have an extremely fast onset, and a decay time constant of roughly 3 ms.

NMDA receptor-mediated responses, in contrast, are slow; their decay time constants are about 40 ms. They are also highly permeable to Ca^{2+} , in addition to K^+ and Na^+ . NMDA receptors have a nonlinear voltage dependence, with highest conductance at potentials positive to about -30 mV. The voltage dependence of the NMDA channel is caused by extracellular Mg^{2+} ions, which block the channel due to electrostatic attraction; membrane depolarization releases this attraction. As a consequence, at standard excitatory synapses containing both NMDA and AMPA receptors, synaptic release of glutamate always opens AMPA receptors, whereas NMDA receptors are open only during the conjunction of glutamate release and relative depolarization of the postsynaptic

membrane. This property allows NMDA receptors to act as coincidence detectors, sensing the simultaneous activation of presynaptic glutamate release and postsynaptic activation of the cell through other synaptic inputs. The opening of NMDA receptors, and the subsequent influx of postsynaptic Ca^{2+} they provide, is thought to mediate many forms of long-term synaptic change.

Fast inhibitory synaptic transmission in neocortex seems to be exclusively mediated by the neurotransmitter GABA (Connors, 1992). GABA binds to ionotropic GABA_A receptors, inducing the opening of channels permeable to Cl^- . Cl^- reversal potentials tend to be about -70 mV in mature neocortical neurons. If the postsynaptic membrane is positive to this potential, inhibitory activation evokes a transient hyperpolarization—an inhibitory postsynaptic potential (IPSP)—owing to the influx of Cl^- . An inhibitory effect may result either from this shift in membrane potential, or from a current “shunt” caused by the increase in membrane conductance to Cl^- . Synaptically triggered GABA_A currents have a decay time constant of about 10 ms in cerebral cortex.

So far, we have only discussed synaptic transmission mediated by ionotropic receptors, i.e., receptors that are also ion channels. There are also many examples of synaptic transmission mediated by metabotropic receptors, i.e., receptors that activate G proteins, which in turn interact with downstream effector proteins. All neurotransmitters known to exist in neocortex activate specific metabotropic receptors. These include the two most important neurotransmitters that activate ionotropic receptors, glutamate and GABA, as well as other neuromodulators. Metabotropic responses tend to start more slowly than ionotropic responses (tens to hundreds of milliseconds versus fractions of a millisecond), last much longer (hundreds of milliseconds to seconds versus a few milliseconds), and can occur in both postsynaptic cells and presynaptic terminals. Postsynaptically, responses can be inhibitory, excitatory, or both. Most notable is the slow IPSP mediated by the GABA_B receptor, which is generated by an increase in K^+ conductance (Connors, 1992). Presynaptic metabotropic effects modify, and usually depress, the subsequent release of neurotransmitter.

Functional Properties of Chemical Synapses

The effectiveness of a chemical synapse is determined by several factors, including the probability of transmitter release (which can range widely, from near zero to almost one), the mean number of transmitter quanta released (which for many neocortical synapses is a maximum of one quantum per presynaptic terminal), the dynamics of the release process, and the history of prior activity. These factors often interact. We will describe how the functional properties of synapses vary between different pathways in neocortex; numbers in boldface refer to specific synapses in Figure 1. When we speak of a “unitary” synaptic response we mean the postsynaptic response to the firing of a single presynaptic cell or axon. One axon may make multiple synaptic contacts onto a single postsynaptic cell, so the strength of a unitary response is strongly determined by the number of synaptic contacts per axon, in addition to the physiological factors listed above. The “short-term dynamics” of a synapse refers to its time-varying changes in strength during repetitive activation over relatively brief intervals (milliseconds to seconds). Synapses may show short-term depression or facilitation, or a combination of the two.

Thalamic Input

Thalamic inputs are the conduit by which all specific information enters the neocortex. Thalamocortical axons form relatively strong excitatory synapses onto excitatory cells (1) and FS inhibitory cells (2), but not LTS cells (Figure 1; Gibson, Beierlein, and Connors, 1999). The strongest responses are generated in cortical layers 4

and 6, where most thalamic axons terminate. Unitary thalamocortical connections generate relatively large-amplitude EPSPs, with high reliability and practically no failures, and the responses show pronounced short-term depression (Figure 2A; Gil, Connors, and Amitai, 1999). Unitary thalamocortical EPSPs onto excitatory cells activate an average of roughly seven synaptic release sites, but unitary inputs to FS cells are twice as strong, with occasional single-axon responses surpassing 12 mV (FS mean = 4 mV, excitatory cell mean = 2 mV). Electron microscopic observations show that thalamocortical synapses terminate primarily on the soma and proximal dendrites of FS cells, and on the proximal dendritic spines of excitatory spiny stellate cells in layer 4 (Somogyi et al., 1998).

Intracortical Excitatory Synapses

The probability and strength of excitatory synaptic contacts vary widely and depend on the nature of the targets. Estimates suggest that the typical unitary connection between two excitatory cells (3) comprises about two to eight synaptic contacts, on average about half that of the average thalamocortical connection (Gil et al., 1999). These synapses target mostly dendritic spines (85%). There are exceptions. For example, a specific population of layer 6 spiny cells projects only 30% of its synapses onto spines (Somogyi et al., 1998). The amplitude of unitary intracortical EPSPs range from less than 0.5 mV up to 9 mV. The larger the response, the more reliable it tends to be. Failure rates are higher in connections with relatively small unitary responses (Thomson and Deuchars, 1997).

The dynamics of synapses between excitatory cells range from moderately depressing to weakly facilitating. The short-term dynamics of these intracortical connections depend on the maturity

of the cortex. For instance, in the adolescent rat (14–17 days postnatal), excitatory synapses between layer 5 pyramidal cells show clear depression, but by postnatal day 28 the same synapses are either weakly depressing or slightly facilitating.

Excitatory cells also synapse onto inhibitory cells, but the properties of the synapses vary with the postsynaptic inhibitory cell type. When excitatory axons fire at a rate above 5 Hz, EPSPs onto FS inhibitory cells (4) generally depress (Thomson and Deuchars, 1997). At low frequencies of stimulation (<0.2 Hz), unitary EPSPs onto FS cells are relatively reliable, with low failure rates. Unitary EPSP sizes can range from a few μ V to 12 mV, and they have a distinctly shorter duration than EPSPs in excitatory cells (Thomson and Deuchars, 1997). Axons from excitatory cells make about one to four synapses onto a single FS cell, and yield a response about 1 mV in amplitude. Most of these synapses terminate on dendrites, with fewer onto the soma.

The intracortical excitatory synapses that terminate on LTS inhibitory cells (5) have unusual dynamics, showing very strong short-term facilitation when activated at frequencies above 20 Hz (Figure 2B). At low-frequency stimulation (<1 Hz), EPSPs onto LTS cells have high failure rates, despite the likelihood that unitary connections are mediated by as many as three to 12 synapses. At higher frequencies, after facilitation develops, unitary EPSPs can be as large as 3 mV. Thus, inhibitory LTS cells may be activated only when local excitatory cells fire at high and sustained rates, suggesting that they serve as a “governor” on the cortical circuit, helping to prevent runaway excitation.

Metabotropic glutamate receptors (mGluRs) are expressed by various types of neurons in neocortex, both pre- and postsynaptically. Postsynaptic effects mediated by these receptors presumably require relatively large rates of glutamate release. Electron microscopy has localized mGluRs to the marginal regions of the subsynaptic membrane (Somogyi et al., 1998). Responses mediated by mGluRs are relatively slow, lasting hundreds of milliseconds. Some types of excitatory cells apparently depolarize when mGluRs are activated, while others hyperpolarize; it may be that responses vary with the neurons’ laminar location. Activating mGluRs depolarizes FS and LTS cells, but LTS cells are more strongly excited. Glutamate can also act on mGluRs expressed on presynaptic terminals. Activation of presynaptic mGluRs usually inhibits further transmitter release, reducing the amplitude of the postsynaptic response.

Fast, ionotropic excitatory synaptic transmission in neocortex is subject to long-term synaptic plasticity. Long-term potentiation (LTP) and long-term depression (LTD) occur primarily at excitatory synapses onto excitatory cells, but generally not at synapses onto inhibitory cells. Some forms of LTP and LTD depend on the opening of NMDA channels, while other forms of LTD depend on activation of the metabotropic glutamate channel.

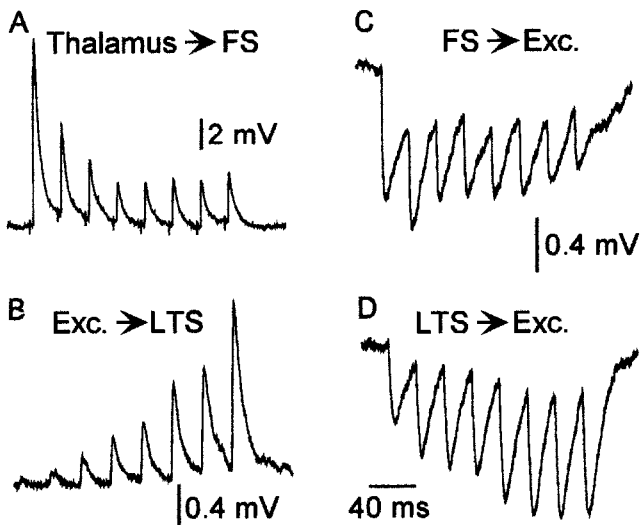


Figure 2. Variability in the strength and short-term dynamics of specific synaptic connections. In each case the presynaptic cell was activated at 40 Hz to test the dynamics of the postsynaptic response. *A*, The excitatory synapses from thalamocortical axons onto FS inhibitory cells (connection 2 in Figure 1) tend to be relatively strong and reliable, but depress during activation at high frequencies. *B*, The intracortical synapses from excitatory cells onto LTS inhibitory cells (5) tend to be quite unreliable when first activated, but strongly facilitate at frequencies above about 20 Hz. *C*, Inhibitory synapses from FS cells onto excitatory neurons (8) tend to be strong and reliable, and show moderate depression. *D*, Inhibitory synapses from LTS cells onto excitatory neurons (9) are moderately strong, and often facilitate during high-frequency activation. (Data from M. Beierlein, J. R. Gibson, and B. W. Connors.)

Inhibitory Synapses

Inhibitory cells may inhibit excitatory neurons or other inhibitory neurons. The connection specificity depends on the cell types involved (Thomson and Deuchars, 1997; Kawaguchi and Kubota, 1997; Somogyi et al., 1998). Calretinin-expressing interneurons preferentially inhibit other interneurons in visual cortex, but the functional properties of their synapses are unknown. FS (parvalbumin-expressing) cells frequently synapse onto excitatory cells (6), other FS cells (7), and onto LTS cells (8), where they tend to form synapses on the soma or dendritic shafts. Unitary axonal connections of up to 20 synapses have been described, from a single basket (FS) cell onto the soma and proximal dendrites of spiny cells and other inhibitory interneurons. Responses mediated by FS-derived synapses are very reliable, but tend to depress during repetitive activation (Figure 2C). The unitary IPSPs of these syn-

apses have peak amplitudes of about 2 mV when the postsynaptic cell's membrane potential is just below firing threshold. Some neocortical areas have an FS-like interneuron, called the axo-axonic cell, whose synapses exclusively and precisely target the axonal initial segments of pyramidal cells. The IPSPs of the axo-axonic connection have not been characterized in neocortex, but a similar connection in hippocampus displays short-term depression (Thomson and Deuchars, 1997).

Anatomical studies show that LTS-like (somatostatin-expressing) cells tend to synapse on the more distal dendritic shafts and spines of excitatory cells (9) and on distal dendritic shafts of FS cells (10). In one example, an LTS cell made ten synapses onto a single pyramidal cell—six onto spines and four onto dendritic shafts. Interestingly, recordings from neurons in layer 4 imply that LTS cells rarely make chemical synapses upon one another. The short-term dynamics of inhibitory synapses from LTS cells differ from those of FS cells. LTS-to-spiny cell IPSPs are fairly stable at 10 Hz and slower, while at 40 Hz and above they often show moderate facilitation (Figure 2D).

The computational consequences of inhibitory synapses can be complex, and synaptic location can play a role (Vu and Krasne, 1992). For example, activation of distal inhibition may shift the preferred input of the postsynaptic cell to a more proximal position. Proximally placed inhibition may be most effective for controlling the precise timing of action potentials and subthreshold activity. Postsynaptic inhibition can be divisive or subtractive in overall quality, depending on whether the inhibitory synapse is near the soma or farther out on the dendrites, respectively.

Activation of single inhibitory neurons tends to induce pure GABA_A receptor-mediated IPSPs in both excitatory and inhibitory cells. However, strong stimulation, which activates multiple inhibitory cells and yields large quantities of GABA release, can evoke an additional, and much longer-lasting, IPSP that is mediated by GABA_B receptors (Connors, 1992). Most GABA_B-mediated inhibition has been studied in excitatory cells, and its possible role in regulating inhibitory neurons is unknown. GABA_B receptors are also located presynaptically on both GABA- and glutamate-releasing terminals. When these receptors are activated, they inhibit evoked transmitter release.

Electrical Synapses

Electrical synapses have been directly demonstrated only between inhibitory neurons in the neocortex (Gibson et al., 1999; Galarreta and Hestrin, 2001), although there is indirect evidence that they may be more widespread early in development. Electrical synapses allow ionic current to pass directly between cells, equally well in both directions, with little or no voltage or time dependence. Recordings from two electrically coupled LTS cells are shown in Figure 3. Current steps of opposite polarity applied to LTS₁ induced either depolarization plus a train of action potentials, or a hyperpolarization (V₁); the voltage of LTS₂ responded in parallel with either strongly attenuated action potentials riding on a slower depolarization, or a slow hyperpolarization, respectively (V₂). The electrical synapses of inhibitory cells have an average "coupling coefficient," or attenuation factor, of about 0.1 for low-frequency signals; this decreases to about 0.01 for faster signals such as action potentials. Thus, a single presynaptic spike of 90 mV induces an electrical PSP that is typically less than 1 mV at its peak.

The functional significance of electrical synapses in neocortex is currently being studied. In general, electrical synapses promote synchronous electrical activity among the interneurons they connect. The electrical synapses between interneurons are located at dendrodendritic or dendrosomatic sites of contact (Tamas et al., 2000). This constrains them to closely neighboring cells, and indeed the most strongly connected inhibitory cells have somata

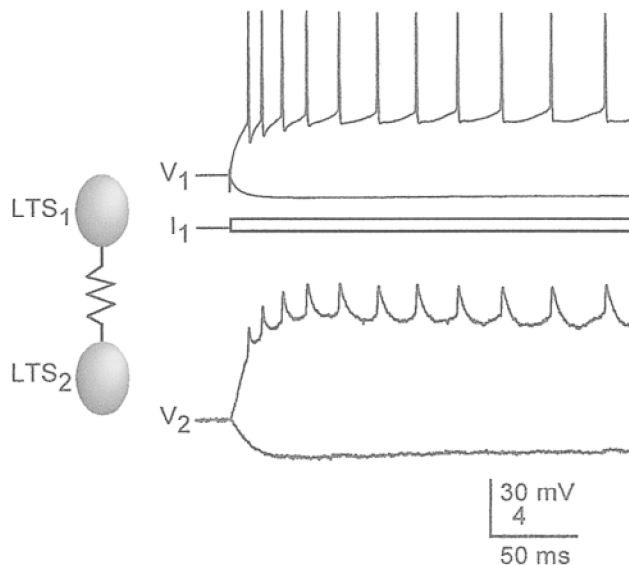


Figure 3. An electrical synapse between two LTS inhibitory neurons. Recordings on the top right show the superimposed responses of the first cell (LTS₁) as it was directly stimulated by two steps of injected current, depolarizing and hyperpolarizing. Responses from the second cell (LTS₂), on the lower right, were generated by current flowing through the electrical synapse from LTS₁. Notice that action potentials were more strongly attenuated than low-frequency voltage components. Voltage calibration is 30 mV for top traces, 4 mV for lower traces. (Data from J. R. Gibson, M. Beierlein, and B. W. Connors.)

within 100 μ m of each other. Extensive sampling of neuron pairs in layer 4 (Gibson et al., 1999) showed that, with rare exceptions, FS cells formed electrical synapses only with other FS cells (11), and LTS cells only with LTS cells (12). Thus, electrical synapses are a molecular and functional marker that serves to define and distinguish the boundaries between the FS and LTS inhibitory networks. It is also possible that the connexin-based channels of electrical synapses mediate the passage of small organic signaling molecules between interneurons.

Neuromodulation

Modulators such as acetylcholine, norepinephrine, serotonin, dopamine, and histamine can modulate presynaptic and postsynaptic neocortical function via G protein-coupled receptors. In general, these modulators are released throughout most or all of the neocortex from synaptic terminals whose axons originate extracortically, and the rates of modulator release depend strongly on behavioral state (Steriade, McCormick, and Sejnowski, 1993).

The two best-studied neuromodulators are acetylcholine and norepinephrine, and many of their effects are similar. Via postsynaptic actions, they can both depolarize and hyperpolarize excitatory cells, depending on the receptor subtypes that they activate, and they can also modulate spiking patterns via more subtle actions. Acetylcholine and norepinephrine depolarize both FS and LTS interneurons, but they are more effective at inducing LTS cells to surpass action potential threshold. Presynaptic effects of these modulators usually suppress transmitter release, resulting in smaller PSPs together with a reduction in short-term depression.

Discussion

There seem to be distinct rules governing the synaptic connections among neurons in the neocortex. First, connections are specific; for

example, thalamocortical synapses are strong and common onto FS interneurons but quite weak and rare onto LTS interneurons, and electrical synapses only interconnect interneurons of similar type. Second, the functional characteristics of synapses, such as efficacy, short-term dynamics, and location, vary with the identity of the pre- and postsynaptic neurons that participate in making the synapse. Third, synaptic function can be modulated by the common transmitters intrinsic to neocortex, glutamate and GABA, and by transmitters such as acetylcholine and norepinephrine that are mainly released by extrinsic neurons whose activity depends on behavioral state. The computational consequences of these complex synaptic properties are rich (e.g., Markram et al., 1998; see also TEMPORAL INTEGRATION IN RECURRENT MICROCIRCUITS and TEMPORAL DYNAMICS OF BIOLOGICAL SYNAPSES), although far from fully understood.

Road Map: Biological Neurons and Synapses

Related Reading: Neocortex: Basic Neuron Types; Neuromodulation in Mammalian Nervous Systems; Synaptic Interactions

References

- Connors, B. W., 1992, GABA_A- and GABA_B-mediated processes in visual cortex, *Prog. Brain Res.*, 90:335–348. ♦
- Connors, B. W., and Gutnick, M. J., 1990, Intrinsic firing patterns of diverse neocortical neurons, *Trends Neurosci.*, 13:99–104. ♦
- Galarreta, M., and Hestrin, S., 2001, Electrical synapses between GABA-releasing interneurons, *Nature Rev. Neurosci.*, 2:425–433. ♦
- Gibson, J. R., Beierlein, M., and Connors, B. W., 1999, Two networks of electrically coupled inhibitory neurons in neocortex, *Nature*, 402:75–79.
- Gil, Z., Connors, B. W., and Amitai, Y., 1999, Efficacy of thalamocortical and intracortical synaptic connections: Quanta, innervation, and reliability, *Neuron*, 23:385–397.
- Gupta, A., Wang, Y., and Markram, H., 2000, Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex, *Science*, 287:273–278.
- Kawaguchi, Y., and Kubota, Y., 1997, GABAergic cell subtypes and their synaptic connections in rat frontal cortex, *Cereb. Cortex*, 7:476–486.
- Markram, H., Gupta, A., Uziel, A., Wang, Y., and Tsodyks, M., 1998, Information processing with frequency-dependent synaptic connections, *Neurobiol. Learn. Mem.*, 70:101–112.
- Ozawa, S., Kamiya, H., and Tsuzuki, K., 1998, Glutamate receptors in the mammalian central nervous system, *Prog. Neurobiol.*, 54:581–618. ♦
- Somogyi, P., Tamas, G., Lujan, R., and Buhl, E. H., 1998, Salient features of synaptic organization in the cerebral cortex, *Brain Res. Rev.*, 26:113–135. ♦
- Steriade, M., McCormick, D. A., and Sejnowski, T. J., 1993, Thalamocortical oscillations in the sleeping and aroused brain, *Science*, 262:679–685. ♦
- Tamas, G., Buhl, E. H., Lorincz, A., and Somogyi, P., 2000, Proximally targeted GABAergic synapses and gap junctions synchronize cortical interneurons, *Nature Neurosci.*, 3:366–371.
- Thomson, A. M., and Deuchars, J., 1997, Synaptic interactions in neocortical local circuits: Dual intracellular recordings in vitro, *Cereb. Cortex*, 7:510–522.
- Vu, E. T., and Krasne, F. B., 1992, Evidence for a computational distinction between proximal and distal neuronal inhibition, *Science*, 255:1710–1712.

Neural Automata and Analog Computational Complexity

Hava T. Siegelmann

Introduction

Computational theory has developed hand in hand with the field of neural computation. The Turing machine was suggested in 1935 as a model of a mathematician who solves problems by using a specifiable algorithm. McCulloch and Pitts demonstrated the first computational model of a neuron, where they explicitly sought to provide a “brain” for the Turing machine. They proposed to model the nervous system as a finite interconnection of logical devices. Following the development of von Neumann’s universal model of computation based on the principle of the McCulloch-Pitts neuron, the digital approach prevailed in the field of cybernetic research.

In the last few decades, continuous, rather than digital, neural network models have been emphasized. Unlike the output values 0, 1 of the McCulloch-Pitts neuron, these models calculate continuous values. This was key to the development of the backpropagation algorithm that learns neural parameters; it enabled the foundation of machine learning, engineering tools such as optimal controllers, and adaptive technologies.

The connection between neural networks and computation, or more generally between physical systems and computation, is formalized by computational theories that describe features and capabilities of computational process. Automata are the basic mathematical abstractions used in such formalization.

A digital automaton has a set of internal states Ω , it receives a string of input symbols belonging to an alphabet Σ , and it moves from state to state according to the transition rules until the computation ends. Current neural automata are quite different from classical automata. Whereas classical automata describe digital machines, neural models frequently require a framework of analog

computation. In these terms, a physical system, beginning from an initial state (input), evolves in its state space according to an update equation (the computation process) until it reaches some designated state (the output).

Properties that distinguish analog from digital computation include the following:

1. Analog models are defined on a continuous phase space, while the phase space of a digital model is inherently discrete.
2. Physical dynamics is characterized by the existence of real constants that influence the macroscopic behavior of the system. In contrast, in digital computation all constants are in principle accessible to the programmer.
3. The motion of a physical system has local continuity. Unlike the flow in digital computation, analog models do not include locally discontinuous statements such as if $x > 0$ then compute one thing and if $x > 0$ continue in another computation path.
4. Continuous time dynamics is part of some analog systems.
5. A system composed of analog components may be sensitive to external noise.

In this article, we first review some work on analog and neural computation, and then focus on analog computation under noise. The fundamental question is, how should computational models described by networks of continuous neurons be characterized? The need for theories describing the operations and capabilities of machines that are analog or adaptive arises when one wants to describe and analyze nature’s computation, as well as the already developing analog chips and adaptive technologies.

Analog Computation

Blum, Shub, and Smale (1989) introduced a discrete-time computational model that operates in each time step on real-valued registers. The BSS model is considered a model of computation over the real numbers, rather than a model of analog computation, because it lacks the property of local continuity (item 3 above). Hybrid models of computation behave similarly. These models combine discrete- and continuous-time dynamics, usually by means of ODEs that are governed by finite automata. Because of their finite automaton component, hybrid systems also do not adhere to local continuity. A coupled map lattice is the analog version of a cellular automaton. This model is composed of an infinite lattice of variables with a local homogeneous transition rule, and can be defined generally enough to include practically any discrete-time model. The general-purpose analog computers, by Shannon (1941) and by Pour-el (1974), are based on continuous-time operators and include integrators. Unlike their name, these models are not universal and not very general.

Of special interest in physical computation are analog systems based on dissipative dynamics. Dynamical systems are called dissipative if their dynamics converge to attractors. When a dissipative system has energy (Lyapunov) functional, the attractors are fixed points; otherwise, more complex attractors may appear. Dissipative systems are mainly popular in neural modeling of memory, such as in the Hopfield and other related models. The meaningful attractors of these networks, where information is stored, are all simple: either stable fixed points or limit cycles. There are various models of neural activity that report chaos, such as in the olfactory, though they are typically not being considered in computational terms.

The dynamics of dissipative systems can be thought of as computation: either the initial state or parameter values can be considered the input to a computational problem, the evolution along the trajectory is the computation process, and the attractor describes the solution. We realized (Siegelmann and Fishman, 1998) that while fixed points can be computed efficiently, chaotic attractors could only be computed efficiently by means of nondeterminism. The inherent difference between fixed points and chaotic attractors led us to propose that, in the realm of dynamical systems, efficient deterministic computation differs from efficient nondeterministic computation: $Pd \ll NPD$.

A general theory of computation for dissipative dynamical systems was developed in Siegelmann, Ben-Hur, and Fishman (1999). This theory interprets the evolution of dissipative dynamical systems, both discrete and continuous in state space and in time, as a process of computation, and it relates the computational complexity to the true relaxation time. Prior to this work, no tool existed with which to analyze continuous-time algorithms and analog VLSI systems; ODE-based algorithms could only be analyzed by means of time discretization, which could result in loss of the main characteristics of these systems. We exemplified our theory with popular computer science problems, showing, e.g., a continuous algorithm for the maximum network flow problem that has linear time complexity, an algorithm for MAX that converges in logarithmic time, and a probabilistic continuous algorithm for the linear programming problem, which for a Gaussian distribution over instances of LP converges in linear time.

The analog recurrent neural network consists of a finite assembly of simple processors (or neurons), each of which computes a scalar—real-valued, continuous function, or activation, of an integrated input (Siegelmann and Sontag, 1994). This activation function is nonlinear and monotonic with bounded range, reminiscent of neural responses to input stimuli. The scalar value produced by a neuron is, in turn, broadcast to the successive neurons involved in a given computation. The existence of feedback loops in the

interconnection graph allows the processing of arbitrarily long data with a fixed size network. A related model is the network of spiking neurons (see SPIKING NEURONS, COMPUTATION WITH, and INTEGRATE-AND-FIRE NEURONS AND NETWORKS). There, the neurons output binary values only, but the time intervals between consecutive spikes are considered exact analog value. The computational analysis of the two models is similar under the transformation of neural value and time interval.

Neural Automata

Unlike the von Neumann computer model, the structure of neural networks is not separated into a memory region and a processing unit; memory and processing are strongly coupled. Each neuron is part of the processing unit, and the memory is implicitly encoded in the mutual influence between any pair of neurons. The influence can be represented by a real number weight. The status of the weights prompts two different views of the neural mode according to whether the weights are perceived as unknown parameters or as fixed constants. When the weights are considered unknown parameters, the network is a semiparametric adaptive technology, able to approximate input-output mappings by means of parameter estimation/learning. When the weights are considered constant (after or without a process of adaptation), the networks can perform exact computations rather than mere approximations. We next consider the latter case.

In the McCulloch-Pitts neuron, the potential is updated by

$$u_i(t) = \sum_{j \in \text{in}(i)} w_{ij} x_j(t - 1) \quad (1)$$

where $\text{in}(i)$ is the set of presynaptic neurons of i and $w_{ij} \in \mathbb{R}$ is the weight on the edge directed from neuron j to neuron i . The activation value is updated by

$$x_i(t) = \mathcal{H}(u_i(t) - c_i) \quad (2)$$

where c_i is the firing threshold of neuron i and \mathcal{H} is the binary response function

$$\mathcal{H}(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

In the analog recurrent network, the activation function is continuous, e.g., the sigmoid $\sigma(x) = 1/(1 + e^{-x})$, or the *saturated-linear function*:

$$\sigma(x) = \begin{cases} 0, & x \leq 0 \\ x, & 0 < x < 1 \\ 1, & x \geq 1 \end{cases} \quad (3)$$

The update equation of the basic analog neuron is

$$x_i(t) = \sigma(u_i(t) - c_i) \quad (4)$$

where u_i is defined as in Equation 1. Despite the similarity to the McCulloch-Pitts neurons, these two models are very different in their dynamical behavior and their computational power.

An asynchronous version of the analog neuron uses a potential, as in spiking neurons or integrate and fire neurons. Let $\mathcal{T}_j(t)$ be the set of firing times of neuron j until time t , and let ε_{ij} be a kernel function. The update equation of the neuronal potential is

$$u_i(t) = \sum_{j \in \text{in}(i)} \sum_{\tau \in \mathcal{T}_j(t)} w_{ij} \varepsilon_{ij}(t - \tau) \quad (5)$$

Some variants of this model take into account the refractory period as well.

Preliminaries: Language Recognition

The computer science approach of characterizing the discernibility between different inputs employs the concept of formal languages.

Let Σ be a set of symbols, e.g., $\{0, 1\}$, called an *alphabet*. Finite sequences of symbols in Σ are called *strings*. The set of all strings over Σ is denoted by Σ^* . A subset of Σ^* is called a *formal language*.

There are various ways to associate languages with automata, which are mostly equivalent. In this chapter we focus on language recognition. A *recognizer* reads an input string and decides whether the string is a correctly formed string of its language. Such a device has two types of halting states: accepting states and rejecting states. The language recognized or accepted by the automaton is the set of all input strings for which a computation ends in an accepting state. Thus, each automaton defines a language. Languages are divided into classes according to the type of automata that are needed to recognize them or the “difficulty” of recognizing them. Two types of automata are said to be equivalent if they accept the same class of languages.

A finite network of McCulloch-Pitts neurons has a finite memory only. This is far less than what one expects from a general digital computer. Even tasks like “decide whether there are more 0’s or more 1’s in a given binary input string” is not possible in a pre-defined size of memory. The formal description of a digital computer is in terms of the Turing machine. This consists of a control box with a finite memory coupled to a tape that is indefinitely expandable. The input is given by the initial state of the tape; the output is read off the tape if and when the Turing machine halts. Only with this extra tape are the languages recognized by digital machines captured mathematically.

To perceive neural automata as language recognition machines one needs to define an input-output convention. Two possibilities are outlined here. In the first the input is static, appearing in a designated set of neurons at the beginning of the computation. In the case of Boolean neurons, a finite network cannot encode an arbitrary number of symbols. (In this input convention one needs to consider a series of networks with an increasing number of input neurons.) In this article we consider mainly another input convention, where input arrives as a stream of symbols appearing consecutively on the input channels. This allows us to use a single network to recognize a language with strings of arbitrary length. In this convention, Equation 1 for the potential becomes

$$u_i(t) = \sum_{j \in \text{in}(i)} w_{ij} x_j(t-1) + \sum_{j \in I} b_{ij} I_j(t-1) \quad (6)$$

where I is the set of input channels, $I_j(t)$ is the value of input channel j at time t , and b_{ij} are the weights that connect the input channels to the rest of the network. There are various possible conventions for halting a computation of an analog machine, such as convergence to a fixed point, or, as we will consider here, a designated neuron reaching some value. Acceptance is then decided from the value of another neuron. In asynchronous models, the input stream arrives in a predetermined sequence of times $t_{\text{in}}^1, t_{\text{in}}^2, \dots$ at the input neurons.

Computational Power

Deterministic networks of Boolean McCulloch-Pitts neurons were studied first. Minsky (1967) showed that Boolean networks can simulate any combination of Boolean gates, and thus in particular, finite automata. Sima and Wiedermann (1998) have quantified the size of a network required to recognize a particular regular language. They also characterized the Hopfield languages—the languages recognized by Boolean automata with symmetric interconnections—as a strict subset of regular languages. If a series of Hopfield networks is considered in the static input convention, the computational power is much higher.

Analog recurrent neural networks have been analyzed in a series of papers and in Siegelmann (1999). Although the structure and

dynamics of analog recurrent neural automata are very different from those of the von Neumann machine, the automata were found to compute exactly like the digital computer when their parameters took rational values. Moreover, in the absence of noise, and if the networks took real-valued weights, they became more powerful than digital computers and recognized nonrecursive (super-Turing) languages. In particular, the efficient computation class is “P/poly.”

To explain the extra power stemming from the real weights, two more results were obtained (Siegelmann, 1999). In one, the complexity, or information content, of the weights was measured by a variant of resource-bounded Kolmogorov complexity, taking into account the time required for constructing the numbers. With Balczar and Gavald, we showed a full and proper hierarchy of non-uniform complexity classes associated with networks having weights of increasing Kolmogorov complexity, of which the classes “P” and “P/poly” are the two extreme cases.

Second, we proposed a new model that, although it seems digital, is still hypercomputational. This model is a network with rational weights only. In addition to the neurons, it includes a binary coin that outputs 0/1 with a probability p that is a real number. Although p is a real number, it is never accessed by any neuron (the coin is binary, so the other neurons receive from it digital values only). The computational power of this type of network is still hypercomputational (though less than “P/poly”). That is, the real value does not have to be explicit, but any process that is affected by a real number brings on nonrecursive computation.

All of the results mentioned in this section assume a noise-free environment.

Noisy/Probabilistic Analog Automata

The two classes of language that have arisen in noisy analog models are known as *regular* and *definite*. *Regular* languages are those accepted by finite automata. A language is called *definite* if for some integer r , any two strings coinciding on the last r symbols are either both or neither in the language. Definite languages are reminiscent of short-term memory. If the alphabet is finite, then definite languages are also regular; otherwise, they are not comparable with them.

Models of noisy analog neural networks were recently examined (e.g., Casey, 1996; Maass and Orponen, 1998; Maass and Sontag, 1999). The networks appeared to compute either regular or definite languages. Similarly, earlier models of probabilistic automata and circuits (e.g., Rabin, 1970; Paz, 1971; Pippenger, 1990) were found to compute sometimes regular and sometimes definite languages, depending on the properties of the probabilistic transition rules. These results make one wonder which details of the stochastic systems are essential for the computational power and which details are “accidental” aspects of the specific models. In particular, can one describe a general mechanism leading to the generation of these two classes of languages?

This challenge was partially met by Roitershtein and Siegelmann (1999), who proposed a general framework of stochastic computation. This framework, called *Markov computational systems* (MCS), includes all of the models just mentioned. Furthermore, it provides a natural way to introduce probabilistic counterparts of many diverse computational systems, such as topological automata, networks with nonfixed (e.g., growing) dimensions, hybrid systems that combine discrete and continuous variables, cellular automata, coupled map lattices, and the Blum, Shub, and Smale (1989) model.

In our generalization, the internal states of an automaton are substituted by distributions of states, and the transition function that guides the movement is substituted by operators, using an “operator theoretic” framework. The alphabet can be any set (e.g., real

numbers) and so is the state space (e.g., not necessarily Euclidean or even metric).

Definition 1. An operator P acting in the space of finite measures defined on a measurable space (Ω, \mathcal{B}) is said to be a *Markov operator* if for any probability measure μ , the image $P\mu$ is again a probability measure. A *Markov system* is a set of Markov operators defined on an alphabet Σ : $T = \{P_u : u \in \Sigma\}$.

A Markov system $T = \{P_u : u \in \Sigma\}$ is associated with a computational system, as follows. At each computational step $t = 0, 1, \dots$, the system receives an input symbol, $u_t \in \Sigma$, and updates its state, $x_t \in \Omega$. Let $\mu_t(A)$ be the probability of finding x_t in the set A . Then the evolution of the system is governed by the update equation, $\mu_{t+1} = P_{u_t}\mu_t$.

For an input sequence $w = w_0w_1 \cdots w_n \in \Sigma^{n+1}$, define a Markov operator $P_w = P_{w_n} \cdots P_{w_1}P_{w_0}$. If the probability distribution on the initial states is given by the probability measure μ_0 , then the distribution of states after $n+1$ computational steps on the input $w = w_0, w_1, \dots, w_n$ is defined by

$$P_w\mu_0(A) = P_{w_n} \cdots P_{w_1}P_{w_0}\mu_0(A) \quad (7)$$

Let \mathcal{A} be the set of “accepting” probability distributions, and let \mathcal{R} be the set of “rejecting” probability distributions. We need the condition

$$\text{dist}(\mathcal{A}, \mathcal{R}) = \inf_{\mu \in \mathcal{A}, \nu \in \mathcal{R}} \|\mu - \nu\|_1 = \rho > 0 \quad (8)$$

where $\|\cdot\|_1$ is the total variation norm $\|\mu\|_1 = \sup_A \mu(A) - \inf_A \mu(A)$. Then, a Markov computational system is defined to accept or reject the input according to the distribution it has after reading the input string.

Quasi-compactness and Regular Languages

$T = \{P_w : w \in \Sigma^*\}$ is considered quasi-compact when:

1. The alphabet Σ is finite.
2. The operators P_w are “close to being compact” in the sense that there exist $r > 0$ and $\delta < 1$ such that for every input string w of length r , there exists a compact operator Q_w , which satisfies $\|P_w - Q_w\|_1 \leq \delta$.

We proved the correlation between quasi-compact operators and the recognition of regular languages. Special cases of automata that satisfy the above condition are the probabilistic automata of Rabin (1970) and the noisy model of Maass and Orponen (1998). Various conditions leading to quasi-compactness were shown.

Weak Ergodicity and Definite Languages

Probabilistic computational models with the power to recognize definite languages have their roots in Rabin’s (1970) pioneering work on probabilistic automata. Paz (1971) generalized the method and introduced the notion of weak ergodicity for automata having a denumerable state space. Maass and Sontag (1999) focused on analog recurrent neural nets perturbed with additive Gaussian-like noise of a certain type. We pinpointed the underlying connection

of this chain and formalized its ultimate generalization. For this, we redefined weak ergodicity to describe all probabilistic systems that mix their probability distributions to the extent that they approach the uniform distribution. More formally,

Definition 2. A Markov system $\{P_u, u \in \Sigma\}$ is called *weakly ergodic* if for every $\alpha > 0$, there is an integer $r = r(\alpha)$ such that for any string w with length larger than r and any two probability distributions μ, ν ,

$$\|P_w\mu - P_w\nu\|_1 \leq \alpha \quad (9)$$

Let \mathcal{M} be defined with a weakly ergodic operator set. Then it recognizes only definite languages.

Remark 1. Weakly ergodic systems are robust with respect to perturbations of the system parameters and under some types of external noise.

Road Map: Computability and Complexity

Background: I.1. Introducing the Neuron

Related Reading: Analog Neural Nets: Computational Power

References

- Blum, L., Shub, M., and Smale, S., 1989, On a theory of computation and complexity over the real numbers: NP completeness, recursive functions, and universal machines, *Bull. AMS*, 21:1–46.
- Casey, M., 1996, The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction, *Neural Computat.*, 8:1135–1178.
- Maass, W., and Orponen, P., 1998, On the effect of analog noise in discrete time computation, *Neural Computat.*, 10:1071–1095.
- Maass, W., and Sontag, E., 1999, Analog neural nets with Gaussian or other common noise distribution cannot recognize arbitrary regular languages, *Neural Computat.*, 11:771–782.
- Minsky, M., 1967, *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice Hall.
- Paz, A., 1971, *Introduction to Probabilistic Automata*, London: Academic Press.
- Pippenger, N., 1990, Developments in: The synthesis of reliable organisms from unreliable components, *Proc. Symp. Pure Math.*, 5:311–324.
- Pour-El, M. B., 1974, Abstract computability and its relation to the general purpose analog-computer (some connections between logic, differential equations and analog computers), *Trans. AMS*, 199:1–29.
- Rabin, M., 1970, Probabilistic automata, *Inform. Control*, 41:539–550.
- Roitershtein, A., and Siegelmann, H. T., 1999, On Markov computational systems, Information Systems Engineering, Haifa, Israel, typescript.
- Shannon, C. E., 1941, Mathematical theory of the differential analyzer, *J. Math. Phys. MIT*, 20:337–354.
- Siegelmann, H. T., 1999, *Neural Networks and Analog Computation: Beyond the Turing Limit*, Boston: Birkhauser.
- Siegelmann, H. T., Ben-Hur, A., and Fishman, S., 1999, Computational complexity for continuous time dynamics, *Phys. Rev. Lett.*, 83:1463–1466.
- Siegelmann, H. T., and Fishman, S., 1998, Computation by dynamical systems, *Physica D*, 120:214–235.
- Siegelmann, H. T., and Sontag, E. D., 1994, Analog computation via neural networks, *Theoret. Comput. Sci.*, 131:331–360.
- Sima, J., and Wiedermann, J., 1998, Theory of neuromata, *J. ACM*, 45:155–178.

Neuroanatomy in a Computational Perspective

Almut Schüz

Introduction

This article is intended to help the modeler get a feel for real brains. It provides an introduction to basic principles of brain organization and to network structures within the brain. By focusing especially on the cerebral cortex, the article shows how quantitative neuroanatomy can contribute to brain theory.

The Role of Neuroanatomy in Brain Theory

In simple organisms in which the sensory organs are connected by direct pathways and effectors (muscles, glands), tracing such pathways may provide a sufficient explanation of behavior. However, in the course of evolution, the number of neurons has increased considerably and the pathways between input and output have become less and less direct.

The human brain contains between 7×10^{10} and 8×10^{10} neurons (Haug, 1986). Most of them belong to the cerebellar cortex (approximately 5×10^{10}) and the cerebral cortex (approximately 1.5×10^{10}). Many other parts of the human brain, such as the first relay station of the optic nerve in the thalamus, contain about 10^6 neurons. The corpus callosum, the main fiber bundle that connects the two hemispheres, consists of about 10^8 fibers. For a comprehensive collection of neuroanatomical measures, see Blinkov and Glezer (1968).

In small mammals, the numbers of neurons are reduced by a factor of about 1000 compared to the human brain. The cerebral cortex of the mouse, for example, contains about 10^7 neurons, and the corpus callosum of the mouse contains about 10^5 fibers. But even with these reduced numbers, a complete assessment of the neuronal network would be a hopeless undertaking.

However, the high number of neurons is not the main obstacle to understanding the neural mechanisms underlying higher brain functions. The situation is complicated by the fact that most connections in the brain are not one to one. In most nerve networks, it is beyond practical means to trace the anatomical pathways in detail. Many neurons receive input from thousands of other neurons and distribute their output to just as many. Physiologically, even if a direct anatomical connection between two neurons were demonstrated, whether the activity of neuron A leads to the activation of neuron B largely depends on what else is going on in the network.

Thus, in complex nerve networks, the task of neuroanatomy is not so much to study all of the connections in detail, but rather to show the typical structural properties that characterize a specific part of the brain. These properties provide clues to the understanding of its specific function, as will be shown later for the cerebral cortex.

White and Gray Matter: Projection Versus Computation

One general principle of construction in the brain is the spatial separation of long axons (composing the “white” matter) from the synaptic tissue in which the signal exchange between neurons occurs—the “gray” matter. Over short axons running within the gray matter, a neuron can reach other neurons within a radius of a few hundred microns, up to a few millimeters at most; but through the white matter, neurons can reach cells located in the centimeter range and in distant regions of the brain and spinal cord.

The largest mass of white matter in the human brain is that of the cerebral cortex. The white matter of the hemispheres consists largely of fibers that connect different parts of the cortex to each

other. Apparently, one of the principles of cortical connectivity is the rich projection of the cortex onto itself. In contrast, the thin sheet of white matter accompanying the cerebellar cortex indicates that local computation plays a major role in this part of the brain (Braitenberg, Heck, and Sultan, 1997).

White Matter, Brain Size, and Connectedness

Large brains have comparatively more cortical white matter than do small brains. For example, 42% of the human neocortex (white and gray matter taken together) is white matter, while in the hedgehog the white matter content is only 13% (Frahm, Stephan, and Stephan, 1982). Clearly, if one wants to connect a large cortex in a style similar to one with fewer neurons, the fiber mass has to increase more than proportionately to the number of neurons. The question is whether the increase in white matter with brain size indicates an increasing, decreasing, or constant degree of interconnectedness. The answer is either decreasing or constant, depending on whether one is focusing on individual neurons or on groups of neurons. The percentage of neurons of the cortex reached by an individual neuron is definitely lower in larger cortices than in smaller ones. However, if one defines compartments the size of the largest dendritic trees in each species, and if one postulates a complete set of connections between all the compartments, the relative increase in white matter corresponds to a roughly constant interconnectedness (Braitenberg, 2001).

In a similar way, Mitchison (1992) showed that if the neurons in the cortex were not organized into spatially distinct areas connected through the white matter, but were merged into one huge piece of gray matter, the volume of the human cortex would have to increase by a factor of 10 to maintain the same degree of connectedness.

Not only the number of connections but also time delays play a role in the interconnectedness of various parts of a network. In a large brain, distant elements may not be able to collaborate efficiently owing to delays in the transmission from one point to the other. Since conduction velocity depends on the thickness of an axon, this problem could theoretically be solved by an appropriate increase in the diameter of the longer axons in larger brains. However, this remedy is self-defeating; thicker fibers would further increase brain size and therefore increase time delays, which would in turn require thicker fibers, and so on. Starting with a brain as large as the human one, such a series would converge at a volume approximately 50% larger if one wanted interhemispheric signals to travel twice as fast as they do (Ringo et al., 1994). As a way out of this dilemma, Ringo et al. (1994) have proposed a higher degree of functional specialization of cortical regions in larger brains.

Overall Connectivity of the Brain

There are no isolated parts of the brain; there are fiber connections, direct or indirect, from any part of the brain to every other. Nevertheless, the brain is highly structured. Some parts (e.g., the cerebral cortex and the thalamus) interact directly with each other by way of reciprocal connections. Other parts may be arranged in loops, such as (1) the cerebral cortex–basal ganglia–thalamus (ventral anterior nucleus)–cerebral cortex loop and (2) the cerebral cortex–pontine nuclei–cerebellum (ventral lateral nucleus)–thalamus–cerebral cortex loop.

Such loops may traverse the same part of the brain and remain separate, as is the case for the two loops just mentioned when they pass through the thalamus. At other places, cross-talk may occur.

As a matter of fact, the cortical regions involved in these two loops overlap (Rouiller et al., 1994). In addition, shortcuts to the main stream may exist, parallel routes through further relay stations may accompany part of a loop, or subloops may be added to the main loop (see BASAL GANGLIA). Depending on the neurons involved, a signal can be fed back either negatively or positively onto the place of its origin. Positive feedback can also be transmitted by way of disinhibition when two inhibitory stations are connected in series. This type of positive feedback characterizes the cortico-cortical loop through the basal ganglia.

The projections between the various parts of the brain mostly suggest parallel processing in the sense that large regions of one part are connected to large regions of another by thousands or millions of fibers.

Sometimes, the projections from one part of the brain to another form patterns that suggest special kinds of computation. For example, the projections may be organized such that neighborhood relationships are maintained (e.g., the retinotopic projections in the visual system). In other cases, complex, patchy divergent, or convergent patterns may occur, suggesting a combination of inputs from different sources and/or distribution of inputs to disparate loci (e.g., projections from the various regions of the cerebral cortex to the basal ganglia).

The projections can also differ in that they may be point to point in some cases (sometimes one point to several points), while in others the terminal arbors of the axons may be smeared over large regions of the target structure. In a sense, this anatomical distinction corresponds to an important functional difference. Relatively restricted terminal arbors are typical for pathways that are involved in computation, such as those between the specific thalamic nuclei and the cerebral cortex or between cortical neurons. In contrast, huge terminal arbors extending over large portions of the brain and coming from small nuclei in the brainstem (e.g., the locus coeruleus) are involved in the global regulation of the level of activity, providing the background on which the information processing takes place (see EMOTIONAL CIRCUITS).

Geometry of the Gray Matter

Some of the structural features of the various parts of the brain are easily recognized at both the macroscopic and the microscopic level. The gray matter can either form lumps (often called nuclei) or show a two-dimensional, layered arrangement as in the so-called cortices. With respect to function, the latter type suggests that the same kind of operation is performed over the whole surface. One would expect such an arrangement, for example, in the processing of two-dimensional pictures. Indeed, this type of arrangement is found in centers for visual processing in all vertebrates, as well as in many invertebrates. Among the cortices, further distinctions can be made (Figure 1). In the cerebral cortex, the plane is isotropic in the sense that different directions cannot be distinguished in the histological picture. In contrast, in the cerebellar cortex, two perpendicular dimensions of the cortical plane are organized in completely different ways. Most of the axons run in a laterolateral direction, while the dendritic trees of the same layer extend in planes perpendicular to this.

A mixture of these two types also exists. In the hippocampus, the dendrites and some axonal systems are spread in all directions of the plane, while other axonal systems (mossy fibers, Schaffer collaterals) superimposed onto these run in one direction only. Furthermore, the serial reentrant arrangement of the latter is suggestive of cyclic operation (Figure 1C).

In contrast to cortices, geometry does not seem to play an important role in many nuclei. Their fine structure appears to be isotropic in all directions (Figure 1D).

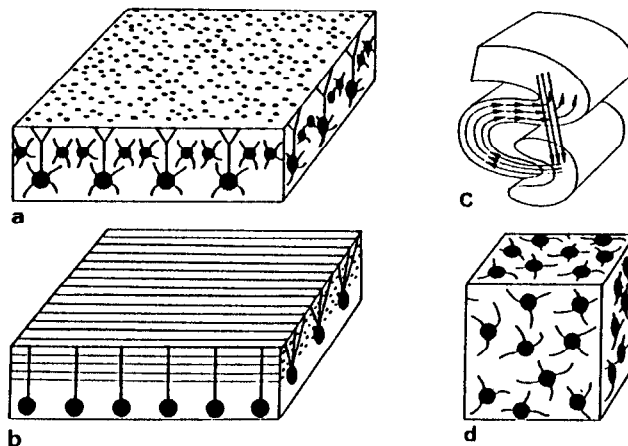


Figure 1. Different types of geometry in the gray matter. *A*, The structure in the horizontal plane is isotropic and different from that in the vertical plane (as in the cerebral cortex). *B*, All three planes of sectioning are different from each other (as in the cerebellar cortex). *C*, In the hippocampus, the horizontal (folded) plane is isotropic, but overlaid by a subpopulation of fibers that run in one direction only. These are, furthermore, arranged such that a cyclic operation is suggested. *D*, All three dimensions are equal. [Modified from Braitenberg, V., and Schüz, A., 1993, *Allgemeine Neuroanatomie*, in *Neuro- und Sinnesphysiologie* (R. F. Schmidt, Ed.), Berlin: Springer.]

Histology and Connectivity

Different parts of the brain differ with respect to density and size of neurons, shape of axonal and dendritic trees, or arrangement in layers. But in spite of our detailed knowledge of such structural features, it is difficult to grasp the underlying principles of connectivity. What are the rules, for example, behind the felt of axons stained in Figure 2? To how many neurons do they belong and where do they come from? Is what we see in Figure 2 an intermingling of separate circuits, or are these fibers part of the same network?

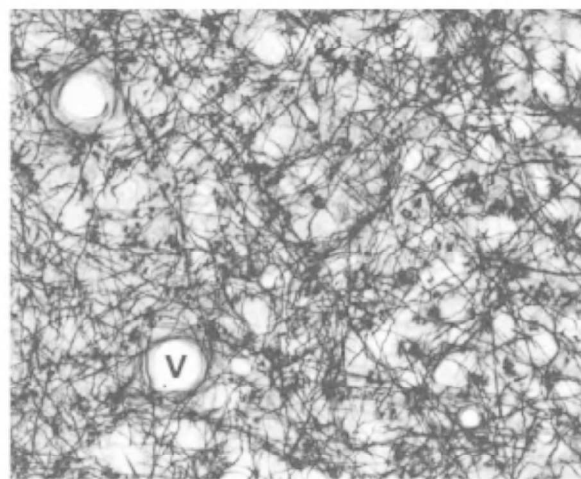


Figure 2. Axonal net. Light micrograph of the visual cortex of a monkey (area 17). A horizontal section is shown through layer IV. Only axons are stained. V indicates a blood vessel. The bar equals 50 μm .

Specific Versus Statistical Connectivity

One crucial problem springs to mind in view of a fiber felt as shown in Figure 2: the degree to which the target of an individual fiber is defined. This question is relevant to the anatomist in determining at what level of detail to analyze the structure. The theoretician, too, must decide whether to base a model on a certain well-defined connectivity or on a random network. (For a more comprehensive treatment of this topic, see Arbib, Érdi, and Szentágothai, 1998).

In Figure 3, three possibilities are depicted. In Figure 3A, neurons of type A connect to neurons of type B in a strictly defined manner. There is specificity between types (A, B) and between individual neurons ($A_{1/1}$ and $A_{2/1}$ to B_1 ; $A_{1/2}$ and $A_{2/2}$ to B_2 ; etc.). A prerequisite for this kind of connectivity is that the neurons can be labeled individually, either by their geometric arrangement or by some other means, such as a chemical marker. On the other hand, the specificity could be restricted to types of neurons (e.g., A connects to B only) without further specification, as shown in Figure 3B). In this case, it is no longer crucial whether a certain neuron A connects to a particular neuron B rather than to its neighbor.

In the third case (Figure 3C), there is no specificity whatsoever. The neurons of both types are intermingled and connect to the cells they happen to meet.

All three types of connectivity are realized in nature. An example of the first kind is the visual system of the fly. There, the photoreceptors are arranged in strict geometric order and connect to the first optic ganglion in a completely determined manner.

The second kind of network describes the situation in the cerebellar cortex. There, it is determined how the various types of neurons are connected. The granule cells, for example, connect to the other four cell types of the cerebellar cortex, but not to other granule cells; basket cells contact only Purkinje cells; and so on.

The network in Figure 3C comes close to the situation in the cerebral cortex. With the exception of the chandelier cells, which have been found to connect to pyramidal cells only, pyramidal and nonpyramidal cells connect to each other and often in the expected proportion, although interesting biases also occur (see White, 1989). The overall connectivity depends, then, on the density of neurons, on their ramification patterns, on the location of their terminal arbors, and on the relative numbers of the various cell types. These constraints, which are of a probabilistic nature, are in part genetically determined and in part refined by activity-dependent self-organization (see SELF-ORGANIZATION AND THE BRAIN).

Statistical Neuroanatomy

One approach to the question of connectivity in complex nerve networks is a quantitative assessment of the various components

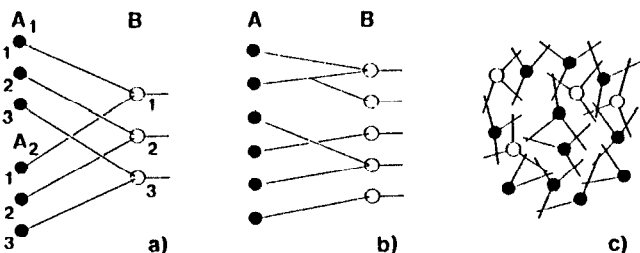


Figure 3. Three networks illustrating various degrees of specificity. A, All connections are determined. B, The connections are specified only with respect to type, not with respect to individual neurons. C, There is no specificity. [From Schüz, A., 1992, Randomness and constraints in the cortical neuropil, in *Information Processing in the Cortex* (A. Aertsen, and V. Braitenberg, Eds.), Berlin: Springer, fig. 1, p. 4. Reprinted with permission.]

shown by the different histological methods (cell bodies, synapses, axons, etc.). How such data can be used to determine the structural properties of a network and to constrain neural models accordingly was shown in detail for the cerebral cortex (see Braitenberg and Schüz, 1998, and references therein). The main points are briefly summarized here.

Basic Structure of the Cerebral Cortex

Some of the quantities that have been measured are shown in Table 1 (*a–m*). They refer to the mouse cortex and are corrected for tissue shrinkage. The letters *n–s* show further quantities that can be derived from those. A number of these quantities require some explanation.

The *relative density of axons* (Table 1, *q*) is defined as follows. Imagine a piece of cortex punched out perpendicularly to the cortical surface and just large enough to contain the local axonal tree of an individual neuron within the gray matter. If the total length of this individual axon is divided by the sum of the lengths of all axons within this volume, the relative axonal density is obtained. It quantifies the axonal contribution of an individual neuron to the

Table 1. Quantitative Anatomy of the Mouse Cerebral Cortex

<i>Measured quantities</i>	
<i>a</i> : Volume (iso- and allocortex)	$2 \times 87 \text{ mm}^3$
<i>b</i> : No. of sensory input fibers	$<10^6$
<i>c</i> : Density of neurons	$9.2 \times 10^4/\text{mm}^3$
<i>d</i> : Percent pyramidal cells	85%
<i>e</i> : Density of synapses	$7.2 \times 10^8/\text{mm}^3$
<i>f</i> : Percent type I synapses	89%
<i>g</i> : Percent synapses on spines	75%
<i>h</i> : Density of axons	$4 \text{ km}/\text{mm}^3$
<i>i</i> : Density of dendrites	$0.4 \text{ km}/\text{mm}^3$
<i>j</i> : Length of axonal tree	10–40 mm
<i>k</i> : Length of dendritic tree	4 mm
<i>l</i> : Range of axonal tree (pyramidal cell)	1 mm
<i>m</i> : Range of dendritic tree	0.2 mm
<i>Deduced quantities</i>	
<i>n</i> : (<i>a</i> , <i>c</i>) Total no. of neurons	1.6×10^7
<i>o</i> : (<i>c</i> , <i>e</i>) Synapses/neuron	8000
<i>p</i> : (<i>e</i> , <i>h</i>) Synapses/length of axon	200/mm
<i>q</i> : (<i>h</i> , <i>j</i> , <i>l</i>) Relative density of axons (pyramidal cells)	10^{-5}
<i>r</i> : (<i>i</i> , <i>k</i> , <i>m</i>) Relative density of dendrites (pyramidal cells)	10^{-3}
<i>s</i> : (<i>p</i> , <i>q</i> , <i>r</i>) Probability of synapses between two pyramidal cells 0.2–0.3 mm apart	0 synapses, $p = 0.9$ 1 synapse, $p = 0.09$ 2 synapses, $p = 0.004$

Conclusions

- t*: (*b*, *n*) No. of neurons \gg no. of input fibers
- u*: (*d*, *f*, *g*) Most connections between neurons of one kind
- v*: (*f*) Most connections excitatory
- w*: (*o*, *s*) Great divergence and convergence
- x*: (*s*) Connections very weak
- y*: (*u–x*) Mixing machine
- z*: (*t*, *u*, *g*) Memory rather than computation
- z'*: (*y*, *z*) Associative memory with formation of cell assemblies

Modified from Braitenberg and Schüz (1998). The numbers represent measurements on light and electron micrographs from the cortex of the mouse (*a–m*), quantities that can be deduced from those measurements (*n–s*), and conclusions that can be drawn for the connectivity (*t–x*), as well as the functional interpretation (*y–z'*). The letters in parentheses indicate from which other quantities the corresponding quantity or conclusion can be derived.

neuropil within which it ramifies and is a measure for the intermingling of axons belonging to different neurons. The *relative density of dendrites* (r) is derived similarly by dividing the dendritic length of an individual neuron by the sum of the lengths of all dendrites present in the volume within which the dendritic arbor ramifies.

The *probability of synapses between two pyramidal cells* (s) is determined on the basis of $p-r$ and the assumption that a synapse between two neurons is made wherever the axon of one touches a dendrite of the other. The probability of this occurrence decreases with distance, i.e., with decreasing overlap between the two neurons.

From these data, some properties of cortical connectivity can be inferred (Table 1, $t-x$). A functional interpretation is also given in Table 1 ($y-z'$).

In short, the network of the cortex consists mainly of one type of neuron, the pyramidal cells (including the spiny stellate cells), which are connected by excitatory synapses. Most of these synapses are located on dendritic spines. Since synapses on spines are assumed to be modifiable in strength, one may conclude that one of the main tasks of the cortex is storage of information.

Nonpyramidal cells (about 15%) are interspersed among the pyramidal cells. They are inhibitory. Pyramidal and most kinds of nonpyramidal cells connect onto each other. The inhibitory neurons, however, contribute only about 11% of the synapses in the cortex. The number of synapses contributed by input fibers is small compared with the number of synapses that connect cortical neurons to each other. Most pyramidal cells have a local axonal tree, which connects them to other neurons in their neighborhood, as well as a far-reaching axonal tree, which, in most cases, connects them to another region of the cortex through the white matter.

The low relative axonal and dendritic densities have interesting implications. In the mouse, the local axonal tree of an individual pyramidal cell is interwoven with the dendrites of 10^5 other neurons and competes with axons from approximately 5×10^5 other neurons ramifying in the same territory. Geometric considerations suggest that multiple synapses (more than 2 or 3 between any pair of neurons) account for only a small percentage of the whole synaptic population of a neuron. This makes it probable that the 8000 synapses of a pyramidal cell connect to thousands of different neighbors. Thus, theories based on a probabilistic connectivity, as sketched in Figure 3C, seem to be appropriate.

For the most part, the knowledge gained from the mouse cortex applies to mammals in general. Some of the numbers in Table 1 differ with brain size according to known rules ($a-c$, $j-m$); others seem to be constant ($d-i$). For some, the dependence on brain size is not yet clear ($q-s$), but the differences and uncertainties are not strong enough to affect the conclusions about the basic connectivity and function of the cortex.

Basic Function of the Cortex

This description of cortical structure fits Hebb's theory of cell assemblies remarkably well. This theory postulates that meaningful events are represented by groups of neurons that are connected more strongly to each other than to other neurons. These groups are formed through a learning process (see HEBBIAN SYNAPTIC PLASTICITY). The learning is assumed to strengthen the connections between neurons that are often activated together, a process known as *associative storage* (see ASSOCIATIVE NETWORKS).

For such cell assemblies to form, a large number of neurons of the same type must be connected into a network. What is crucial is an initial connectivity that is sufficiently rich to allow as many constellations of neuronal activity as possible to be detected and learned in the connections. The fact that the individual pyramidal cell seems to strive toward a large number of different synaptic neighbors, together with the large mass of corticocortical fibers,

indicates that the cortex is well suited for this task. The excitatory and modifiable synapses that are implicit in Hebb's theory are also present in the cortex. The fact that individual neurons are weakly connected through one or only a few synapses implies that only correlated activity of many neurons can activate another neuron. This implication is also in agreement with the theory of cell assemblies.

The linkage of the structure of the cortex with the theory of cell assemblies has led to more precise formulations of this concept. It permits estimates of the storage capacity of the cortex and the size and internal structure of cell assemblies (Palm, 1993) and allows concrete ideas to develop about the regulation of their dynamics through the hippocampus (Miller, 1991) or the striatum (Miller and Wickens, 1991).

Other Basic Principles of Connectivity

The connectivity of the cortex contrasts with that of other parts of the brain. The *cerebellar cortex* differs from the cerebral cortex primarily in its complete lack of positive feedback connections within the cortex and in its strict geometric order, which indicates computation along quasi-one-dimensional lines (Figure 1). These and other structural features suggest that the cerebellar cortex plays a role in the detection of sequences, a view supported by recent experimental evidence (Braitenberg et al., 1997). In addition, the cerebellar cortex stands out by virtue of its very large number of granule cells, which exceeds the number of neurons in the rest of the brain. Their small size and large numbers suggest a combinatorial richness of mossy fiber inputs that may be essential for the subtleties of motor coordination (Arbib et al., 1998).

Another contrasting network is the *striatum*. Although the cerebral cortex and the striatum have a number of features in common, a fundamental difference is that the cortex operates primarily on the basis of mutual excitation of neurons, while mutual inhibition seems to play an essential role in the striatum. There, more than 90% of the neurons (the medium spiny neurons) are GABAergic and, in addition to providing the output, make axon collaterals within the striatum. Thus, while the dominating principle in the cortex seems to be cooperation, the internal connectivity of the striatum suggests competition between neurons (Miller and Wickens, 1991; Wickens and Oorschot, 2000).

A basic difference also exists between cortex and *thalamus*. As in the cortex, most of the neurons in the thalamic relay nuclei are excitatory. However, in contrast to the neurons in the cortex, the lack of axon collaterals within the relay nuclei indicates that the excitatory neurons in the thalamus do not make synapses with each other (Steriade, Jones, and Llinás, 1990).

Homogeneity Versus Heterogeneity of Brain Structures

Some parts of the brain show local anatomical variations; others do not. For example, the cerebellar cortex exhibits no obvious differences over its whole extent.

In the striatum, compartments known as *striosomes* can be detected with histochemical methods. They differ from the surrounding matrix not only in their histochemistry but also in their input-output organization. The medium spiny neurons, the main cell type, are located in both regions but tend to avoid crossing the boundaries between striosomes and matrix (Penny, Wilson, and Kitai, 1988).

In the cerebral cortex, local differences on a large scale are the basis of its division into areas. These differences are of a kind that influences the statistics of the connectivity between neurons (density of neurons in the various layers; size, shape, and density of dendritic and axonal fields). Cortical areas can differ along two coordinates: the coordinate of hierarchy and the coordinate of modality. Hierarchy is reflected, for example, in the degree of myelination. Overall myelination diminishes with increasing distance

from the primary sensory and motor areas. The existence of a well-developed layer IV in primary sensory fields is another example. In addition, the size of the local axonal fields of pyramidal cells, as well as the size of basal dendritic trees, increases with distance from the periphery, at least in the visual system (for details, see Schüz and Miller, 2002). The coordinate of modality is reflected in the fact that primary areas not only share commonalities that set them apart from higher cortical areas, but also differ one from the other. Such differences characterize the most conspicuous neocortical areas: the primary visual cortex in primates, with its particularly prominent layering; the somatosensory cortex in some rodents, with the barrel field (see SOMATOSENSORY SYSTEM); and the primary motor cortex, with its huge corticospinal neurons.

Some histological methods, particularly the cytochrome oxidase method and tracer methods, also reveal substructure *within* cortical areas—usually, bands or patches a few hundred microns in diameter, which are best known in the visual system (Arbib et al., 1998; see also OCULAR DOMINANCE AND ORIENTATION COLUMNS; VISUAL CORTIX: ANATOMICAL STRUCTURE AND MODELS OF FUNCTION). To a large extent, this phenomenon is the result of the alternating insertion of input bundles from different sources (via the white matter) into a relatively homogeneous internal network. Intracortical axonal ramifications of pyramidal cells, however, may also exhibit a patchy distribution, particularly in large brains. These patches connect columns with similar functional properties. In contrast to the situation in the striatum, the dendritic trees seem to freely cross borders between columns, an arrangement that suggests a maximization of diversity in neuronal connections (Malach, 1994; see also Dinse and Schreiner in Schüz and Miller, 2002).

Road Map: Mammalian Brain Regions

Related Reading: Basal Ganglia; Cerebellum and Motor Control; Cortical Hebbian Modules; Neocortex: Basic Neuron Types; Visual Cortex: Anatomical Structure and Models of Function

References

- Arbib, A. M., Érdi, P., and Szentágothai, J., 1998, *Neural Organization: Structure, Function, and Dynamics*, Cambridge, MA: MIT Press. ♦
- Blinkov, S. M., and Glezer, I. I., 1968, *The Human Brain in Figures and Tables: A Quantitative Handbook*, New York: Plenum.
- Braitenberg, V., 2001, Brain size and number of neurons: An exercise in synthetic neuroanatomy, *J. Computational Neurosci.*, 10:71–77.
- Braitenberg, V., and Schüz, A., 1998, *Cortex: Statistics and Geometry of Neuronal Connectivity* (2nd edition of *Anatomy of the Cortex: Statistics and Geometry*), Berlin: Springer-Verlag. ♦
- Braitenberg, V., Heck, D., and Sultan, F., 1997, The detection and generation of sequences as a key to cerebellar function: Experiments and theory, *Behav. Brain Sci.*, 20(2):229–277.
- Frahm, H. D., Stephan, H., and Stephan, M., 1982, Comparison of brain structure volumes in insectivora and primates, I: Neocortex, *J. Hirnforsch.*, 23:375–389.
- Haug, H., 1986, History of neuromorphometry, *J. Neurosci. Methods*, 18:1–17.
- Malach, R., 1994, Cortical columns as devices for maximizing neuronal diversity, *Trends Neurosci.*, 17(3):101–104.
- Miller, R., 1991, *Cortico-hippocampal Interplay and the Representation of Contexts in the Brain*, Berlin: Springer. ♦
- Miller, R., and Wickens, J. R., 1991, Corticostriatal cell assemblies in selective attention and in representation of predictable and controllable events: A general statement of corticostriatal interplay and the role of striatal dopamine, *CINS (Concepts Neurosci.)*, 2:65–95.
- Mitchison, G., 1992, Axonal trees and cortical architecture, *Trends Neurosci.*, 15:122–126.
- Palm, G., 1993, On the internal structure of cell assemblies, in *Brain Theory: Spatio-temporal Aspects of Brain Function* (A. Aertsen, Ed.), Amsterdam: Elsevier, pp. 261–270.
- Penny, G. R., Wilson, C. J., and Kitai, S. T., 1988, Relationship of the axonal and dendritic geometry of spiny projection neurons to the compartmental organization of the neostriatum, *J. Comp. Neurol.*, 269:275–289.
- Ringo, J. L., Doty, R. W., Demeter, S., and Simard, P. Y., 1994, Time is of the essence: A conjecture that hemispheric specialization arises from interhemispheric conduction delay, *Cereb. Cortex*, 4:331–343.
- Rouiller, E. M., Liang, F., Babalian, A., Moret, V., and Wiesendanger, M., 1994, Cerebellothalamocortical and pallidothalamocortical projections to the primary and supplementary motor cortical areas: A multiple tracing study in macaque monkeys, *J. Comp. Neurol.*, 345:185–213.
- Schüz, A., and Miller, R. (Eds.), 2002, *Cortical Areas: Unity and Diversity*, London: Taylor & Francis.
- Steriade, M., Jones, E. G., and Jlinás, R. R., 1990, *Thalamic Oscillations and Signaling*, New York, Chichester: John Wiley & Sons.
- White, E. L., 1989, *Cortical Circuits: Synaptic Organization of the Cerebral Cortex—Structure, Function and Theory*, Boston: Birkhäuser. ♦
- Wickens, J., and Oorschot, D. E., 2000, Neural dynamics and surround inhibition in the neostriatum, in *Brain Dynamics and the Striatal Complex* (R. Miller and J. Wickens, Eds.), Australia, Canada: Harwood Academic Publishers.

Neuroethology, Computational

Dave Cliff

Introduction

Over the past decade, a number of neural network researchers have used the term *computational neuroethology* to describe a specific approach to neuroethology. *Neuroethology* is the study of the neural mechanisms underlying the generation of behavior in animals, and hence it lies at the intersection of neuroscience (the study of the nervous systems) and ethology (the study of animal behavior); for an introduction to neuroethology, see Simmons and Young (1999). The definition of computational neuroethology is very similar, but is not quite so dependent on studying animals: animals just happen to be biological *autonomous agents*. But there are also non-biological autonomous agents, such as some types of robots and some types of simulated embodied agents operating in virtual

worlds. In this context, autonomous agents are self-governing entities capable of operating (i.e., coordinating perception and action) for extended periods of time in environments that are complex, uncertain, and dynamic. Thus, computational neuroethology can be characterized as the attempt to analyze the computational principles underlying the generation of behavior in animals and in artificial autonomous agents. For the sake of brevity, in the rest of this article autonomous agents will be referred to simply as agents, and computational neuroethology will be abbreviated to CNE.

CNE can be distinguished from classical computational neuroscience by its increased emphasis on studying the neural control of behavior within the context of neural systems that are both embodied and situated within an environment. The “computational” nature of CNE comes not so much from treating neural systems as

inherently computational devices, but rather in the use of sophisticated computer-based simulation and visualization tools for exploring issues in neuroethology.

Put most simply, CNE involves the use of computational modeling in trying to understand the neural mechanisms responsible for generating “useful” behaviors in an agent. The word useful is rather imprecise; it is more common to talk of *adaptive* behaviors. In the ethology literature, an adaptive behavior is usually defined as a behavior that increases the likelihood that an animal will survive long enough to produce viable offspring. Often implicit in this definition is the assumption that the animal’s environment is sufficiently unforgiving (or hostile) that if the animal does nothing, it will die before it can reproduce. In studying artificial agents, the utility of behavior is frequently evaluated by less harsh criteria, such as observed behaviors scored according to some metric that indicates how close they come to satisfying some set of performance objectives or criteria.

Neural networks that generate adaptive behavior should not be confused with adaptive neural networks, where connection strengths may alter as a result of experience. Adaptation or plasticity may itself give rise to new or improved adaptive behaviors, but there are many cases of adaptive behaviors that are genetically determined (e.g., “hardwired” behaviors such as reflexes and instincts).

When CNE is approached in the context of adaptive behavior research, it becomes clear that the neural system is just one component in the *action-perception cycle*, where an agent’s actions may alter what information it perceives concerning its environment, and where those alterations in perceived information may lead to changes in the agent’s internal state, and where those changes in state may in turn affect further actions, thereby affecting what information is subsequently perceived, and so on. Thus, crucially, the agent’s nervous system, body, and environment all combine to form a tightly coupled dynamical system. This is a notion long stressed by Arbib:

In speaking of human perception, we often talk as if a purely passive process of classification were involved—of being able, when shown an object, to respond by naming it correctly. However, for most of the perception of most animals and much of human behavior, it is more appropriate to say that the animal perceives its environment to the extent that it is *prepared to interact* with that environment in some reasonably structured fashion (1972, p. 16).

As defined thus far, CNE may not seem to be particularly distinguishable from most work in neural network research. After all, many people in computational neural network research might argue that their work will, ultimately, lead to understanding of the neural mechanisms underlying the generation of (some) adaptive behaviors. For example, face recognition is an adaptive behavior in humans and could probably be classed as an adaptive behavior in, say, a security robot. So why can’t a backpropagation network that learns to distinguish between photographs of human faces (for example) be classed as work in CNE?

Motivations

Typically, artificial neural network models employ homogeneous groups of highly idealized and simplified neuron models (called *units*), connected in a regular fashion, that exhibit some form of “learning” or adaptation. The large majority of such models can be described in essence as mapping or transforming between representations: input data are presented to the network in a particular format, and the network is judged successful when its outputs can be interpreted as a correct representation of the results of performing the desired transformation. In almost all cases, the input and output representation formats are prespecified by the experimenter

(although this is not entirely true of unsupervised learning networks, and there are a number of artificial neural network models that draw inspiration from biological data in their choice of input and output representations). If such networks are to be employed in artificial agents, or are to be of use in understanding biological agents, then this can only be so under the (often unspoken) assumption that, eventually, it will be possible to assemble a “pipeline” of such input-output transducer networks that links sensory inputs to motor outputs, and produce adaptive behavior. The most significant issue here is the heavy dependence on a priori intermediate representations, which may not be justifiable: neural sensorimotor pathways generating adaptive behaviors might not be neatly partitioned into representation-transforming modules; such pathways may not exhibit any patterns of activity identifiable as a representation in the conventional sense, and even if they do, there is no guarantee that they will be in strong accordance with representations chosen a priori by modelers.

This should not be mistaken for an argument against representation or for a denial of the vital role played by internal states in the generation of adaptive behaviors; it is simply an awareness of the dangers of being misled by a priori notions of representation. One of the safest ways of avoiding these dangers is to model, as far as is possible, *entire* sensorimotor pathways (i.e., the complete sequence of neural processing, from sensory input to motor output) involved in the generation of adaptive behavior. This requires that the agent be studied *while situated in an environment*: most sensorimotor processing for adaptive behavior involves dynamic interaction with the environment, and a situated agent is part of a closed-loop system, because certain actions can affect subsequent sensory inputs. Thus, the sensorimotor pathway should not be viewed as a “pipeline” transforming from a given input representation to a desired output representation, but rather as one link in the action-perception cycle.

When such an approach is adopted, the true nature of the representations and processing necessary for the generation of relatively complex adaptive behaviors is more likely to be revealed, and the validity of any a priori assumptions is clarified.

Naturally, it is beyond the state of the art to attempt to model complete sensorimotor pathways in humans or other large mammals, but experimental work in the neuroethology literature provides a wealth of data from less intellectually able animals, such as arthropods (the animal class that includes insects, spiders, and crustacea), amphibia, and other “simple” vertebrates such as eels or salamanders. Such animals are used as the domains of study in some CNE research, but in other work, simple idealized models are rigorously studied, in a manner akin to Galileo’s models of perfect spheres rolling down inclined frictionless planes.

The argument that a priori commitment to certain representations or architectures for sensorimotor processing can lead to surprisingly wrong conclusions can be illustrated by reference to a classic series of thought experiments devised by Braitenberg (1984). Braitenberg described specifications for a series of simple mobile vehicles operating in a world with simplified kinematics. The series of vehicles starts with an elementary device that performs primitive heat-seeking behavior and progresses through vehicles that exhibit positive or negative taxes (i.e., orientation toward or away from a directional stimulus) and primitive forms of learning, pattern detection, and movement detection, culminating in vehicles that exhibit chaotic dynamics and predictive behavior. The internal control mechanisms of all the vehicles are rigorously minimal. The simpler vehicles contain nothing more than wires connecting sensors to actuators, while the more advanced ones employ nonlinear threshold devices with delays and pseudo-Hebbian adaptation.

Braitenberg notes that the psychological language indicative of intentional mental states has compelling intuitive appeal in describing the observed behavior of the vehicles. He ascribes *fear, aggression, love, values and taste, rules, trains of thought, free will,*

foresight, *egotism*, and *optimism* to his vehicles. But he also demonstrates that whereas such terms may be very useful at the level of description of an external observer, the internal causal mechanisms could be surprisingly simple and, crucially, could contain nothing that can meaningfully be said to either “represent” or “implement” these intentional mental states. That is to say, the intentionality is in the eye of the beholder, not in the workings of the agent. For further discussion of these issues, see Cliff and Noble (1997).

While Braitenberg’s vehicles are nothing more than thought experiments, they provide insight into possible organizational principles in natural and artificial creatures, and demonstrate the limits of applicability of intentional terminology. Further discussion of the utility of agent models in biology can be found in Dean (1998).

To summarize, research in CNE can be characterized as placing increased emphasis on modeling entire adaptive behavior—generating sensorimotor pathways in embodied agents, where those agents are situated in environments that supply sensorimotor feedback. Such an approach lessens the chances of making untenable assumptions concerning issues of representation and processing. Moreover, in order to study such pathways where there is reliable biological data, it may often be necessary to focus attention on relatively simple animals, such as arthropods or amphibia. For further discussion of the rationale for CNE, see Beer (1990) and Cliff (1990).

It is important to note that there is a tradition of related work in the artificial neural network literature. Research on reinforcement learning for control tasks is most close; see REINFORCEMENT LEARNING IN MOTOR CONTROL.

Selected Current Research Projects

Two specific longstanding CNE research programs are discussed in this section: the work of Arbib’s research group on visuomotor behavior in simple vertebrates, and Beer’s work on the neural foundations of adaptive behavior in even simpler agents (i.e., cockroaches) and in abstract idealized agents. Before delving into these bodies of work, it is useful to consider where they sit in the CNE canon, and to point to CNE research that resides at other points in that space.

For the purposes of framing, there are three major axes along which work in CNE can be categorized. In no particular order, they are the degree of reliance on computer software simulation, the degree of concentration on a specific animal species or class, and the extent to which semiautomated design techniques are employed.

The degree of reliance on computer simulation in CNE research projects varies from the complete, where all work is carried out using software simulations, to the minimal, where the CNE model takes the form of an operational physical robot, with the model neurons (individually or at the network level) being constructed from electronic circuits. Examples of the former include work by Arbib (1987) and Beer (1990), while many examples of the latter are discussed by Webb (2001). Note also that Beer went on to use robot platforms in continuations of his work that was initially software-only (Beer et al., 1992) while Arkin (see REACTIVE ROBOTIC SYSTEMS) made similar use of Arbib’s works. A comprehensive review of the merits of using physical robots (rather than computer software) as simulations of animals has recently been published by Webb (2001), with copious references to work in this field; see also BIOLOGICALLY INSPIRED ROBOTICS.

The extent to which CNE research projects concentrate on a specific animal species or class varies from, at one extreme, CNE studies of one specific species (e.g., Beer, 1990) through generic CNE studies of several species of animals within the same order (e.g., Arbib’s 1987 work on anuran visual control of action), to the other extreme, where neural mechanisms underlying the generation

of adaptive behavior in wholly abstract and idealized agents is explored within the CNE methodology (e.g., Beer, 2002).

Finally, with the continuing falls in the real cost of processor power and memory and disk storage, there has been an increased tendency over the past decade to move away from hand-designed computational/robot models and toward models that are the product of automated or semiautomated design processes. The use of evolutionary computation techniques such as genetic algorithms in particular (see EVOLUTION OF ARTIFICIAL NEURAL NETWORKS) has proved fruitful. At the hands-on extreme, there are CNE models where each artificial neuron’s parameters (e.g., its time constants, thresholds, and connectivity to other components) are specified by the designer of the model (see, e.g., Arbib, 1987; Beer, 1990), whereas at the hands-off extreme the modeler sets up a (usually truly vast) space of possible network designs and then uses an evolutionary search process to identify points in that design space that best satisfy some performance metric (i.e., the fitness evaluation function). Examples of this latter approach include Ijspeert (2001) and Beer (2002).

Computational Frogs, Toads, and Salamanders

Probably the most mature body of work in CNE is the research program led by Arbib for two decades on a family of models of visually mediated behavior in simple vertebrates. In the initial years of this project the focus was on visuomotor activity in frogs and toads; see Arbib’s 1987 paper for a review of the project with peer commentary and his 1997 publication for a discussion of how this work integrates with studies of monkeys and of rats. Arbib named his simulation model *Rana computatrix*, the computational frog, in homage to W. Grey Walter’s seminal *Machina Speculatrix* robots from the 1950s.

The *R. computatrix* models are faithful to the known biology, and there is an interplay between the experimental and theoretical work: an initial first approximation model was extended and refined in a number of stages, leading to a family of models.

Arbib’s approach involves defining a number of functional *schemas*. Schemas can be modeled by interacting layers of neuron-like elements or by nets of intermediate-level units; the network models can be related to experimental data concerning neural circuitry, and the development process iterates (Arbib, 1987, p. 411 ff.). Further details can be found in SCHEMA THEORY.

The primary focus in the *R. computatrix* models has been on how frogs and toads use vision to detect and catch prey, in environments that include obstacles and barriers. Arbib has developed a series of schema-based models that account for depth perception as interaction between accommodation and binocular clues, and at the lowest level the schemas are plausibly based on known details of the relevant neurological data.

One of the more striking results from this work, with reference to Marr’s well-known theory of vision, is the indication that, at least in frogs and toads, there are different perceptual mechanisms for different visual stimuli. That is, the depths to prey and to barriers are extracted from the optic array by different processing channels and are integrated in the sensorimotor pathways much later than Marr’s theory might suggest. Arbib and Liaw (1995) went on to demonstrate how lessons learned from the *R. computatrix* project could inform the design of visually guided robot systems.

In more recent work, Ijspeert and Arbib (2000) have reported on experiments in which a sophisticated simulation of a 3D multi-segmented biomechanical model of a salamander’s body is controlled by a complex neural network model. The network is composed of separate central pattern generators (CPGs; see LOCOMOTION, VERTEBRATE) for the body and the limbs, each of which may be activated and modulated by descending tonic inputs. Ijspeert and Arbib use this simulation system to explore the neural

circuitry underlying the generation of visually steerable salamander locomotion behaviors in water and on land. One notable aspect of this work in relation to the earlier studies of anuran circuitry is that, while the gross morphology of the CPG circuits is decided by the experimenters, a genetic algorithm is used to determine the fine details of the CPG circuits' internal connectivity and parameter values, the intersegmental coupling, and the coupling between the limb CPG and the body CPG. Thus, unlike the hands-on incremental modeling employed in the *R. computatrix* models, the salamander model is the product of a semiautomatic evolutionary design process.

Computational Cockroaches and Vehicles Redux

Beer's 1990 book, *Intelligence as Adaptive Behavior*, contains both methodological arguments for CNE and details of experimental work on his model of a computational cockroach, *Periplaneta computatrix*, which is a simulated hexapod agent embedded in an environment, inspired by neuroethological studies of the cockroach *Periplaneta americana*. The real cockroach uses chemotaxis as one of several strategies to locate food sources. If its path along an odor gradient is blocked by an obstacle, it performs stereotyped "edge-following" behavior. The artificial cockroach is controlled by a heterogeneous neural network that was inspired by biological data and has been used to study issues in locomotion, guidance, and behavioral choice.

The primary external sensory input was simulated chemosensory information: patches of food in the environment gave off odor gradients detectable under an inverse square law relating distance to odor intensity. The neural nets also received mechanosensory input from proprioceptors in the limbs and tactile sensors that signaled the presence of food under the mouth. The simulation model included elementary kinematics: if the artificial cockroach failed to adopt a stable position for a sufficient length of time, it fell down.

Results from the simulation sessions demonstrated behavior in the model that was highly similar to behavior in the real animal, and Beer subsequently performed "lesion" experiments by selectively deleting connections or units from the *P. computatrix* control network. Again, the results obtained with the artificial system were in agreement with the biological data.

P. computatrix was inspired by biological data, but it was not intended as a biological model. The various behaviors were generated by heterogeneous neural networks. The neuron model employed by Beer was more faithful to biology than many of the "formal neurons" used in conventional artificial neural network research: the units involved differential equations modeling membrane potentials, which gave his model neural assemblies a rich intrinsic dynamics. For further details, see LOCOMOTION, INVERTEBRATE.

The central focus in Beer's (1990) work was on designing architectures composed from such neural units that could act as controllers for the various behaviors that *P. computatrix* should exhibit. Thus, there was no treatment of learning in the initial body of work on the cockroach. Subsequently, Beer reported on work that extended the original *P. computatrix* simulation model, testing it by allowing it to control walking in a real hexapod robot (Beer et al., 1992).

In the robot implementation, the control network was still simulated (i.e., the units in the neural network were not realized physically), but the sensorimotor connections to the artificial neural network were interfaced to physical sensors and actuators by means of analog-digital and digital-analog converters. Beer et al. reported that in all cases, the response of the physical robot was highly similar to that previously observed in simulation. The implementation did, however, reveal one problem in the controller that had not been examined in the simulation. This problem, involving disturbances in the crossbody phasing of the legs, was easily rectified,

but nevertheless this demonstrates that simulation models cannot be trusted to perfectly replicate any physical implementation they may ultimately be intended for.

For a wider unified perspective on this work, see LOCOMOTION, INVERTEBRATE; VISUOMOTOR COORDINATION IN FROG AND TOAD; and LOCOMOTION, VERTEBRATE.

Subsequent to his work on *P. computatrix*, one line of research that Beer has pursued is, in comparison, radically simplified, divorced from any specific animal, and yet in its simplicity it reaches to the core of fundamental issues in cognitive science and adaptive behavior research. Rather than be constrained (and potentially confused) by biology, Beer (2002) developed a series of simple idealized embodied and embedded model agents, each of which is capable of "minimally cognitive" behaviors. Beer defines a minimally cognitive behavior as one that is just above the threshold for raising issues of genuine interest to cognitive science (see also COGNITIVE MODELING; PSYCHOLOGY AND CONNECTIONISM).

Beer's minimal agents exist in a two-dimensional world, but can only move along a bounded horizontal baseline. Various geometric shapes such as circles or diamonds drop from above, toward the agent's baseline. In each experiment, the intention is that the minimal agents use their sensors to detect the nature of whatever geometric shape or shapes is or are currently falling toward it, and thereby generate behavior "appropriate" to the current situation. The definition of appropriate behavior depends on the experiment but may, for example, be as apparently trivial as "intercept circular objects and avoid diamond-shaped ones." To achieve this sensorimotor coordination, each minimal agent is equipped with a small continuous-time recurrent neural network (CTRNN) (see RECURRENT NETWORKS: LEARNING ALGORITHMS).

The CTRNN for each minimal agent has a small number (e.g., seven) of fixed-orientation ray-casting "visual" proximity sensors (each of which sends a straight limited-length ray out at a particular angle to the agent's body and reports on how far the ray traveled before it intercepted an object, if at all). Each sensor feeds onto a small number (e.g., five) of fully interconnected "interneurons," and all of these in turn feed onto a small number (e.g., two) of "output" neurons—one for moving to the left and one for moving to the right. Thus, a typical minimal agent may have 14 units and perhaps 70 connection weights in its CTRNN.

Any particular design for a CTRNN sensorimotor controller for one of Beer's minimal agents specifies the time-constant, bias, gain, and input weights for each neuron. Rather than design appropriate networks by hand, Beer employs a "hands-off" genetic algorithm to explore a very large space of possible network designs, evaluating each design on a measure of its observed behavior. To halve the size of the search space, Beer imposed a bilateral symmetry requirement. Other than this enforcement of symmetry, there is very little a priori commitment to any particular CTRNN solution. Over a reasonably small evolutionary experiment (e.g., 2,000 generations with a population size of 100), minimal agents evolve that reliably score well on the experiment's evaluation function, and that also generalize well to situations not encountered in the evolutionary adaptation period.

So far, so simple. Yet, in a series of papers published since 1996, Beer and his colleagues have reported on the evolution of CTRNNs for sensorimotor control in minimally cognitive agents that have been evaluated on the basis of their ability to perform a variety of increasingly sophisticated behaviors. These behaviors include orientating toward and reaching for objects, discriminating between objects, judging the passability of openings relative to the agent's own body size, discriminating between visible parts of the agent's body and other objects in the agent's environment, predicting and remembering the future location of falling objects so that they can later be intercepted "blind," and switching attention between multiple objects as they fall. All of these behaviors are achieved with the same simple agent CTRNN architecture outlined above.

This array of cognitively interesting behaviors achieved by Beer's minimally cognitive agents prompts the question of what, precisely, is happening at the mechanistic level within the evolved CTRNNs to generate these behaviors. And at this point we return to the arguments and issues explored in the opening sections of this article. Beer presents concrete analyses of the CTRNNs of these agents, demonstrating a full understanding of their mechanistic activity from a *dynamical systems* perspective; and yet, as he points out, this analysis is of little or no use in attempts at elucidating an understanding from a *computational* (and hence *representational*) perspective: there is nothing readily identifiable in the CTRNNs that represents a circle or a diamond, or the action of intercepting or of avoiding. Rather, a full explanation of the behavior exhibited by one of Beer's minimally cognitive agent's CTRNNs can only be given in the context of the dynamics of that agent's embodiment and of the environment in which it is situated. See Beer (2002) and Cliff and Noble (1997) for further details.

Discussion

Computational neuroethology studies neural mechanisms that generate adaptive behaviors, and hence requires that embodied agents be studied within the situated context of their environmental and behavioral niches.

From the descriptions given in this article, some patterns emerge: the animal-specific CNE projects mentioned are dependent on the availability of fairly detailed neuroethological data. Such data invariably come from invasive in vivo experimentation, and the neuroanatomy of "simpler" animals such as arthropods or simpler vertebrates is particularly amenable to such techniques. For arthropods in particular, certain neurons performing particular functions are readily locatable in different individual animals of the same species. Although there are manifest obstacles preventing the collection of such data from more complex vertebrate subjects, research in these areas is making significant progress. Furthermore, by definition, any truly *general* principles underlying the neural generation of adaptive behaviors are those that are common to a number of species, so only cross-species studies will help identify general principles (Cliff, 1990, p. 37).

Yet surely the most general principles of all are those that apply to all agents within a certain class of cognitive or behavioral niches, regardless of the hardware (or software) that those agents are implemented in. In this respect, Beer's minimally cognitive agents are highly cogent. Until the representation-manipulating explanatory language that has traditionally been brought to bear on the supposed neural behavior-generating mechanisms of "complex" animals (including humans) can be demonstrated to be routinely applicable to "simpler" agents (including Beer's *vehicle*-like minimal

agents), the rigor and limits of that explanatory language will remain in doubt.

Road Map: Neuroethology and Evolution

Related Reading: Action Monitoring and Forward Control of Movements; Arm and Hand Movement Control; Eye-Hand Coordination in Reaching Movements; Motor Cortex: Coding and Decoding of Directional Operations; Pursuit Eye Movements; Reaching Movements: Implications for Computational Models; Sensorimotor Learning; Vestibulo-Ocular Reflex

References

- Arbib, M. A., 1972, *The Metaphorical Brain: An Introduction to Cybernetics as Artificial Intelligence and Brain Theory*, New York: Wiley-Interscience. ♦
- Arbib, M. A., 1987, Levels of modelling of mechanisms of visually guided behavior, *Behav. Brain Sci.*, 10:407–465.
- Arbib, M. A., 1997, From visual affordances in monkey parietal cortex to hippocampal-parietal interactions underlying rat navigation, *Philos. Trans. R. Soc. Lond. B*, 352:1429–1476.
- Arbib, M. A., and Liaw, J., 1995, Sensorimotor transformations in the worlds of frogs and robots, *Artif. Intell.*, 72:53–79.
- Beer, R. D., 1990, *Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology*, New York: Academic Press. ♦
- Beer, R. D., 2002, The dynamics of active categorical perception in an evolved model agent, *Behav. Brain Sci.*, in press; available: <http://vorlon.cwru.edu/~beer/Papers/BBSpaper.pdf>.
- Beer, R. D., Chiel, H. J., Quinn, R. D., Espenschied, K., and Larsson, P., 1992, A distributed neural network architecture for hexapod robot locomotion, *Neural Comput.*, 4:356–365.
- Braitenberg, V., 1984, *Vehicles: Experiments in Synthetic Psychology*, Cambridge, MA: MIT Press/Bradford Books. ♦
- Cliff, D., 1990, Computational neuroethology: A provisional manifesto, in *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90)* (J.-A. Meyer and S.W. Wilson, Eds.), Cambridge, MA: MIT Press/Bradford Books, pp. 29–39.
- Cliff, D., and Noble, J., 1997, Knowledge-based vision and simple visual machines, *Philos. Trans. R. Soc. Lond. B*, 352:1165–1175. ♦
- Dean, J., 1998, Animats and what they can tell us, *Trends Cognit. Sci.*, 2:60–67.
- Ijspeert, A. J., 2001, A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander, *Biol. Cybern.*, 84:331–348.
- Ijspeert, A. J., and Arbib, M. A., 2000, Visual tracking in simulated salamander locomotion, in *From Animals to Animats 6: Proceedings of the Sixth International Conference of the Society for Adaptive Behavior (SAB2000)* (J.-A. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S. W. Wilson, Eds.), Cambridge, MA: MIT Press, pp. 88–97.
- Simmons, P., and Young, D., 1999, *Nerve Cells and Animal Behaviour*, Cambridge, Engl.: Cambridge University Press. ♦
- Webb, B., 2001, Can robots make good models of biological behavior? *Behav. Brain Sci.*, 24(6).

Neuroinformatics

Michael A. Arbib

Introduction

Some define the term "neuroinformatics" as the use of databases, the World Wide Web, and visualization for the storage and analysis of neuroscience data, but we here broaden the definition to include the role of computational models in structuring masses of data. From the perspective of this *Handbook*, the key challenge for neuroinformatics is to integrate insights from synthetic data obtained

from running a model with data obtained empirically from studying the animal or human brain. Moreover, we must integrate all levels from molecules to compartments and neurons up to biological neural networks and on to the behavior of organisms. One must thus maintain an architecture for a federation of empirical databases in which the results from diverse laboratories can be integrated, providing an environment in which we can make quantitative verifiable or disprovable predictions from the model to the database.

Neuroscience integrates anatomy, behavior, physiology, and chemistry. Each vertebrate has hundreds of brain regions, discriminated from other regions by gross anatomy, cytoarchitectonics, input and output connections of the region, and detailed neurophysiology. Then, for a variety of behaviors—whether they be eye movements, aspects of motor control, performance on memory tasks, and so on—we may seek to characterize the regions of the brain that are most involved in such behaviors and then characterize the firing of particular populations of neurons in temporal correlation with different aspects of the task. In such modeling studies, we seek to understand what must be added to the available database on neural responsiveness and connectivity to explain the time course of cellular activity, and the way in which they mediate between sensory data, the animal's intention, and behavior. But we also seek to use insights from these studies to better understand the human brain.

Shepherd et al. (1998) review the contribution to neuroinformatics tools for integrating, searching, and modeling multidisciplinary neuroscience data made by researchers supported by the Human Brain Project, a consortium of U.S. federal research agencies led by the National Institute of Mental Health. Kötter (2001) reviews current issues in the representation, integration, and analysis of neuroscience data from molecular to brain systems levels, including issues of implementation, standardization, management, quality control, copyright, confidentiality, and acceptance, with particular emphasis on integrative neuroinformatics approaches for exploring structure-function relationships in the brain. Arbib and Grethe (2001; hereafter *CtB*) present an integrated approach to neuroinformatics whose delineation will structure much of this article.

Federating a Variety of Databases

A database on the neurochemistry of synaptic plasticity might be constructed as a view of plasticity data within various databases for different brain regions. An atlas of brain regions can be used to structure data both on the location of single cells (a link to a neurophysiology database) and for standardizing slice-based data (such as stains of receptor activity in a neurochemistry database). In short, neuroinformatics requires a *federation* of databases with tools for linking data from diverse databases to answer complex questions (Heimbigner and McLeod, 1985; Liao and McLeod, Chapter 5.1 of *CtB*). Databases in such a federation may be of one or more of the following types.

Article Repositories

Many publishers now offer their journals on-line. Even if articles migrate from linear text to hypertext, such narratives about the data—"This is the recent experiment that I did," "Here is my review," and so on—will often provide the way for humans to get started in understanding what is going on in some domain, even if they will eventually search specific data sets of the kind described below.

Repositories of Empirical Data

This is where we get data from different laboratories and make them generally available by linking each data set to the *protocol* that produced it: information (like that in the Methods section of an article but in a more algorithmic form) on the hypotheses being tested, the experimental methods used, and so on.

Summary Databases

In these databases are assembled the assertions, summaries, hypotheses, tables, and figures that encapsulate the state of knowledge in a particular domain. Assertions in summary databases can be

linked through the database federation not only to primary literature, but also to models or empirical data. However, in many fields, there is no consensus as to just which hypotheses have been firmly established. Different reviewers may therefore assign different confidence levels to different primary data, and these will affect the confidence level of assertions in the summary database.

Model Repositories

These repositories will not only provide access to computational models, but also link each model to empirical and summary databases to provide evidence for hypotheses in the model or data to test predictions from simulation runs made with the model. We have viewed the protocol as a way to delineate the structure of an experiment. When we design a model, we often give an interface that mimics the protocol so that operations on the model capture the manipulations the experimenter might have made on the nervous system. This makes it easy for somebody who is not expert in modeling to nonetheless evaluate a model by seeing how it runs in a variety of situations.

NeuroCore and Time Series Databases

Neuroscience provides many examples of time series data. For example, the time series data for a single experiment for a study of classical conditioning of a rabbit's eyeblink (see CEREBELLUM AND CONDITIONING) might include for each trial traces showing the movement of the eyelid and a display of firing of a single neuron. The firing data might then be aggregated across trials in a histogram. This example motivates the database issues: How do we store time series? How do we register data with time stamps to facilitate interesting processing of sets of data? How do we link each firing pattern to the position in the brain of the neuron it is taken from? We again see the need to store the protocol as well as, making explicit what hypotheses were being tested, what experimental methods were used, and what conditions were required to elicit each data set.

NeuroCore (Grethe, Mureika, and Merchant, Chapter 3.2, in *CtB*) provides a core schema, an extendible object-relational database schema implemented in Informix. The schema (structure of data tables, etc.) for each NeuroCore database is an extension of the core database schema that has links to repositories of neuroanatomical and neurochemical concepts and provides an extendible specification of items needed in most experimental records, such as research subject, experimental manipulation, structure of the research data, and the statistics performed on the data. There is a slot for research data and a standard extension for handling time series data. NeuroCore comes with a Java applet called the Schema Browser, which allows one to learn the structure of a particular laboratory's database by showing, for each familiar core table, the extensions particular to that laboratory.

Mediator Systems and Common Ontologies

In a cooperative federated database system, the challenge is to provide conceptual links between data sources to begin to address the problem of data integration: assisting users by selecting, restructuring, and merging information from different databases and providing an integrated view of the information. Each database in the federation has its own *ontology* (the collection of concepts and their relationships used to describe information units in the database). But these ontologies may use terms differently (a "cell" in one may be a "neuron" in another and a "neuron" or a "glia cell" in yet another). If we seek a direct translation between each pair of ontologies, then n ontologies require $n(n - 1)$ translators. However, if we can define a common ontology and translate back and forth between it and the other ontologies, then only $2n$ translators are

required. With a common ontology that serves the basis of mutual understanding among participants in the federation, information is shared via what are called *mediators*. These support import (folding remote information into local environments), export (registering information to share), discovery (searching for relevant information), and browsing (navigating through information sources).

Kahng and McLeod (Chapter 5.1 of *CtB*) show one way to dynamically build a common ontology. Their motivation is that it is extremely difficult to reach total agreement on an ontology if there are many participants and that the ontology should be allowed to change dynamically as the federation evolves.

The model-based mediation architecture offered by Ludaescher, Gupta, and Martone (2001) employs F-logic as a data and knowledge representation and reasoning formalism. Integrated database views are defined and executed at the level of conceptual models rather than at the usual level of database schemas or XML. To semantically correlate across databases with little or no overlap in their schemas, they introduce domain maps, which are semantic nets of terms and relationships used to mediate across sources; for example, Purkinje cell “is a” neuron; cerebellum “has a” cerebellar cortex. The mediator then uses these rules to navigate through the various levels of brain analysis to allow for complex queries that span multiple levels of resolution and multiple data sources. Sources of domain knowledge include existing ontologies such as those provided by brain atlases and the Unified Medical Language System (UMLS) project of the National Library of Medicine (<http://www.nlm.nih.gov/research/umls/>), but the aim is to provide tools to allow different groups of users to tailor their own extensions to the ontology.

Modeling and Simulation

Consider modeling the role of neural circuitry of certain brain regions in a given set of tasks. For each brain region, a survey of the neurophysiological data calls attention to a few basic cell types with firing characteristics that are strongly correlated with some aspect of that task. For example, in eye movement tasks, some cells fire most strongly near the onset of the target stimulus, others seem to be active during a delay period, and others are more active near movement initiation. The data tell the modeler what the activity of the cells should be in a variety of situations, but in many cases, experimenters do not know in any quantitative detail the way in which the cell responds to its synaptic inputs, nor do they know the action of the synapses in great detail. In short, the available empirical data might not be rich enough to define a *causally complete* model. Therefore, to get the model to run, the modeler has to make a number of hypotheses about some of the unknown connections, weights, time constants, and so on. The modeler may even have to postulate cell types that experimenters have not yet looked for and show by computer simulation that the resulting network will indeed perform in the observed way when known experiments are simulated, in which case it must match both external behavior and the key data on model populations that were based on cell populations with measured physiological responses. What raises the ante is that (1) the modeler’s hypotheses suggest new experiments on neural dynamics and connectivity and (2) the model can be used to simulate experiments that have never been conducted with real nervous systems.

A few years from now, new models will both examine the interactions of a larger number of brain regions and analyze cells within each region in increasing detail. There is no way we would be able to keep cognitive track of these models if we had to look at everything at once. One approach (e.g., Arbib, Chapter 2.1, in *CtB*) is to represent complex models in an object-oriented way, using a hierarchy of interconnected modules. A module might be an interconnected set of brain regions; each region in turn might itself be a module composed of yet smaller modules that represent

arrays of neurons sharing some common anatomical or physiological property. In any case, a module is either decomposable, in which case this “parent module” is decomposed into submodules known as its child modules, or the module is a “leaf module” that is not decomposed further but is directly implemented in the chosen programming language. There are basically two ways to proceed for a complex model. We can get a hierarchical view of the overall model, or we can zoom in on subsystems and study them in detail.

Multilevel Simulation

As recently as 2000, a detailed model of chemicals reacting and diffusing around a single synapse could require a full day of workstation processing to simulate just a few seconds of synaptic activity. There may be of the order of 10,000 synapses on a “typical” neuron, millions of neurons in a single region, and hundreds of regions in a brain. Clearly, any simulation methodology that required one to simulate every synapse in such detail would be doomed to failure. No short-term increase in computer power will allow us to reduce the simulation of a second’s activity in a system with 10^{15} synapses from 10^{15} hours or minutes down to a single second. A major challenge for work in multilevel simulation is thus to undertake detailed simulation at one level to validate a (possibly context-dependent) approximation that can be used in far more efficient large-scale simulations at the next level.

NEUROSIMULATION: TOOLS AND RESOURCES, NSL NEURAL SIMULATION LANGUAGE, GENESIS SIMULATION SYSTEM, and NEURON SIMULATION ENVIRONMENT present tools bridging from circuits in interacting brain regions down to compartmental models of individual neurons. For example, a NSL model might employ a neuron module that is far simpler than a corresponding compartmental model developed in NEURON but that has been validated by careful studies to yield an economical but effective approximation to it. Or a GENESIS modeler might want to check that a model of a compartment provides a satisfactory approximation to a far more detailed model of neurochemical details for the neuron. All this raises two important challenges for the neural simulation community. One is to increase the range of tools currently available for comparing model to model as well as model to data (with the parameter search methods that this implies). The other is to develop “wrapping” technology so that modules developed by using one simulator can indeed be used to replace objects (whether to simplify them or attend to crucial new details) in an existing model developed by using another simulator.

Young, Hilgetag, and Scannell (2000) offer one example of modeling at the highest level, far removed from the details of neural circuits (see also COVARIANCE STRUCTURAL EQUATION MODELING). They developed a formal framework for inferring function from structure in which knowledge of connectivity is necessary but not sufficient. They applied this framework to inferences about a simple network that reproduces intact, lesioned, and paradoxically restored orienting behavior. Lesion effects could be used to infer which structures contributed to particular functions in this simple network. Clearly, such an approach can complement, but not replace, attempts to link high-level data from lesion effects or brain imaging to the analysis of detailed neural circuitry. SYNTHETIC FUNCTIONAL BRAIN MAPPING reviews methods for calibrating human brain imaging against simulations of monkey neurophysiology; while Kötter et al. (2002) simulate activity propagation in the primate visual cortex with the aim of relating neuronal activity to cortical activation patterns and relating onset response latencies to the structure of the underlying anatomical network.

Usui (2002) describes the NRV (Neuroinformatics Research in Vision) project, which aims to construct mathematical models for each level of the visual system (single neuron, retinal neural circuit, visual function), build resources for neuroinformatics, and develop a new vision device based on brain-type information-processing

principles. A mathematical model of a retinal neural circuit will be constructed from neurophysiological experimental data and from the characteristics of single neurons. This circuit will form the basis for a “virtual retina” that encompasses everything from the light energy conversion mechanism in a photoreceptor to the encoding mechanism of impulse sequence in a ganglion cell, which is the retinal output. The results of this project will be made available in a database, the VISIOME Platform, that integrates morphological and physiological knowledge and mathematical models with related studies and references.

Arbib (1995) formulated the challenge of building databases that link models developed with different simulators to each other and to empirical databases, and Bischoff-Grethe, Spoelstra, and Arbib (Chapter 6.2 in *CtB*) explore the integration of one such model repository, Brain Models on the Web, with a summary database. Goddard et al. (2001) further develop the theme that software tools are needed that support discussion, development, and exchange of computational models. They describe methodologies that focus on these tasks and discuss the use of templates, declarative forms of model description equivalent to object-oriented classes and database schemas, to describe models ranging from neuron cell membranes to neural networks. The paper introduces NeuroML, a markup language for neuroscience that is defined syntactically using templates, with a component designed to support communication between modeling-related tools.

Neuroanatomy

One way to integrate data from diverse experiments on the brains of a given species is to register the data—whether the locations of cells recorded neurophysiologically, the tract tracings of an anatomical experiment, or the receptor densities revealed on a slice of brain in a neurochemical study—against a standard brain atlas. Just as people have different faces, so do rats and other animals have different brains, and therefore there is a registration problem: Given a location in an individual brain, what is the best bet as to the corresponding location in the “standard” brain?

Part 4 of Arbib and Grethe (2001) offers a number of approaches to atlas-based databases. The core of the work is NeuART, a neuroanatomical viewer for the rat brain based on the Swanson atlas (Swanson, 1998). It also presents NeuroSlicer, a tool for registering 2D slice data against a 3D model of the rat brain reconstructed from the Swanson atlas, as well as the design of an atlas-based database of neurochemical data. The Swanson atlas contains 73 plates representing cross sections of one-half of the rat brain. These are not uniformly spaced but rather were chosen to exhibit many crucial features of the rat’s neuroanatomy. Each plate contains a photomicrograph of a stained brain section on the left and Swanson’s representation of that section on the right, in which he draws boundaries separating different brain regions and labels the regions. Many of the curves dividing one nucleus from another correspond obviously to boundaries in the cell densities visible on the micrograph. Others cannot be seen from that particular micrograph and can be revealed only by a variety of staining techniques or by the incorporation of physiological and other data. It therefore requires great skill on the part of the anatomist to draw those nonobvious divisions, and in fact even expert neuroanatomists may disagree. Therefore, although there is much agreement between the Swanson atlas and the other leading atlas of the rat brain, the Paxinos-Watson atlas (Paxinos and Watson, 1998), there are also disagreements. Thus we have the future challenge not only of registering data against a particular choice of atlas, but also of facing the issue of how to update such data sets as future anatomical research resolves certain disagreements and leads to more reliable demarcation of boundaries.

NeuART allows one to view any template of the Swanson atlas through a Web browser, together with any data overlays retrieved

from the database. A Display Manager allows one to see these different results, and a Viewer Manager allows one to customize the Display Manager to one’s needs. The Query Manager provides forms that make it easy to request anatomical information from the Informix database; the results of these queries are described textually by a Results Manager, and the user can maintain a set of results of interest. The Level Manager allows one to choose which level (template) of the brain to examine, and the Active Set Manager then shows which results of the query have data that are relevant for that set. These can then be displayed by clicking on the appropriate elements.

Another approach to preparing neuroanatomical data is to flatten computer images of monkey or human cerebral cortex. As we know from atlases of the world, such flattening requires cuts if it is to preserve areas of the surface and then can preserve only local, but not global, spatial relationships. Nonetheless, when data about cortex rather than deep brain structures is paramount, display of data on such a flattened map of cortex allows one to take in patterns at a glance in a way that is impossible if one must scroll slice by slice through the pages of a conventional brain atlas. Van Essen et al. (2001) describe three software programs for carrying out surface-based analyses of cerebral cortex: SureFit (Surface Reconstruction by Filtering and Intensity Transformations) is used primarily for cortical segmentation, volume visualization, and initial surface generation; Caret (Computerized Anatomical Reconstruction and Editing Toolkit) provides a range of surface visualization and analysis options plus capabilities for surface flattening, surface-based deformation, and other surface manipulations; and SuMS (Surface Management System) provides a version control system that is capable of handling large numbers of surface and volume data sets. With built-in database management system support, SuMS provides rapid search and retrieval capabilities across all the data sets while also incorporating multiple security levels to regulate access.

Of course, the problem of linking data to neuroanatomy is not limited to vertebrates, let alone mammals. Jacobs and Theunissen (2000) examine the anatomical basis for the representation of stimulus parameters within a neural map and examine the extraction of these parameters by sensory interneurons in the cricket cercal sensory system. Their modeling of the cricket cercal system makes crucial use of their identified neuron database (<http://cns.montana.edu/research/neurosys/>).

We close this section with the work of Toga and Thompson (2002) on the collection of images of normal and diseased human brains brain in vivo and post vivo. They stress that the design of appropriate reference systems for human brain data presents considerable challenges, since these systems must capture how brain structure and function vary in large populations, across age and gender, in different disease states, and across imaging modalities, not to mention comparison across species. This work requires new approaches in computer vision, partial differential equations, and statistical field theory to detect and visualize disease-specific patterns. They survey the types of maps relevant to mental disorders, including maps that capture dynamic patterns of brain change in dementia.

The NeuroHomology Database

The term “homology” is a central one in comparative biology, referring to characteristics of different species that are inherited from a common ancestor. Defining homologies between brain structures requires a process of inference from distinct clusters of attributes. Bota and Arbib (Chapter 6.4 of *CtB*) introduce the concept of *degree of homology*. To define a neural structure, neuroscientists use numerous attributes, including gross morphology, relative location, cytoarchitecture, types of cell responses to different ways of stimulation, and function. In similar fashion, Bota and Arbib employ eight criteria for determining the degree of homology of two brain

structures: the morphology of cells within each brain structure and the relative position, cytoarchitecture, chemoarchitecture (neurotransmitters that are found within a brain structure), myeloarchitecture, afferent and efferent connections, and function of each of a pair of brain structures from two species. If two brain structures have common cell types, chemoarchitectonics, and cytoarchitectonics and common connectivity patterns, then one should expect that those two brain structures have the same function or related functions. This is the case for the primary visual area (area 17). In each major mammalian species, area 17 can be delimited on the basis of myeloarchitecture (heavy myelination) and cytoarchitecture (the presence of a granular layer IV), the presence of a single and systematic visuotopic map, a well-defined pattern of subcortical afferents, small receptive fields, and the presence of many orientation-selective neurons with simple receptive fields. Bota and Arbib not only discuss the homology criteria that can be established between pairs of brain structures across species, but also introduce the NeuroHomology (NHDB) summary database. This database contains three interconnected entities: Brain Structures, Connections, and Homologies. A user who wants to find whether there is any homology between structures X and Y from two different species can also find the definitions of structures X and Y according to different sources, as well as the afferents and efferents of these two structures. More important, the latest version of NHDB has three inference engines: one for combining data of differing reliability on the connections between two brain regions, one for comparing neuroanatomical data for a given species when the data come from the different parcellations provided by different brain atlases, and one for estimating the degree of homology according to multiple criteria.

Discussion

The full development of neuroinformatics will provide an environment that helps the user pass back and forth between empirical data and related models, even though these are distributed across a federation of databases. Mediation technology will help to integrate these databases despite their diversity of ontologies and database schemas. In particular, a variety of brain atlases will provide reference platforms for a host of data within a species, with the analysis of homologies supporting the linkage of data across species. Future standards activity will provide modelers using a variety of simulation environments with tools to develop interfaces that make it easy for nonprogrammers to run basic “experiments” with the models and add to the database comments on the comparison of simulation results with available empirical data, to install models, to create versions of both models and parameter sets, and to freeze models in various interesting states for later analysis under varying conditions. A crucial aspect in all this is to catalyze a truly cumulative style of modeling in neuroscience by facilitating the reus-

ability of modules within current neural models, with the pattern of reuse fully documented and tightly constrained by the linkage with a federation of databases of empirical neuroscientific data.

Road Map: Implementation and Analysis

Related Reading: Databases for Neuroscience; Neurosimulation: Tools and Resources

References

- Arbib, M. A., 1995, Brain models on the Web, in *Computational Intelligence, A Dynamic Systems Perspective* (M. Palaniswami, Y. Attikouzel, R. J. Marks II, D. Fogel, and T. Fukuda, Eds.), New York: IEEE Press, pp. 219–231.
- Arbib, M. A., and Grethe, J. S. (Eds.), 2001, *Computing the Brain: A Guide to Neuroinformatics*, San Diego: Academic Press. ♦
- Goddard, N. H., Hucka, M., Howell, F., Cornelis, H., Shankar, K., and Beeman, D., 2001, Towards NeuroML: Model description methods for collaborative modeling in neuroscience, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 356(1412):1209–1228.
- Heimbigner, D., and McLeod, D., 1985, A federated architecture for information management, *ACM Transactions on Office Information Systems*, 3(3):253–278.
- Jacobs, G. A., and Theunissen, F. E., 2000, Extraction of sensory parameters from a neural map by primary sensory interneurons, *J. Neurosci.*, 20(8):2934–2943.
- Kötter, R., 2001, Neuroscience databases: Tools for exploring brain structure-function relationships, *Philos. Trans. R. Soc. Lond. B.*, 356:1111–1120. ♦
- Kötter, R., Nielsen, P. D., Johnsen, J., Sommer, F. T., and Northoff, G., 2002, Multi-level neuron and network modeling in computational neuroanatomy, in *Computational Neuroanatomy: Principles and Methods* (Ascoli, G., Ed.), Totowa, NJ: Humana, pp. 359–382.
- Ludaescher, B., Gupta, A., and Martone, M. A., 2001, Model-based mediation with domain maps, *17th International Conference on Data Engineering (ICDE)*, IEEE Computer Society, Heidelberg, Germany, pp. 81–90.
- Paxinos, G., and Watson, C., 1998, *The Rat Brain in Stereotaxic Coordinates*, 2nd ed., San Diego, CA: Academic Press.
- Shepherd, G. M., Mirsky, J. S., Healy, M. D., Singer, M. S., Skoufos, E., Hines, M. S., Nadkarni, P. M., and Miller, P. L., 1998, The Human Brain Project: Neuroinformatics tools for integrating, searching, and modeling multidisciplinary neuroscience data, *Trends Neurosci.*, 21:460–468.
- Swanson, L. W., 1998, *Brain Maps: Structure of the Rat Brain*, Amsterdam: Elsevier Science Publishers.
- Toga, A. W., and Thompson, P. M., 2002, New approaches in brain morphometry, *Am. J. Geriatr. Psychiatry*, 10(1):13–23. ♦
- Usui, S., 2002, The NRV project (Neuroinformatics Research in Vision), <http://www.neuroinformatics.gr.jp/>.
- Van Essen, D. C., Dickson, J., Harwell, J., Hanlon, D., Anderson, C. H., and Drury, H. A., 2001, An integrated software system for surface-based analyses of cerebral cortex, *JAMA (Special issue on the Human Brain Project)*, 41:1359–1378.
- Young, M. P., Hilgetag, C. C., and Scannell, J. W., 2000, On imputing function to structure from the behavioural effects of brain lesions, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 355(1393):147–161.

Neurolinguistics

Barry Gordon

Introduction

Neurolinguistics began as the study of the language deficits occurring after brain injuries but now encompasses all aspects of language and the brain, normal as well as disturbed. To help place neural modeling efforts in perspective, this chapter is intended as

an overview of current understanding of language and its putative neural bases.

Recent reviews of neurolinguistics and its methods can be found in Berndt (2001) and Brown and Hagoort (1999). Speech perception is reviewed in SPEECH PROCESSING: PSYCHOLINGUISTICS; speech production in LANGUAGE PROCESSING, SPEECH PRODUC-

TION and Levelt (2001). Gordon (1997) provides an example of a detailed, multilevel analysis of one language function, the visual naming task.

First-Order Model of Speech/Language and Neuroanatomy

There is now general agreement concerning what constitutes actual language and what does not, what major functional abilities underlie speech and language, how these are interconnected, and their approximate neuroanatomic dependencies. Collectively, this first-order model can summarize many features of normal ability as well as developmental and acquired disorders. A century and a half of intensive study has also identified many errors that might occur in the experimental and theoretical analysis of behavior-brain relationships. Understanding these pitfalls is useful for interpreting current evidence and for modeling attempts in the future.

Language and Language Development

Babbling, grunts, shouts, and other emotional expressions are not language; they occur in non-human primates and are not affected by the same brain lesions that may abolish almost all language abilities. Even curses, idiomatic expressions, and singing may also be spared by such lesions. The language and related speech capabilities that are the focus of this chapter are propositional (i.e., both symbolic or referential, and created from combinations of elements). Although language can express concepts and reasoning, concept formation and reasoning do not necessarily depend on language.

All normal individuals have the capability to rapidly acquire language (LANGUAGE ACQUISITION). Some of this capability is expressed neonatally and even in utero. An extensive search for genes that might be responsible for language capabilities has produced several candidates and much controversy.

Until recently, it was thought that this facile ability to acquire language might atrophy in most people after approximately five years of age. New evidence, both empirical and theoretical, suggests that at least some of the apparent loss of plasticity could be the result of entrapment by the first-learned language. With special

training or stimuli, some phonological limitations may be overcome.

Functional Architecture

It is generally agreed that the complex overt functions of language and speech are the byproduct of various combinations of internal subprocesses (stages). Different overt functions may use these stages in different arrangements, and with different demands. Figure 1 is a schematic of these stages and their inputs, outputs, and interconnections. Every normal auditory/oral language user has functional modules for auditory phonemic perception, auditory word recognition (phonologic word-form recognition), lexical-semantic, abstract word form retrieval (which may or may not involve syntactic information, depending on the account), phonological word-form retrieval, and production of articulatory patterns. The process of learning to read and to write also establishes language-specific capabilities within the visual and motor systems. In the case of an alphabetic language such as English, this includes stages that perform abstract visual letter- and word-form recognition. These stages are connected to phonemic, phonologic word-form, and lexical-semantic processes. Output (in English) maps from phonologic and subphonologic word-forms to the written forms. Language communicated by gesture (such as American Sign Language) requires its own specialized visual recognition and manual production mechanisms, which also communicate with core language faculties (Crone et al., 2001). Figure 2 shows the approximate neuroanatomic associations of some of these stages in normal right-handed individuals.

In normal right-handed individuals, core functions required for speech and language are almost always (>99%) the responsibility of the left (dominant) hemisphere. In left-handed individuals (who make up ~13% of the population), or even those who are right-handed but with a family history of left-handedness, the core functions can be in the right hemisphere, or more bilaterally distributed. The lateralization of speech and language is under genetic control.

Lesion Effects and Classic Syndromes

A large part of the evidence for this lateralization, parcellation, and mapping of language functions has come from individuals with

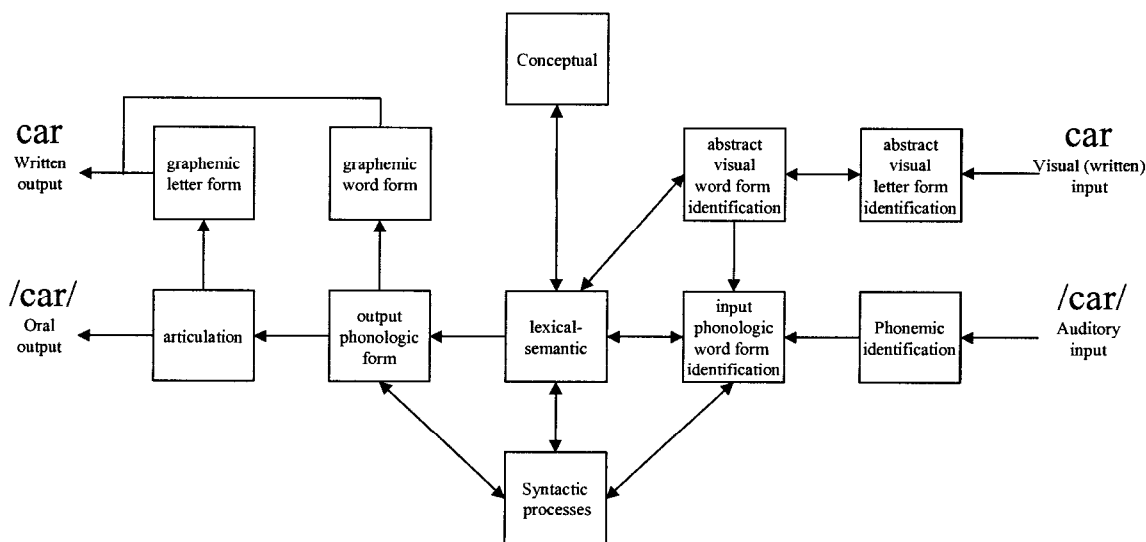
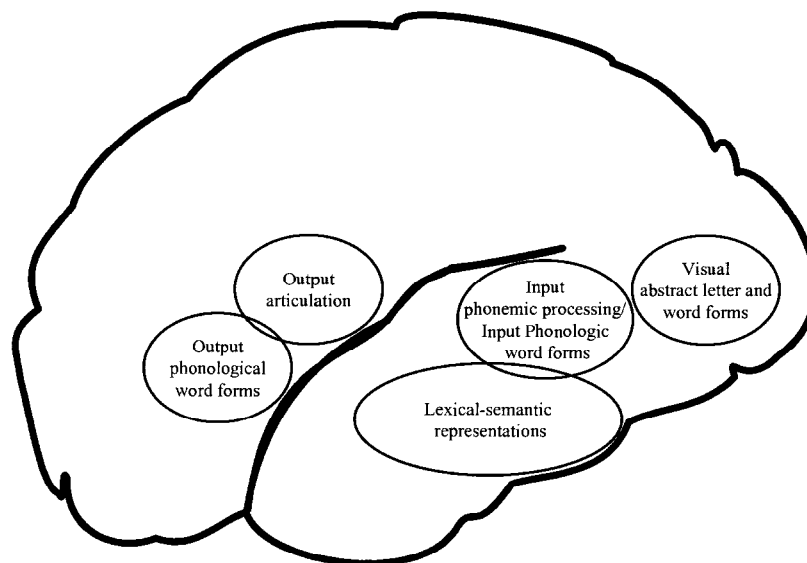


Figure 1. Schematic of larger-scale functional modules and interconnections involved in speech/language and related functions. Visual confront-

tation naming uses visual inputs analogous to those used for reading; see Figure 4. (Copyright IA, Inc. Used with permission.)

Figure 2. Neuroanatomic associations in the left (dominant) hemisphere of some of the functional modules shown in Figure 1. Note that these are approximate, and that there is also considerable individual variation; see text. (Copyright IA, Inc. Used with permission.)



acquired brain injuries. The common forms of brain injury that occur in adults tend to produce somewhat distinctive patterns of impairment in language and speech (collectively termed “aphasia”), particularly in the late (>6 month) period after the injury. *Broca’s aphasia* is typified by slow, effortful, halting speech, frequently described as telegraphic and agrammatic because function words and grammatical endings are often omitted. These individuals may appear to have good comprehension of speech, but it is now appreciated that their comprehension of syntax is also impaired. Lesions associated with Broca’s aphasia are typically in the anterior regions of the left hemisphere, including the posterior inferior frontal lobe (Broca’s area), but are more extensive than this area. The pattern of deficits in Broca’s aphasia contrasts with that of *Wernicke’s aphasia*, in which speech is fluent and well articulated but often empty of specific meaning (as in “Jabberwocky”). Content words (nouns and verbs) are often misused or not recognizable (neologisms). Grammatical function words and endings are present but used incorrectly (with errors in inflections, and with nongrammatical (paragrammatic) constructions). These deficits are found in both production and comprehension as well as in reading. Individuals with Wernicke’s aphasia are usually found to have lesions that include the posterior superior temporal lobe and adjacent parietal lobe (Wernicke’s area) of the dominant hemisphere. In addition to these syndromes, a number of other relatively distinctive ones have been described, with varying degrees of anatomic lesion specificity.

Aphasic deficits reveal themselves as either complete failures to perform a function or as delays and/or errors in performance. Among the errors commonly observed are those that sound like the intended target (such as “bot” for “dot”) (*phonemic paraphasias*) and those with a meaning similar to the intended target (as in “door” for “exit”) (*lexical or semantic paraphasias*). Normal individuals also make most, if not all, of the errors made by aphasic individuals, as Freud and others pointed out, but aphasic individuals make them with a much higher frequency.

The Expanded First-Order Model

Aphasic syndromes and the errors that normal and aphasic individuals may make have not changed in the past hundred years. As a

result, the basic block diagrams of the models explaining them have not changed much (compare Figure 3), but critical details are now different. Methodological errors have been identified, and additional methods and subject groups have been used. As a result, finer details have been added. In addition, speech and language are now recognized as the products of interactive dynamic systems, with major implications for modeling normal and abnormal performance and for understanding their neural substrates.

Methodological Refinements

Broca’s aphasia, Wernicke’s aphasia, and the other aphasic syndromes are now understood to be collections of more than one fundamental deficit. For example, Broca’s aphasia is a variable mix of difficulties in comprehension of syntax (IMAGING THE GRAMMATICAL BRAIN) and deficits in word retrieval, production of syntax, and articulation. These independent functions happen to co-occur in Broca’s aphasia because the typical brain injury that causes the syndrome, vascular infarction, affects all the different regions responsible at once. Similarly, Wernicke’s aphasia has now been reinterpreted as a variable mixture of deficits in phonemic speech

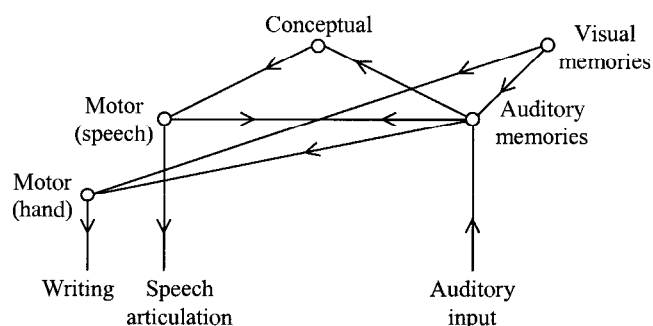


Figure 3. An example of a classical model of speech/language functions: Lichtheim’s 1885 model, redrawn and labeled for brevity. Note that the modern meanings of the labels may not correspond exactly to the meanings Lichtheim originally intended. (Copyright IA, Inc. Used with permission.)

tant for speech and language processing (but see Martin et al., 1999).

Syntactic functions are clearly important for both comprehension and production. Some consider syntactic abilities to be the product of a relatively small set of functions (IMAGING THE GRAMMATICAL BRAIN), whereas others have suggested that syntactic abilities are the product of a much larger and more variegated collection of processes (Dick et al., 2001).

There is no consensus on the nature of conceptual and semantic abilities. Much evidence favors the existence of modality-specific semantic capabilities, connecting to an amodal repository of additional semantic knowledge. However, both extremes (only modality-specific semantic processes, or only an amodal semantic system) have also been argued.

It is also now understood that, in any one individual, the areas devoted to specific language functions are likely to be smaller in extent, and more variable in location, than the maps derived from chronic studies of large numbers of individuals (Gordon et al., 2001). As a result, when it is necessary to have precise localization in any one individual (as in patients requiring focal cerebral excisions for treatment of seizures or tumors), detailed, individualized mapping is necessary (Gordon et al., 2001).

Dynamic Processing Considerations

Perhaps the most important addition to the modern understanding of how language is represented and processed in the brain is an understanding of its dynamic properties. That language is the product of complex, interacting, dynamic systems may explain the richness of language abilities and hitherto intractable problems such as how word and sentence information might be integrated online. Furthermore, what is known or should be known about these dynamics will need to be explained by any theory of the neural processes responsible for them. What follows is a synthesis of the inferences and assumptions that have been made concerning the dynamics of processing of the cognitive information relevant to speech/language and related functions.

Between-Stage Information Flow

The approximate sequence of the flow of information between stages is shown in Figure 1. Auditory speech perception somewhat precedes lexical semantic comprehension; visual perception of a picture somewhat precedes its comprehension and the generation of its name. However, the times involved in each stage are relatively long compared to the time required for the overall process, so there is considerable overlap in processing between stages (cascaded processing).

In addition, there is almost certain to be extensive feedback between stages (only shown for the most likely ones in Figure 1), both at adjacent levels and at greater psychological (and neural) distances. Many potential neuroanatomic mechanisms for such feedback are known to exist, and some have been shown to be operational (with the most compelling examples coming from the visual domain).

Dynamics of Within-Stage Processing

There is fair agreement that processing within a stage is not a step function; it has a rise and fall that can be measured by behavioral as well as by more interventional techniques (Hart et al., 1998). Although the data on durations of processing are difficult to interpret, it certainly appears that speech perceptual processes can be accomplished in as little as 20–50 ms (or less), whereas lexical-semantic selection involved in a function such as naming may take 200–400 ms.

Unitary Representations and Activation

It has been widely assumed that the contents of any particular stage are some form of unitized, independent representations. The terminology of these unitized representations is confusing (Gordon, 1997). In cognitive science, they are often termed “nodes.” However, these are not the nodes of connectionist networks. Nor do these unitized representations almost ever correspond to the activity of single neurons, either in actuality or in theory (although they could, if there really were “grandmother cells”). Here, we will use the standard cognitive science terminology of “nodes.”

What information these unitized representations convey has not been settled. They may correspond to features that have already been psychologically identified (e.g., the /ba/ part of the phonological representation of /bat/, or the {round} semantic feature of a lexical-semantic representation of {ball}). Others have assumed that what they represent is much less transparent.

Nodes are generally assumed to have a number of characteristics and properties (although not every theorist who uses nodes necessarily endorses or uses all of the ones listed here; see Gordon, 1997):

Except for perhaps very basic perceptual and motor features, nodes are not preexisting; they must be created by experience.

Once created, nodes persist in latent form, available for activation. The latent strength of a node is greater, depending on its experiential frequency. Experiential frequency is probably a surrogate for memory; the strength of a node is a form of memory, increased by frequency of exposure and by all the other mechanisms that consolidate learning (such as salience).

Nodes can be converted from a latent to an active (activated) state. Nodes are activated by virtue of their connections to other nodes (“spreading activation”).

The degree to which a node is activated is determined by several factors:

By how well the input(s) match the node’s receptive pattern. Inputs can be inhibitory as well as facilitatory.

By the node’s latent strength

By a random component (noise). Noise may be absolute, or it may be correlated with the degree of activation, or both.

Activation of a node grows over time to the maximum determined by the factors listed previously.

After a node has reached maximal activation, its activation may persist for a time or may decay.

It is likely that there is some degree of control over nodal properties, both internal and external to each stage: how easily they are formed, how they activate and decay, and how they can be altered by experience.

There is little direct evidence for nodes, but there are several lines of indirect support. Postulating nodes with properties such as those listed previously has proven to be very useful for understanding many otherwise puzzling phenomena of normal and brain-injured performance (Gordon, 1997). Some brain injuries can be interpreted as though they selectively destroy some nodal representations within a stage (see Gordon, 1997). In other cases, focal brain injuries have been successfully modeled in terms of changes in nodal properties or connection strengths, as in the Dell et al. (1997) and Foygel and Dell (2000) simulations discussed later in this article.

Processing by Constraint Satisfaction

The input information available to a stage is very likely to under-specify the correct output, particularly at the beginning of pro-

cessing. Evidence exists from several different levels of language processing (speech perception, reading, and sentence-level comprehension, among others) that all possible candidate representations are activated initially. With continued input, and with input from other stages, the candidate set is winnowed down to the correct one. The specifics of this process are debated, but it is likely that such dynamic, interactive computation plays a critical role in many known examples of language processing. Dynamic processing by interactive constraint satisfaction may also explain how multiple sources of information can interact in the course of comprehension and production. See, for example, Arbib and Caplan (1979), *LESIONED NETWORKS AS MODELS OF NEUROPSYCHOLOGICAL DEFICITS*, and *OPTIMALITY THEORY IN LINGUISTICS*.

Possible Neural Bases

As depicted in the schematic of Figure 2, some stages or collections of stages seem to be the product of relatively discrete areas of the cerebral cortex. Connections between stages seem to correspond to short- and long-range subcortical white matter pathways.

How the cognitive elements (nodes) of psychological theorizing correspond to actual neuronal activity is not known for certain. However, the attractor states that can occur in neuronal networks with feedback are viable candidates for behaving as nodes are posited to behave (*LESIONED NETWORKS AS MODELS OF NEUROPSYCHOLOGICAL DEFICITS*). A single neural network can have multiple attractor states. The sculpting of connections to create an attractor can be equated with the learning of a node. The width of a basin of attraction of a neural activity state may be the neural counterpart of a node's similarity relations (its receptive field). The depth of a basin of attraction, as well as its width, may correspond to frequency ("strength"). The process of "activation" of a cognitive node may correspond to the evolution of neural activity toward the attractor. Attractor dynamics have proven to be very useful in modeling a number of language and related functions (see, for example, McLeod, Shallice, and Plaut, 2000; also *COMPUTING WITH ATTRACTORS; NEUROLOGICAL AND PSYCHIATRIC DISORDERS*). However, attractors are not the only emergent features of neural network activity that might be candidates for nodes (see, for example, *STRUCTURED CONNECTIONIST MODELS*).

Models of Speech/Language Functions

Many modeling efforts in neurolinguistics have been concerned with the consequences of relatively large-scale assumptions about stages and connections. Examples of models that have incorporated both stage- and substage-level assumptions are those used by Dell et al. (1997) and by Foygel and Dell (2000). Dell et al. (1997) simulated picture naming using semantic, lexical, and phonologic stages. Each level represented its information as nodes, with properties similar to those discussed earlier. Spreading activation drove production. Parameters that replicated normal performance were derived from control data. Then, to model the picture naming performance of aphasic subjects, two parameters were altered for each aphasic subject: one governed how much activation spread, and one determined how rapidly activation declined. There was a "fairly good" fit between simulated and actual performance with just these assumptions (as characterized by Foygel and Dell, 2000). In addition, Dell et al. (1997) were able to capture much of the pattern of recovery of patients in terms of changes in these same parameters back to their normal values. Even so, Foygel and Dell (2000) have suggested that the data used by Dell et al. (1997), and other cases in the literature, might be better explained by alterations in two other parameters: the connection strengths (weights) be-

tween semantic and lexical units, and those between lexical and phonological units.

Modeling efforts by Dell and his colleagues represent one motivation for this chapter's method of review. Modeling efforts are likely to be most productive when they are informed by a good knowledge of existing data and theoretical explanations and also by a realistic understanding of when those data and explanations may be incomplete or arguable. *LESIONED NETWORKS AS MODELS OF NEUROPSYCHOLOGICAL DEFICITS* provides additional perspectives on modeling attempts.

Discussion

The emergent complexity of language functions is a daunting challenge for both its experimental investigation and for modeling attempts. However, it is also encouraging, because it makes it more likely that the same fundamental building blocks and processes that are identified in other domains of brain function will be applicable to understanding language, and vice versa.

Road Maps: Cognitive Neuroscience; Linguistics and Speech Processing

Related Reading: Imaging the Grammatical Brain; Language Evolution: The Mirror System Hypothesis; Language Processing; Lesioned Networks as Models of Neuropsychological Deficits

References

- Arbib, M. A., and Caplan, D., 1979, Neurolinguistics must be computational, *Behav. Brain Sci.*, 2:449–483.
- Berndt, R. S., 2001, *Language and Aphasia* (2nd ed.), Amsterdam, Netherlands: Elsevier. ♦
- Brown, C. M., and Hagoort, P., 1999, *The Neurocognition of Language*, Oxford, UK: Oxford University Press. ♦
- Crone, N. E., Hao, L., Hart, J., Jr., Boatman, D., Lesser, R. P., Irizarry, R., and Gordon, B., 2001, Electrocorticographic gamma activity during word production in spoken and sign language, *Neurology*, 57(11):2045–2053.
- Dell, G. S., Schwartz, M. F., Martin, N., Saffran, E. M., and Gagnon, D. A., 1997, Lexical access in aphasic and nonaphasic speakers, *Psychol. Rev.*, 104(4):801–838.
- Dick, F., Bates, E., Wulfeck, B., Utman, J. A., Dronkers, N., and Gernsbacher, M. A., 2001, Language deficits, localization, and grammar: Evidence for a distributive model of language breakdown in aphasic patients and neurologically intact individuals, *Psychol. Rev.*, 108(4):759–788.
- Foygel, D., and Dell, G. S., 2000, Models of impaired lexical access in speech production, *J. Mem. Lang.*, 43:182–216. ♦
- Gordon, B., 1997, Models of naming, in *Anomia: Neuroanatomical and Cognitive Correlates* (H. Goodglass and A. Wingfield, Eds.), San Diego, CA: Academic Press, pp. 31–64. ♦
- Gordon, B., Boatman, D., Hart, J., Jr., Miglioretti, D., and Lesser, R. P., 2001, Direct cortical electrical interference (stimulation), in *Language and Aphasia*, 2nd ed., Vol. 3 (R. S. Berndt, Ed.), Amsterdam: Elsevier Science B.V., pp. 375–391.
- Hart, J., Jr., Crone, N. E., Lesser, R. P., Sieracki, J., Miglioretti, D. L., Hall, C., Sherman, D., and Gordon, B., 1998, Temporal dynamics of verbal object comprehension, *Proc. Natl. Acad. Sci. USA*, 95(11):6498–6503.
- Hickok, G., and Poeppel, D., 2000, Towards a functional neuroanatomy of speech perception, *Trends Cogn. Sci.*, 4(4):131–138.
- Hillis, A. E., Kane, A., Tuffiash, E., Ulatowski, J. A., Barker, P., Beauchamp, N. J., and Wityk, R. J., 2001, Reperfusion of specific brain regions by raising blood pressure restores selective language functions in subacute stroke, *Brain Lang.*, 79(3):495–510.
- Levelt, W. J., 2001, Inaugural Article: Spoken word production: A theory of lexical access, *Proc. Natl. Acad. Sci. USA*, 98(23):13464–13471. ♦
- Martin, R. C., Lesch, M. F., and Bartha, M. C., 1999, Independence of input and output phonology in word processing and short-term memory, *J. Mem. Lang.*, 41(1):3–29.
- McLeod, P., Shallice, T., and Plaut, D. C., 2000, Attractor dynamics in word recognition: Converging evidence from errors by normal subjects, dyslexic patients and a connectionist model, *Cognition*, 74(1):91–114.

Neurological and Psychiatric Disorders

Eytan Ruppín and James A. Reggia

Introduction

In the last decade it has become natural to ask how neural modeling may be harnessed to investigate the pathogenesis and potential treatment of brain disorders, and in what ways it may complement more traditional research methodologies. Indeed, early attempts in this direction have been extensively developed in recent years.

The interest of the psychiatric and neurological communities in neural network modeling probably reflects the belief that, although the gathering of neurobiological data has led to much progress in our understanding of basic brain mechanisms, we do not appear to have come much closer to understanding how these mechanisms result in behavior. Neural modeling is a methodology that is precisely aimed at bridging this gap, by studying the relation between the “microscopic” pathological alterations of the underlying neural networks and the “macroscopic” functional and behavioral disease manifestations that characterize the network’s function.

To study brain or cognitive disorders computationally, one first has to construct a model network that is capable of performing some basic functions, such as controlling movements or storing and retrieving information. Thereafter, by lesioning the intact network’s structural components or disrupting its dynamic mechanisms, the specific neuroanatomical and neurophysiological findings assumed to characterize the pathogenesis of the disease can be modeled, and the resulting changes in the behavior of the network can be examined. It is then also possible to search for mechanisms that may counteract the damaging effects of the simulated pathological alterations.

Neural models are limited in that they necessarily simplify the biological phenomena occurring in the nervous system and are generally constrained in size. The simulated lesions in such models are substantial simplifications of abnormal events occurring within the brain and/or in cognitive processes. Nevertheless, such computer-based models complement traditional methods of studying brain disorders in substantial and important ways. The size and location of simulated brain damage can be controlled precisely and can be systematically varied over arbitrarily large numbers of experimental “subjects” and information processing tasks. Further, the computational experiments are open to detailed inspection in ways that biological systems are not.

Neural models of brain and cognitive disorders, like neural network models in general, vary widely in the level of realism with which they aim to model the underlying phenomena. This is true both with regard to the level of biological detail employed in describing the individual building blocks themselves (the neurons and their interactions), and also with regard to the description level of the network’s architecture, i.e., to the extent the latter aims to reconstruct a specific brain region. In general, computational studies addressing neuropsychological or cognitive disorders tend to describe more abstract models, fairly removed from specific brain architectures, compared with the models addressing neurological and psychiatric disorders. Models of neuropsychological disorders are reviewed in *LESIONED NETWORKS AS MODELS OF NEUROPSYCHOLOGICAL DEFECTS*. Here we review some computational studies of a few major neurological and psychiatric disorders. In addition to studies of Alzheimer’s disease, Parkinson’s disease, and stroke reviewed in this chapter, neurological modeling studies have addressed a wide variety of other disorders, including multi-infarct dementia, migraine, and delirium. Our review of psychiatric disorders focuses on schizophrenia and affective disorders, but preliminary computational modeling studies have also addressed paranoid

disorders, dissociative disorders, and others. For recent papers describing these studies the interested reader is referred to Reggia, Ruppín, and Berndt (1996); Reggia, Ruppín, and Glanzman (1999); and Parks, Levine, and Long (1998).

Neurological Disorders

Many models of memory impairment have addressed various aspects of Alzheimer’s disease (Horn et al., 1993, Hasselmo, 1994), the most common dementing illness. The essential clinical feature of Alzheimer’s disease is a broad-based intellectual decline from previous levels of functioning, but memory impairment is a major clinical hallmark of the disease. These models have examined two main hypotheses concerning the pathogenesis of the disease. One hypothesis has been that *failure of neuronal synaptic compensatory mechanisms* (which in normal subjects successfully counteracts synaptic degenerative changes) plays a primary role in Alzheimer’s pathogenesis (Horn et al., 1993). Variations on the rate and exact functional form of synaptic compensation were used to define various compensation strategies, and these could account for the observed variation in the severity and progression rate of Alzheimer’s disease. The second hypothesis has focused on *synaptic runaway*, a pathological exponential growth of synaptic connections that may occur due to interference by previously stored memory patterns during the storage of new patterns (Hasselmo, 1994). Several factors can lead to the initiation of synaptic runaway, but once it occurs, its increased metabolic demands or excitotoxic effects could presumably be sufficiently severe to cause neuronal degeneration, parallel to that found in Alzheimer’s disease. Interestingly, the pattern of spread of pathological damage in Alzheimer’s disease fits well the hypothesis of synaptic runaway.

Recent work on modeling Alzheimer’s disease has focused on more realistic models incorporating spiking neurons. These models share the view that the hippocampus (and more generally, medial temporal lobe structures) plays a primary role in consolidation of memories in the cortex. Extending their previous work, Hasselmo and his co-workers have studied a fairly realistic model of the hippocampus, both in terms of its subdivision into various substructures and in terms of accounting for several known neuromodulatory effects on hippocampal memory processing. The latter enables one to address the cholinergic disturbances that are assumed to play an important role in memory and learning dysfunction in Alzheimer’s disease. The possible role of cholinergic neuromodulation of the hippocampus in Alzheimer’s disease has also been recently addressed by Menschick and Finkel (1998). Their work is performed in a realistic compartmental model of associative memory, and discusses the pathogenesis of memory decline in Alzheimer’s disease in dynamic terms that emerge from such spiking models, identifying a cascade of malfunctions occurring at multiple levels.

Models of stroke (sudden focal brain damage due to impaired regional blood flow) have been developed to address the events occurring immediately following the acute stroke event, and also later on during the chronic, reorganization phase. Naturally, these models differ in the kind of computational framework involved. Models of acute focal stroke encompass a combined neural/metabolic description that traces the temporal evolution of several variables that play a critical role in ischemic stroke (Revett et al., 1998). This work has examined the hypothesis that *cortical spreading depression* waves play a primary role in the spread of damage into the penumbral perinfarct region from the infarct core. It successfully reproduced several experimental dependencies and has made testable predictions about the number, velocity, and duration of

spreading depression waves. In contrast, studying chronic reorganization after focal stroke has been based on models simulating map formation in the cortex (Goodall et al., 1997). These models involve the projection of high-dimensional data on a two-dimensional cortical surface, and generate computational maps that mimic cortical “maps” representing relevant aspects of the external world (e.g., the homunculus in primary somatosensory or motor cortex). When a lesion is introduced into the simulated map, the model reorganizes such that the sensory surface originally represented by the lesioned area spontaneously reappears in adjacent cortical areas, as has been seen experimentally in animal studies. Two key hypotheses emerged from this modeling work. First, that postlesion map reorganization is a two-phase process, consisting of a rapid phase due to the dynamics of neural activity and a longer-term phase due to synaptic plasticity. Second, that increased perilesion excitability is necessary for useful map reorganization to occur. Similar self-organizing models, but based on cortical deaf-ferentation, have been used recently to support a theory of *phantom limb* experiences (Spitzer et al., 1994). This latter work presented an interesting solution, in neural network terms, to the ongoing controversy about the relative weight of central versus peripheral nervous system alterations in the pathogenesis of these disturbing symptoms.

Parkinson’s disease is an important motor and cognitive degenerative disorder. Its primary pathology has been traced to the degeneration of nigral dopaminergic neurons projecting on the striatum, and it has been a subject of quite a few modeling studies in recent years (see BASAL GANGLIA and DOPAMINE, ROLES OF). Again, the pathogenesis of the disease can be studied at different levels. In a more high-level model, Contreras-Vidal, Teulings, and Stelmach (1998) describe the activation of many of the pathways known to exist in basal ganglia in terms of a system of coupled differential equations, composing a gross-scale representation of basal ganglia structures as single dynamical variables. This model was able to produce handwriting patterns that were comparable to handwriting changes observed in Parkinson’s patients before and after L-DOPA treatment. On a lower-level of description, Kotter and Wickens (1998) have presented a more detailed, realistic model of the striatum, including effects of dopamine on the various receptor subtypes. This work suggests that dopamine therapy would tend to reverse only a subset of the changes produced by dopamine depletion, resulting in striatal dynamics that are significantly different from those encountered in the normal, premorbid state.

Psychiatric Disorders

Neural models have been created for a wide range of psychiatric disorders, but most of the work has focused on schizophrenia and affective disorders.

Schizophrenia is a clinically heterogeneous disorder with a broad spectrum of manifestations. Its symptoms include both “positive symptoms,” such as hallucinations, delusions, disorganized speech and behavior, and “negative symptoms,” such as loss of fluency of thought and speech, impaired attention, abnormalities in the expression and observation of emotion, and loss of volition and drive. The course of the illness tends to be marked by exacerbations and remissions, but the persistence of the impairment may give the disease a “dementia-like” quality in more advanced stages. The pathogenesis of schizophrenia is unknown. Perhaps the most enduring biochemical explanation of the pathophysiology of schizophrenia is the dopamine hypothesis, which postulates the coexistence of hypodopaminergic activity in the mesocortical system, resulting in negative symptoms, and hyperdopaminergic activity in the mesolimbic system, resulting in positive symptoms. Structural and functional imaging and neuroanatomical postmortem studies are providing converging evidence of the involvement of specific brain regions in schizophrenia, such as the prefrontal areas, tem-

poral lobes and the temporo-limbic circuitry, and subcortical circuitry. Integrative pathophysiological hypotheses have been proposed, but so far no single explanatory mechanism has prevailed.

Neural modeling of schizophrenia has taken two main paths, reflecting the view of schizophrenia as composed of positive symptoms that arise due to temporo-frontal pathology, and negative symptoms that are a result of prefrontal abnormalities. The first avenue has concentrated on modeling schizophrenic positive symptoms in the framework of an associative memory attractor network (Hoffman, 1987). In this framework, pathological alterations in an attractor neural network modeling excessive synaptic pruning can lead to the formation of *parasitic attractors*, whose cognitive and perceptual manifestations may play an important role in the emergence of schizophrenic delusions and hallucinations, by altering speech perception and production processes. In this line of research, Ruppel, Reggia, and Horn (1996) have modeled a frontal module as an associative memory neural network receiving its inputs from degenerating temporal projections and undergoing reactive synaptic regeneration. They have shown that while preserving memory performance, compensatory synaptic regenerative changes coupled with Hebbian activity-dependent synaptic changes may eventually lead to a *biased* retrieval distribution that is strongly dominated by few memory patterns, resembling the concentration of psychotic delusions and hallucinations on very few cognitive and perceptual themes.

Building upon their work on modeling the neuromodulatory effects of catecholamines on information processing, Cohen and Servan-Schreiber (1992) have presented a modeling study of the performance of normal subjects and schizophrenics in three attentional and language processing tasks. These tasks are important indices of cognitive dysfunction in schizophrenia, and are related to schizophrenic negative symptoms. In all of the tasks modeled, a backpropagation algorithm was used to train the networks to simulate normal performance. Although each task was modeled by a network designed specifically for that task, the networks used rely on similar information processing principles and share a common module for representing context, which is identified with the prefrontal cortex. The hypothesized neuromodulatory effects of dopamine on information processing were modeled as a global change of the input gain. Simulations demonstrated that a change in the gain of neurons in the context module can quantitatively account for the differences between normal and schizophrenic performance in the tasks examined. Postulating that the prefrontal cortex plays a central role in establishing context (see Braver and Cohen in Reggia et al., 1999), it has been proposed that dopamine might regulate context information in prefrontal cortex by providing an appropriate gating signal. Their model provides a mechanism for flexible updating of stored information, and is able to accurately simulate the performance breakdown of schizophrenics in the Continuous Performance Test.

More recently, Hoffman et al. have addressed the pathogenesis of schizophrenic positive symptoms from a neurodevelopmental perspective. They show that although the process of synaptic pruning can improve generalization of previously learned information, when taken to excess it would result in spontaneous percepts (hallucinations) and in “hyperpriming,” both seen in schizophrenia (Hoffman and McGlashan, 1997). This provides an interesting view of the pathogenesis of a disease by a normal developmental process that is taken “much too far.” Recent work has investigated the role of the prefrontal cortex in the pathogenesis of schizophrenia using more biologically realistic spike response neurons (see Reid and Willshaw in Reggia et al., 1999). They show that the ability of the prefrontal cortex to hold working memory information may be disrupted by changes in dopaminergic activity, reduced GABAergic activity, and reduced prefrontal connectivity, which have all been implicated in schizophrenia. As in the case of neural models of dementia and Alzheimer’s disease, the general trend is to move

from simplistic models employing McCulloch-Pitts binary neurons to networks using compartmental models of neurons and more realistic descriptions of synaptic transmission.

Attractor neural networks have also been considered as a framework for modeling cognitive manifestations of manic-depressive disorder. Manic bouts are characterized by a distinctly elevated, expansive or irritable mood, accompanied by "hyperactivity" symptoms. In contradistinction to schizophrenic positive symptoms, Hoffman (1987) has suggested that manic "hyperactivity" arises not as a result of structural damage leading to the formation of pathological attractors, but due to an increase in the noise levels resulting in enhanced rate of transition between attractors.

Past work related to major depression has concentrated on modeling learned helplessness, an experimental psychological model of depression, in an adaptive resonance network. More recently, a general model aimed at explaining how cognitive, emotional, and motor processes might influence one another has been presented (see Grossberg, in Reggia et al., 1999). Grossberg postulated opponent processing modules to control reinforcement learning in response to positive and negative reinforcers. Taking yet another approach to modeling major depression, it has recently been shown that the affective interference of depression might produce network overtraining and result in a network that responds excessively to negative stimuli and reinforces the dark obsessions characteristic of rumination.

Discussion

The conceptual and methodological challenges tackled by the studies surveyed here serve to illustrate the early stages of coalescence of a field that ambitiously endeavors to study the pathogenesis of brain disorders computationally. There is a large gap between the conceptual and modeling levels utilized in "realistic" computational studies of brain disorders versus more abstract ones. This gap arises in part because of the different disorders and phenomena addressed. It also reflects the long-standing controversy in the literature between more realistic "bottom-up" models and simpler, conceptual "top-down" models. In our current state of knowledge of the workings of the brain there is certainly room for both kinds of models.

The work reviewed in this paper demonstrates that neural models are a potentially useful methodological tool for examining the feasibility of theoretical hypotheses within a computational context. They can offer new insights into the experimental data, and may unify previously unrelated observations. Even the much simplified models reviewed here are sufficiently complicated to generate interesting and nontrivial predictions, as the feedback structure of the systems and processes involved makes the study of lesioned models a difficult and considerable challenge.

Future challenges and prospects of modeling brain disorders include:

1. *Modeling new experimental data:* Recent advances in several experimental techniques have yielded a number of promising developments. Of special interest to neural modelers are the developments in techniques that provide information on neural and synaptic degenerative processes. Those include neuroanatomical morphometric and immunochemical methods and magnetic resonance spectroscopy. Much hope for further advancement relies on the rapid development of functional imaging techniques, but a significant discrepancy remains between the scale of the distributed networks of brain activation revealed by current functional imaging studies and the scale of current neural models.
2. *Developing neural models of more complex cognitive functions:* current work has concentrated on making use of available neural

modeling tools. This has restricted the cognitive phenomena studied to memory-related processes, and to learning relatively simple tasks in a supervised manner. The development and incorporation of more sophisticated neural models is probably an essential step towards capturing more complex phenomena. Promising venues include models of reinforcement learning, multimodal associative memories, and multilayered recurrent networks.

The studies reviewed here are just the "end of the beginning." As more becomes known about the normal functioning of brain and cognitive systems, we shall be in a much better position to model their abnormalities. Some of the research projects in this field demonstrate another, perhaps not less promising, potential value of computational studies of brain disorders: to use the constraints imposed by such studies to learn more about the normal functioning of the brain by way of "reverse engineering." Computational modeling helps us to formulate our ideas precisely and study their consequences by making them explicit within simulation and analytical models. As such, we believe it will continue to develop as a fundamental research approach, working in a complementary manner with other research methodologies.

Road Map: Cognitive Neuroscience

Related Reading: Developmental Disorders; Dopamine, Roles of; EEG and MEG Analysis; Neuromodulation in Mammalian Nervous Systems

References

- Cohen, J. D., and Servan-Schreiber, D., 1992, Context, cortex, and dopamine: A connectionist approach to behavior and biology in schizophrenia, *Psychol. Rev.*, 99(1):45-77.
- Contreras-Vidal, J. L., Teulings H. L., and Stelmach G. E., 1998, Neural dynamics of short and medium-term motor control effects of levodopa therapy in parkinson's disease, *Artif. Intell. Med.*, 13(1-2):57-80.
- Goodall, S., Reggia, J., Chen, Y., Rupp, E., and Whitney, C., 1997, A computational model of acute focal cortical lesions, *Stroke*, 28:101-109.
- Hasselmo, M. E., 1994, Runaway synaptic modification in models of the cortex: Implications for Alzheimer's disease, *Neural Networks*, 7(1):13-40.
- Hoffman, R. E., 1987, Computer simulations of neural information processing and the schizophrenia-mania dichotomy, *Arch. Gen. Psychiatry*, 44:178.
- Hoffman, R. E., and McGlashan, T. H., 1997, Synaptic elimination, neurodevelopment and the mechanism of hallucinated voices in schizophrenia, *Am. J. Psychiatry*, 154:1683-1689.
- Horn, D., Rupp, E., Usher, M., and Herrmann, M., 1993, Neural network modeling of memory deterioration in alzheimer's disease, *Neural Computation*, 5:736-749.
- Kotter, R., and Wicks, J., 1998, Striatal mechanisms in parkinson's disease: New insights from computer modeling, *Artif. Intell. Med.*, 13(1-2):37-56.
- Menschick, E. D., and Finkel, L. H., 1998, Neuromodulatory control of hippocampal function: towards a model of Alzheimer's disease, *Artif. Intell. Med.*, 13:99-121.
- Parks, R. W., Levine, D. S., and Long, D. L., 1998, *Fundamentals of Neural Network Modeling: Neuropsychology and Cognitive Neuroscience*, Cambridge, MA: MIT Press. ♦
- Reggia, J., Rupp, E., and Berndt, R., 1996, *Neural Modeling of Brain and Cognitive Disorders*, Singapore: World Scientific. ♦
- Reggia, J., Rupp, E., and Glanzman, D., 1999, *Brain, Behavioral and Cognitive Disorders: The Neurocomputational Perspective*. Progress in Brain Research Series, Amsterdam: Elsevier Science Publishers. ♦
- Revet, K., Rupp, E., Goodall, S., and Reggia, J., 1998, Spreading depression in focal ischemia: A computational study, *J. Cerebral Blood Flow Metab.*, 18(9):998-1007.
- Rupp, E., Reggia, J., and Horn, D., 1996, A neural model of positive schizophrenic symptoms, *Schizophrenia Bull.*, 22(1):105-123.
- Spitzer, M., Bohler, P., Weisbrod M., and Kischka, U., 1994, A neural network model of phantom limbs, *Biol. Cybernet.*, 72:197-206.

Neuromanifolds and Information Geometry

Shun-ichi Amari

Introduction

A neural network is specified by its architecture and by a number of parameters consisting of connections or synaptic weights, together with bias terms or thresholds. These parameters are modifiable, and learning or self-organization is carried out by changing them. Let us denote all of these parameters by a vector $\theta = (\theta_1, \dots, \theta_n)$, where n is the number of the parameters, and consider the parameter space where θ is its coordinate system. Any neural network of this architecture is specified by a point θ in the parameter space, so that we identify the parameter space with the set of all the neural networks.

Neural networks are regarded as stochastic nonlinear systems. Some models are intrinsically stochastic, because of the stochastic nature of neural firing. These models include Boltzmann machines (see SIMULATED ANNEALING AND BOLTZMANN MACHINES) and Bayesian or belief-propagation networks (see BAYESIAN NETWORKS). Some are deterministic but work in the noisy environment, so that its behaviors are not free from stochastic fluctuations. A typical example is a multilayer perceptron.

Learning takes place in the parameter space, and a learning process is represented by a trajectory. Here an important problem arises: What is the geometrical structure of the parameter space, and how do learning behaviors depend on its structure? Information geometry, which originated in studies of the manifolds of probability distributions, can be used to answer these questions (Amari and Nagaoka, 2000). It defines a Riemannian metric and a pair of dual affine connections. It gives a geometrical framework to elucidate problems related to stochastic phenomena, so that it is useful not only for statistical inference and information theory but also for neural networks.

This article introduces geometrical structures in the parameter space of neural networks known as neuromanifolds and elucidates how the dynamical behaviors of neural learning are related to the underlying geometrical structures. We use multilayer perceptrons and Boltzmann machines as examples. Geometrical ideas are also applied to support vector machines (Burges, 1999; Amari and Wu, 1999), boosting methods (Lebanon and Lafferty, 2001), and many others. The principles and practical implementations of natural gradient learning (Amari, 1998) are described first.

Neuromanifold of Multilayer Perceptrons

Let us consider a multilayer perceptron consisting of one hidden layer with h hidden neurons and one output neuron. Let its inputs, x_1, \dots, x_k , be denoted by an input vector $\mathbf{x} = (x_1, \dots, x_k)$. The i th hidden neuron receives \mathbf{x} and calculates its weighted sum, $u_i = \mathbf{w}_i \cdot \mathbf{x} = \sum_j w_{ij}x_j$, where $\mathbf{w}_i = (w_{i1}, \dots, w_{ik})$ is the synaptic weight vector. It then emits a nonlinear sigmoidal function $\phi(u_i)$ as its output, where $\phi(u) = \tanh(u)$ is the hyperbolic tangent.

The output neuron summarizes all the outputs of the hidden neurons with weights $\mathbf{v} = (v_1, \dots, v_h)$, and emits its output. The sum is disturbed by Gaussian noise n , so that the overall input-output relation is written as

$$y = \sum_{i=1}^h v_i \phi(\mathbf{w}_i \cdot \mathbf{x}) + n \quad (1)$$

The parameter space of the multilayer perceptron is specified by the set of all the parameters $\theta = (w_{11}, \dots, w_{1k}, w_{21}, \dots, w_{hk}, v_1, \dots, v_h)$. It can be seen that each point θ corresponds to the nonlinear function $f(\mathbf{x}, \theta) = \sum_i v_i \phi(\mathbf{w}_i \cdot \mathbf{x})$ of the perceptron. However, since the behavior of each perceptron is represented by the probability distribution of the output y given \mathbf{x} ,

$$p(y|\mathbf{x}, \theta) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} (y - f(\mathbf{x}, \theta))^2 \right\} \quad (2)$$

we may identify each point θ with the above probability distribution.

We first address the problem of the identifiability of the parameters of neural networks. When two networks with different parameters θ and θ' have the same input-output behavior, it is not possible to identify the parameters from the behaviors. Such networks are said to be equivalent. Two types of unidentifiability are known (Chen, Lu, and Hecht-Nielsen, 1993):

1. *Permutation*: Permutation of the number of hidden units does not change a network's behavior.
2. *Sign change*: Sign change of both \mathbf{w}_i and v_i at the same time does not change its behavior, because $v_i \phi(\mathbf{w}_i \cdot \mathbf{x}) = -v_i \phi(-\mathbf{w}_i \cdot \mathbf{x})$.

The above transformation induces the following equivalent networks (Rüger and Ossens, 1997): When the synaptic weight vectors \mathbf{w}_i and \mathbf{w}_j are equal, the two hidden neurons can be merged into one neuron, where $v' = v_i + v_j$ gives a new connection weight to the output neuron without changing its behavior. Hence, on the submanifold defined by $\mathbf{w}_i = \mathbf{w}_j$, two networks are equivalent when $v_i + v_j$ are equal. It also happens that, in the submanifold defined by $v_i = 0$, whatever value \mathbf{w}_i takes, $v_i \phi(\mathbf{w}_i \cdot \mathbf{x}) = 0$. Hence, whatever \mathbf{w}_i is, the behaviors are the same, and this neuron can be removed. The same holds when $\mathbf{w}_i = 0$. We call these submanifolds defined by $\mathbf{w}_i = \mathbf{w}_j$ or $v_i |\mathbf{w}_i| = 0$ the critical submanifolds. Parameters are not identifiable on critical submanifolds.

When we consider equivalent networks as one object, the equivalent points in the neuromanifold will be merged into one point. For example, the submanifold defined by $v_i = 0$ reduces to one point. The reduced manifold has singularities on which dimensions are reduced. This causes a singular topological structure in the reduced neuromanifold.

It is usual to use the Kullback-Leibler divergence (Cover and Thomas, 1991) as a measure of divergence defined by two probability distributions $p(\mathbf{x})$ and $q(\mathbf{x})$:

$$KL[p(\mathbf{x}) : q(\mathbf{x})] = E_p \left[\log \frac{p(\mathbf{x})}{q(\mathbf{x})} \right] = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \quad (3)$$

where E_p is expectation with respect to $p(\mathbf{x})$. This quantity is non-negative and is equal to 0 when and only when $p(\mathbf{x}) = q(\mathbf{x})$, but it is not symmetric; that is, $KL[p : q] \neq KL[q : p]$ in general. The KL divergence is related to information theory. Let X, Y be two random variables whose joint probability is $p(x, y)$ and whose marginal distributions are $p_X(x)$ and $p_Y(y)$, respectively. The probability distribution $p_X(x)p_Y(y)$ is different from $p(x, y)$ except for the case in which X and Y are independent. How much are the random variables X and Y related? This is measured by the Shannon mutual

information $I[X : Y]$, which is equal to the KL divergence, in this way:

$$I[X : Y] = KL[p(x, y) : p_X(x)p_Y(y)] \quad (4)$$

When the probability distribution is given by a parametric form $p(\mathbf{x}, \boldsymbol{\theta})$ (or conditional distribution $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$, in the present case), the KL divergence $D(\boldsymbol{\theta} : \boldsymbol{\theta}')$ is given in the parameter space by

$$D[\boldsymbol{\theta} : \boldsymbol{\theta}'] = KL[p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) : p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}')] \quad (5)$$

When the two points are close, we put $\boldsymbol{\theta}' = \boldsymbol{\theta} + d\boldsymbol{\theta}$. Then, by Taylor expansion, we have the quadratic form

$$KL[p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) : p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta} + d\boldsymbol{\theta})] = \frac{1}{2} d\boldsymbol{\theta}^T G d\boldsymbol{\theta} \quad (6)$$

where

$$G(\boldsymbol{\theta}) = E \left[\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})^T \right] \quad (7)$$

is a matrix called the Fisher information matrix, and the quadratic form is regarded as the square of the distances between two nearby points $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + d\boldsymbol{\theta}$. Here, $\partial/\partial \boldsymbol{\theta}$ is the gradient and T denotes transposition of a column vector.

The Fisher information matrix is a measure concerning how much information is obtained by observing a random variable, in order to estimate the underlying distribution. On the other hand, the Shannon information matrix is a measure concerning how much information is obtained concerning random variable X when another random variable Y is observed. Hence, they are used for different purposes, although they are derived from the same KL divergence.

A manifold is said to be Riemannian when the square of the distance between two nearby points is given by a quadratic form like that in Equation 6, based on a symmetric positive matrix G . The matrix is called the Riemannian metric.

The neuromanifold is a Riemannian space, having the Fisher information matrix G as its Riemannian metric. It is positive-definite in general, but it degenerates on the critical submanifolds of the neuromanifold of multilayer perceptrons, that is,

$$d\boldsymbol{\theta}^T G d\boldsymbol{\theta} = 0 \quad (8)$$

on a critical submanifold, when $d\boldsymbol{\theta}$ is the direction of unidentifiability. This reflects the fact that the distance between two equivalent points is 0. Such a manifold may be said to be pseudo-Riemannian. Hence, the Riemannian structure accounts for the topological singularity in the reduced manifold.

The gradient of a function $f(\boldsymbol{\theta})$ represents the direction of the steepest change of the function f in a Euclidean space. "The steepest" implies that, when $\boldsymbol{\theta}$ changes by $d\boldsymbol{\theta}$ with a small fixed length, say $|d\boldsymbol{\theta}|^2 = \varepsilon^2$, the change of f , $\Delta f = f(\boldsymbol{\theta} + d\boldsymbol{\theta}) - f(\boldsymbol{\theta})$, is largest in the direction of the gradient $d\boldsymbol{\theta} \propto \nabla f(\boldsymbol{\theta})$. In the case of a Riemannian space, the distance is defined by the quadratic form given in Equation 6. The steepest direction is then given by the natural or Riemannian gradient (Amari, 1998).

$$\tilde{\nabla} f = G^{-1} \nabla f \quad (9)$$

Natural Gradient Learning in a Neuromanifold

Learning takes place in a neuromanifold by modifying the current parameters $\boldsymbol{\theta}$, depending on the current input and output pair $(\mathbf{x}_t, \mathbf{y}_t)$ at time $t = 1, 2, \dots$, in a training set of examples. Let $l(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})$ be a loss function that is to be minimized through learning. A popular loss is the square of errors of the outputs. The backpropagation learning rule is given by the gradient method

$$\Delta \boldsymbol{\theta} = -\eta \nabla l(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) \quad (10)$$

However, backpropagation learning is very slow. The error rate decreases quickly in the early stages of learning, but soon the decrement becomes very small. Such a position is called a plateau, and it takes a long time to get rid of it. Plateaus are not local minima but saddle points.

It is known from the statistical-physical method that plateaus result from the permutation symmetry of the hidden neurons and thus are ubiquitous (Ratnay and Saad, 1999). Moreover, a learning trajectory is usually attracted to such a saddle point, and convergence slows greatly. Such a phenomenon is caused by the geometrical structure of the neuromanifold, corresponding to its topological and metrical properties.

Plateaus mostly occur on critical submanifolds where the parameters are not identifiable. A change in the parameters around a critical submanifold causes only a negligibly small improvement in its behaviors. However, if we take the metrical structure into account, the gradient should be replaced by the Riemannian one, the natural gradient. Then, the plateau phenomenon given rise to by unidentifiability disappears. The natural gradient learning rule is given by

$$\Delta \boldsymbol{\theta} = -\eta G^{-1} \nabla l(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) \quad (11)$$

where η is a learning rate (Amari, 1998). It should be noted that G is singular on a critical submanifold, so that G^{-1} diverges there. This has an effect of preventing the parameters $\boldsymbol{\theta}$ from approaching critical submanifolds, thus avoiding plateaus.

This method is equivalent locally to the Newton method, implying superlinear convergence. However, the merit of natural gradient learning lies not only in the speed with which local convergence can be achieved but also in the avoidance of plateaus in the learning process, which is the main obstacle slowing convergence. In general, natural gradient learning differs from a second-order method such as Newton's in that natural gradient learning depends on both the Riemannian structure and the cost function, while the second-order method takes only the second derivatives of the cost function.

In general, it is difficult to calculate and invert the Fisher information matrix G . In order to overcome this difficulty, the *adaptive natural gradient method* has been proposed (Amari, Park and Fukumizu, 2000) in which the inverse G^{-1} is estimated by an adaptive method, by changing the current G^{-1} into $G^{-1} + \Delta G^{-1}$,

$$\Delta G^{-1} = \eta' G^{-1} \nabla l(G^{-1} \nabla l)^T \quad (12)$$

where η' is another learning rate.

In the special case of the squared loss, the adaptive natural gradient method is equivalent to the adaptive version of the Gauss-Newton method, although the motivation is quite different. However, the adaptive natural gradient method is used for many other types of learning problems, including the Kullback-Leibler loss (Park, Amari, and Fukumizu, 2000), INDEPENDENT COMPONENT ANALYSIS (q.v.) (Hyvarinen et al., 2001), and others.

Information Geometry of Boltzmann Machines and EM Algorithm

We now introduce a more advanced theory of information geometry, in which a manifold has a pair of dual affine connections in addition to the Riemannian metric (Amari and Nagaoka, 2000). We will defer the mathematical details in favor of an intuitive explanation. For the manifold consisting of probability distributions, a Riemannian metric is given by the Fisher information matrix. How is a geodesic defined in such a manifold? The Riemannian geodesic

is given by the shortest path connecting two points. Two other types of geodesics, called the e -geodesic and m -geodesic, are introduced as follows:

Given two probability distributions $p(x)$ and $q(x)$, the e -geodesic connecting them is a curve $r_e(x, t)$ given by

$$\log r_e(x, t) = (1 - t) \log p(x) + t \log q(x) + c(t) \quad (13)$$

where t is the parameter of the curve and c is a normalization constant. In other words, an e -geodesic connects two distributions linearly in the logarithmic scale. Such a curve is an exponential family. The m -geodesic connecting them is given by

$$r_m(x, t) = (1 - t)p(x) + tq(x) \quad (14)$$

In other words, it connects two distributions linearly, giving a mixture family.

We next define the orthogonality. In a Riemannian manifold, two curves $\theta_1(t)$ and $\theta_2(t)$ that intersect at $t = 0$, $\theta_1(0) = \theta_2(0)$ are orthogonal at the intersection when the inner product of their tangents $\dot{\theta}_1(0)$ and $\dot{\theta}_2(0)$ is 0:

$$\langle \dot{\theta}_1, \dot{\theta}_2 \rangle = \dot{\theta}_1^T G \dot{\theta}_2 = 0 \quad (15)$$

Here $\dot{\theta}_i = (d/dt)\theta_i(t)$ represents the tangent of curve $\theta_i(t)$. In the present case, the two curves $r(x, t)$ and $q(x, t)$, $r(x, 0) = q(x, 0)$ are orthogonal when

$$\langle \dot{r}(x, 0), \dot{q}(x, 0) \rangle = \int \frac{\dot{r}(x, 0)\dot{q}(x, 0)}{r(x, 0)} dx = 0 \quad (16)$$

The following is a fundamental theorem of a dually flat manifold (Figure 1).

Generalized Pythagoras Theorem. For three distributions $p(x)$, $q(x)$, and $r(x)$, when the m -geodesic connecting p and q intersects the e -geodesic connecting q and r orthogonally at q ,

$$KL[p : q] + KL[q : r] = KL[p : r] \quad (17)$$

The Boltzmann machine (see SIMULATED ANNEALING AND BOLTZMANN MACHINES) is a recurrently connected stochastic neural network whose behavior is directly connected with a probability distribution. Hence, its performance is elucidated by information geometry. The state of a Boltzmann machine is specified by vector $\mathbf{x} = (x_i)$, where x_i is 1 when the i th neuron is excited and is 0 otherwise. The state changes stochastically at discrete times. The next state \mathbf{x}' is determined as follows. Choose one neuron, say j , randomly. Then, x'_j (the j th component of \mathbf{x}') is determined to be equal to 1 with probability related to $u_j = \sum w_{ji}x_i$, where $\mathbf{w} = (w_{ji})$ are the connection weights between neurons i and j . Here, $w_{ji} = w_{ij}$ and $w_{ji} = 0$ are assumed. The state transition of a Boltzmann machine is described by a symmetric Markov chain. Its stable distribution is explicitly given by

$$p(\mathbf{x}; \mathbf{w}) = c \exp \{-E(\mathbf{x})\} \quad (18)$$

$$E(\mathbf{x}) = -\frac{1}{2} \sum w_{ij}x_i x_j \quad (19)$$

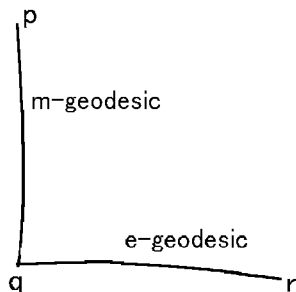


Figure 1. Pythagorean relation in information geometry.

When the Boltzmann machine is working for a long period, state \mathbf{x} appears with relative frequency $p(\mathbf{x}; \mathbf{w})$. A Boltzmann machine is used to simulate an environmental information source that generates signal \mathbf{x} with relative frequency $q(\mathbf{x})$. To this end, we need to train a Boltzmann machine by modifying the synaptic connections $\mathbf{w} = (w_{ij})$ such that $p(\mathbf{x}; \mathbf{w})$ approximates $q(\mathbf{x})$, by using the training data $D_T = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$.

A Boltzmann machine has visible neurons and hidden neurons, where the hidden neurons control the behaviors of the visible neurons. We first explain the simplest case without hidden neurons. Let S be the set of all the probability distributions over the state set $X = \{\mathbf{x}\}$. Since there are 2^n states over n neurons, a probability distribution $q = \{q(\mathbf{x})\}$ over these states specifies 2^n probabilities $q(\mathbf{x})$ for all $\mathbf{x} \in X$. Since

$$\sum_{\mathbf{x} \in X} q(\mathbf{x}) = 1 \quad (20)$$

holds, q has $2^n - 1$ degrees of freedom. Geometrically, this implies that S is a $(2^n - 1)$ -dimensional manifold. The probability distributions $p(\mathbf{x}; \mathbf{w})$ realized by Boltzmann machines are of the form given by Equation 18, having only $0.5n(n + 1)$ degrees of freedom $\mathbf{w} = (w_{ij})$. Therefore, the set B of the probability distributions realized by Boltzmann machines is a $0.5n(n + 1)$ -dimensional submanifold included in the larger manifold S .

Given a training set D_T from the environment distribution q , we need to obtain $p \in B$, which approximates q best. The criterion of approximation is to minimize the Kullback divergence or relative entropy of q and p :

$$D(q||p) = \sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \quad (21)$$

Information geometry elucidates the geometrical structure of S and B , and the optimal approximator is easily obtained (Amari, Kurata, and Nagaoka, 1992). Let $\hat{p} \in B$ be the minimizer of $D(q||p)$. Then, from the Pythagoras theorem, the m -geodesic connecting q and \hat{p} is orthogonal to B , that is, orthogonal to any curves in B . Such a point is called the m -projection or q to B . Hence the optimal approximator of q is given by its m -projection.

In the general case, neurons are divided into two parts, hidden neurons and visible neurons. The state \mathbf{x} is also divided into two parts, \mathbf{x}_V and \mathbf{x}_H , $\mathbf{x} = (\mathbf{x}_V, \mathbf{x}_H)$. The stable distribution of \mathbf{x}_V is

$$p(\mathbf{x}_V; \mathbf{w}) = \sum_{\mathbf{x}_H} p(\mathbf{x}_V, \mathbf{x}_H; \mathbf{w}) \quad (22)$$

which is more general than those specified by Boltzmann machines (cf. Equation 18) without hidden units.

The training data D_T are given only to the visible neurons, and no information is available for the states of the hidden neurons. However, it is more flexible to adjust $p(\mathbf{x}_V)$ to fit $q(\mathbf{x}_V)$ of the environment, by modifying the connection weights, including the hidden neurons. Let S be the set of all the joint probability distributions $q(\mathbf{x}_V, \mathbf{x}_H)$. Let B be the joint distributions $p(\mathbf{x}_V, \mathbf{x}_H; \mathbf{w})$ realized by the Boltzmann machines. Since the training data D_T specify relative frequencies $q(\mathbf{x}_V)$ of the visible part only, the marginal distribution $q(\mathbf{x}_V)$ is available from D_T . Let \tilde{D} be the set of all the probability distributions $q(\mathbf{x}_V, \mathbf{x}_H)$ whose marginal distribution is specified by the training data D_T . We can prove that \tilde{D} is an m -flat submanifold in S .

The best approximation is to minimize $D\{q(\mathbf{x}_V)||p(\mathbf{x}_V; \mathbf{w})\}$. A recursive procedure to obtain the best approximation is shown (Amari, 1995). Starting from any initial guess $p_1 \in B$, project it by the e -geodesic to \tilde{D} , obtaining $q_1 \in \tilde{D}$. Then, project q_1 to B by the m -geodesic, obtaining a new candidate $p_2 \in B$. Here, two dualistic notions of the e -geodesic and m -geodesic play a fundamental role. It is interesting that this coincides with the Expectation-Maximization (EM) algorithm known in statistics. Here, the e -

projection part corresponds to taking the expected value of the unknown (hidden) frequencies of $q(\mathbf{x}_V, \mathbf{x}_H)$ when the candidate is p , as is given by the E procedure in the EM algorithm. The m -projection part is maximization of the likelihood, deriving the maximum likelihood estimator p from q .

Conclusion

Information geometry provides a mathematical structure that originated in the study of the intrinsic geometry of manifolds of probability distributions. It gives a Riemannian metric together with a dual pair of affine connections, where generalizations of the Pythagoras theorem and projection theorem hold. It is applied to various stochastic models in many fields of research, including neural networks. Further developments can be found in Watanabe (2001).

Road Map: Learning in Artificial Networks

Related Reading: Data Clustering and Learning; Learning and Statistical Inference; Model Validation; Simulated Annealing and Boltzmann Machines; Support Vector Machines

References

- Amari, S., 1995, Information geometry of the EM and em algorithms for neural networks, *Neural Netw.*, 8:1379–1408.
- Amari, S., 1998, Natural gradient works efficiently in learning, *Neural Computat.*, 10:251–276.
- Amari, S., Kurata, K., and Nagaoka, H., 1992, Information geometry of Boltzmann machines, *IEEE Trans. Neural Netw.*, 3:260–271.
- Amari, S., and Nagaoka, H., 2000, *Methods of Information Geometry*, New York: AMS and Oxford University Press. ♦
- Amari, S., Park, H., and Fukumizu, K., 2000, Adaptive method of realizing natural gradient learning for multilayer perceptions, *Neural Computat.*, 12:1399–1409.
- Amari, S., and Wu, S., 1999, Improving support vector machine classifiers by modifying kernel functions, *Neural Netw.*, 12:783–789.
- Burges, C. J. C., 1999, Geometry and invariance in kernel based methods, in *Advances in Kernel Methods* (B. Schölkopf, et al., Eds.), Cambridge, MA: MIT Press, pp. 89–116.
- Cover, T. M., and Thomas, J. A., 1991, *Elements of Information Theory*, New York: Wiley. ♦
- Chen, A. M., Lu, H., and Hecht-Nielsen, R., 1993, On the geometry of feedforward neural network error surfaces, *Neural Computat.*, 5:910–927.
- Hyvärinen, A., Karhunen, J., and Oja, E., 2001, *International Component Analysis*, New York: Wiley.
- Lebanon, G., and Lafferty, J., 2001, *Boosting and Maximum Likelihood for Exponential Models*, Technical Report CMU-CS-01-144, Pittsburgh, PA: Carnegie Mellon University, School of Computer Science.
- Park, H., Amari, S., and Fukumizu, K., 2000, Adaptive natural gradient learning algorithms for various stochastic models, *Neural Netw.*, 13:755–764.
- Ratnay, M., and Saad, D., 1999, Analysis of natural gradient descent for multilayer neural networks, *Phys. Rev. E*, 59:4523–4532.
- Rüger, S. M., and Osses, A., 1997, The metric of weight space, *Neural Process. Lett.* 5:63–72.
- Watanabe, S., 2001, Algebraic analysis for non-identifiable learning machines, *Neural Computat.*, 13:899–933.

Neuromodulation in Invertebrate Nervous Systems

Patsy S. Dickinson

Introduction

Because animals live in changing environments, behavior and hence nervous system output must be flexible. This flexibility manifests itself in several ways that are important when using either experimental studies or modeling to understand neural network function. First, neurons are not all alike; they show a rich variety of conductances that endow them with different functional properties. Second, these properties and hence the collective activity of the networks to which the neurons belong are not fixed, but are subject to modulation that can change their characteristics and output. Modulation as a result of both locally released neuromodulators and more widely acting hormones differs qualitatively from the moment-to-moment integration of synaptic excitation and inhibition that a neuron receives. Neuromodulators can provoke dynamic changes in neurons or circuits on time scales ranging from seconds to days.

Neuronal membrane properties and synapses, neuromuscular junctions, and muscle properties are all subject to modulation. Together, these modulations allow the same groups of neurons to generate diverse arrays of behaviors and to respond appropriately to a wide variety of sensory stimuli. Frequently, the same modulators act at multiple levels to influence or bias motor output. Because of their relative simplicity and accessibility, invertebrate nervous systems have provided the clearest examples of modulation and its importance to neuronal output.

Temporal and Spatial Dynamics of Neuromodulation

Neuromodulators act on a variety of both temporal and spatial scales owing to the nature of release and breakdown as well as the

mechanisms through which they exert their effects (Figure 1). Some modulators are hormones; thus, they are spatially widespread and show relatively slow temporal changes. In *Aplysia*, for example, the bag cells release several peptides that change the activity of numerous other neurons, and of themselves via autoreceptors, for up to 18 hours (Levitan and Kaczmarek, 2002). Alternatively, other modulators are released at defined neuropilar locations and may be rapidly broken down. In addition to the amines, which often function as modulators, some peptides, such as proctolin in the crustacean stomatogastric nervous systems, are rapidly broken down by peptidases (Nusbaum et al., 2001). Moreover, the same transmitter substance may be released both hormonally and neuronally. At the same time, the cellular mechanisms by which modulators act may selectively potentiate the effects of some modulators. For example, some modulators activate second messengers, which may outlast the presence of the modulator itself. Others may act on relatively short time scales, so cellular properties in a rhythmic network, for example, may vary over the course of each cycle. Because of these dynamics, networks are continuously refined and reconfigured during behavior (Marder, 1997), and such dynamics must be incorporated into models if they are to explain the flexibility observed in behaving animals.

Factors Determining Variability in Neuromodulator Effects

The effects of a given neuromodulator on its target neuron(s) or network(s) depend on a number of other factors. First, many neuromodulator influences may be dependent on the concentration of

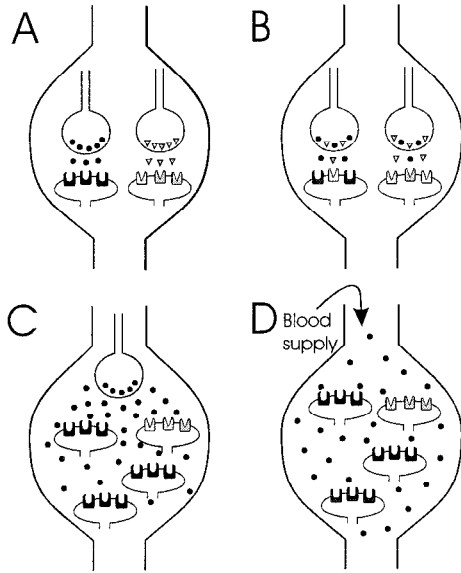


Figure 1. Modulatory transmitters can affect other neurons on a number of spatial scales and in a variety of combinations. For example, each of two different modulatory neurons could release a modulator into specific neuropilar regions, resulting in “pointwise” modulation that is limited in both time and space (A). Alternatively, two or more co-transmitters (one or both of which are modulatory) could be released from the same synaptic terminals (B). The two substances need not act simultaneously on the same postsynaptic terminals. Instead, they could be released by differential spiking frequencies or patterns; moreover, not all postsynaptic neurons necessarily have receptors for both modulatory compounds. A third alternative (C) is more widespread release from nonsynaptic sites, such that the transmitter diffuses to influence a larger group of neurons, with a gradient as a function of distance from the presynaptic release site. Finally, modulators can be released into the bloodstream and thus delivered to all neurons (D), although they affect only those with appropriate receptors.

transmitter present. In most experiments using bath application, there is a dose-dependent effect. When modulators are released from neurons, the amount of transmitter released can be a function not only of spike frequency in the modulatory neuron, but also of the pattern with which the neuron fires. For example, small molecule transmitters may be released with each spike, but peptides and amines, which are frequently modulatory, may be released only when the neuron fires in high-frequency bursts. Thus, the effect of a modulatory neuron firing tonically at 10 Hz may differ both qualitatively and quantitatively from the effects of the same neuron firing 0.5-s, 40-Hz bursts every 2 s, even though the average spike frequency is unchanged. In the *Aplysia* motor neuron B15, for example, acetylcholine is released with each action potential, but the modulatory peptides (small cardioactive peptides, SCs) that it contains are released only with elevated (e.g., 25–50 Hz) spike frequencies (Whim, Church, and Lloyd, 1994). Peptide release from this neuron does not depend on spike frequency alone, but rather on complex interactions between spike frequency, burst duration, and interburst interval. Additionally, it should be noted that while peptide release is often pattern sensitive, this is not always the case, as illustrated by another motor neuron in *Aplysia*, which likewise releases the SCs, but does so even at very low firing frequencies (<1 Hz; Whim et al., 1994).

Second, many modulatory neurons contain more than one transmitter. These co-transmitters may be released differentially as a function of neuronal firing pattern, as described above. Likewise, they may be released differentially at different neuronal terminals

(Nusbaum et al., 2001). Additionally, they may act differentially on different postsynaptic neurons as a function of the availability of different postsynaptic receptors. The situation becomes even more complex when we realize that the same neuron can contain co-transmitters that exert opposing effects on the same postsynaptic targets, so they may be excitatory under certain conditions yet be inhibitory under other firing regimes. Finally, the release of modulatory peptides may itself be modulated by other neurotransmitters. In *Aplysia* motor neuron B1, for example, serotonin increases excitability but simultaneously decreases the amount of the SCs it releases (Whim et al., 1994).

Third, the effect of a given modulator frequently depends on the state of the system when the modulator is applied. Modulators themselves may affect this state, thereby altering the responses of the network to other inputs. In the lobster stomatogastric system, for example, the peptide proctolin by itself has no effect on the cardiac sac pattern. However, if proctolin is applied shortly after red pigment-concentrating hormone (RPCH) is applied, it elicits strong cardiac sac bursting (Dickinson et al., 1997). In formulating models of such systems, it should be remembered that the response of a system to one modulator may not always be identical, but instead may be a longer-term function of the modulatory history of the animal.

Modulation of Sensory Systems

The sensory information that an animal needs depends on a number of factors, including its activity patterns and motivational state. Thus, the sensitivities of many sensory receptors can be modulated, as is seen for a stretch receptor, the oval organ, in crustaceans. This organ contains three sensory afferents and provides proprioceptive feedback to the gill ventilatory system. Proctolin increases the amplitude of the receptor potential and hence the number of action potentials produced in two of these afferents, whereas octopamine and serotonin decrease these responses. Interestingly, the dendrites within the oval organ itself contain proctolin, suggesting that receptor activity might automodulate receptor sensitivity in that increased activity would induce greater proctolin release, thereby increasing receptor gain (Pasztor, 1989).

In addition to receptor sensitivity itself, the extent to which sensory information is conveyed to central and motor systems is subject to modulatory control. This has been extensively examined in the sensorimotor synapses of *Aplysia* in the context of learning, but is also prevalent on shorter time scales associated with ongoing activity of many motor systems. In a number of locomotive systems in arthropods, for example, it has been shown that reflexes that stabilize posture in a quiet animal can change not only in magnitude, but also in direction when the animal begins to move. Such reflex reversal involves a number of mechanisms at both the presynaptic and postsynaptic levels (Clarac, Cattaert, and LeRay, 2000). Moreover, the strength of the reflex can vary cyclically as a function of the ongoing movement, again underscoring the importance of temporal dynamics in neuromodulation.

Modulation at Central Levels

Modulators can activate, terminate, or modify rhythmic pattern-generating networks. The detailed outputs of many rhythmic patterns are highly variable; frequency, phase relationships within the pattern, and number of participating neurons are subject to change. Additionally, many neurons and/or muscles participate in more than one behavior. Thus, in modeling networks, the mechanisms by which modulators sculpt specific patterns of activity from more generalized pattern generators must be considered. Getting and De-kin (reviewed in Harris-Warrick and Marder, 1991), who first described such networks as “polymorphic,” showed that the same

network could be reconfigured by modulatory inputs to produce either escape swimming or reflexive withdrawal in the mollusk *Tritonia*.

Intrinsic Versus Extrinsic Neuromodulation

Until recently, most modulators studied were located in control centers removed from the target network, and hence the release of transmitter was independent of the network and neurons being modulated. It is now clear, however, that neurons within a network may release modulators that act on the same network (Katz, 1998). With such "intrinsic neuromodulation," the pattern of modulator release thus depends on the activity of the network of neurons being modulated. In the CPG network controlling swimming in *Tritonia*, for example, one neuronal type, the dorsal swim interneurons, releases the transmitter serotonin. Serotonin not only reciprocally inhibits the ventral swim interneurons within the pattern generator, but also modulates the strength of other synapses within the network (Katz, 1998). Serotonin levels thus fluctuate dynamically during the course of a swim, and recent models have shown that this intrinsic modulation is a critical component in the production of the swim motor pattern (Frost et al., 1997).

Alteration of Intrinsic Properties of Neurons

Many neurons have voltage-dependent conductances that allow them to generate rhythmic bursts of action potentials in the absence of synaptic input (see OSCILLATORY AND BURSTING PROPERTIES OF NEURONS for further details). However, many of these conductances are activated only in the presence of an appropriate neuromodulator. For example, the Anterior Burster neuron in the lobster pyloric network does not oscillate when completely isolated from its network partners and modulatory inputs. However, the isolated neuron oscillates strongly when superfused with any of several neuromodulators (Harris-Warrick and Marder, 1991). The characteristics of these oscillations (burst period, duration, amplitude) are different for each modulator. This diversity results at least partly from the fact that bursting in this neuron can be driven by a number of different voltage-dependent currents, a finding that has subsequently been confirmed by modeling studies (Marder, 1997). Each amine activates a different subset of conductances, resulting in different bursting patterns (Harris-Warrick and Marder, 1991).

Similarly, dopamine and serotonin modulate different currents in *Aplysia* neuron R15 to produce superficially similar changes in bursting, in this case the cessation of spontaneous bursting. However, when silenced by serotonin but not by dopamine, a brief depolarizing input provokes sustained bursting. A modeling study of this system showed that by modulating different currents, both amines silence R15, but dopamine prevents other input signals from activating the neuron, whereas serotonin amplifies synaptic inputs (reviewed in Fellous and Linster, 1998).

Modulators also alter the ability of neurons to generate plateau potentials, the regenerative switch between two stable membrane potentials. At one, the cell is hyperpolarized and silent; at the other, it is depolarized and fires action potentials. Shifts between the two levels occur abruptly when the neuron's membrane potential crosses a threshold value in response to current injection or postsynaptic potentials. The abilities of neurons to generate plateau potentials can be enhanced or suppressed by modulatory inputs. In the stomatogastric system, for example, activity in the Anterior Pyloric Modulator neuron enhances or suppresses the abilities of different neurons to generate plateau potentials. One effect of the changes in plateau capability is an altered sensitivity to synaptic inputs. When plateaus are generated, inputs strong enough to trigger the regenerative shift from one level to the other have greater effects than they would otherwise have, whereas those that are too

weak to trigger the shift have little effect (Dickinson and Nagy, 1983). Once the threshold is reached, further increases in synaptic strength have little effect. In network computations, this characteristic effectively decreases the importance of synaptic strength (Marder, 1993). Consequently, the postsynaptic response becomes nonlinear. In addition, the effective duration of synaptic inputs is changed, since, once shifted, the postsynaptic neuron remains at its new level; hence, the effect of inputs sufficient to induce a switch from one plateau level to another long outlasts the stimulus duration (Figure 2).

Changes in the abilities of neurons to generate plateau potentials can have far-reaching effects on network activity. For example, when the plateau properties of the ventricular dilator (VD, a pyloric neuron), are suppressed, the VD no longer fires with the pyloric pattern. Instead, if the much slower cardiac sac pattern is active, the VD fires with this network (reviewed in Harris-Warrick and Marder, 1991).

The roles of different conductances in determining the firing patterns of nonoscillatory neurons have now been examined in the stomatogastric nervous system both in models and in experiments using the dynamic clamp technique. The LP neuron, for example, is modulated by proctolin, and adding the proctolin conductance to the model alters both its activity and other membrane currents in ways that reflect the peptide's biological effects. Modeling has confirmed that even small currents that produce minor changes in membrane potential may have profound effects (Marder, 1993).

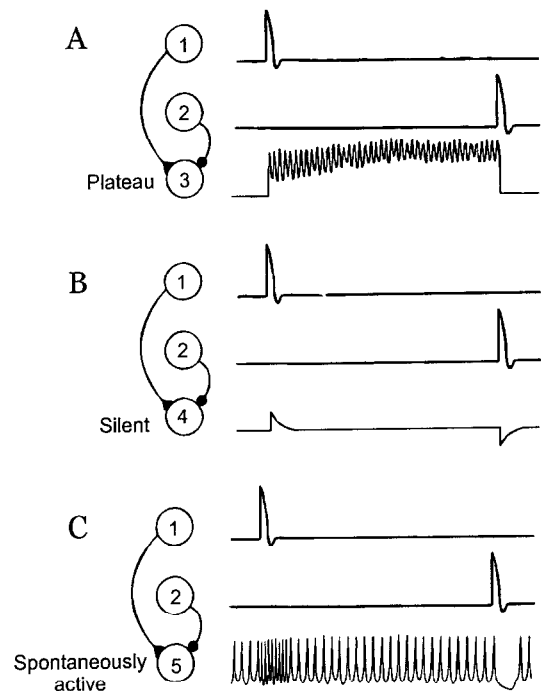


Figure 2. Plateau properties in a postsynaptic neuron increase the duration of its response to synaptic input. *A*, When the follower cell, 3, has plateau properties, an excitatory postsynaptic potential (from 1, triangle) triggers a shift to a depolarized plateau, whereas an inhibitory postsynaptic potential (from 2, circle) terminates the plateau. When the follower cell (4, 5) is silent but does not have plateau properties (*B*) or is spontaneously active (*C*), the effect of the same excitatory or inhibitory postsynaptic potential lasts for only a short time after the input. (Source: Marder, E., 1993, *Modulating membrane properties of neurons: Role in information processing*, in *Exploring Brain Functions: Models in Neuroscience* (T.A. Poggio and D.A. Glasser, Eds.), Chichester, Engl.: Wiley, p. 30. Copyright 1993. Reprinted by permission of John Wiley & Sons, Ltd.)

Similarly, it has been possible to determine which K currents are most important for provoking the changes in postinhibitory rebound, and hence phasing of the pyloric pattern, when dopamine is applied to the stomatogastric system (Harris-Warrick et al., 1998).

An intriguing recent finding is the observation that many different modulators, including the peptides proctolin, RPCH, CabTRP, and TNRNFLRFamide and the muscarinic agonist pilocarpine, activate the same membrane current in neurons of the lobster STG. However, while all these modulators activate the pyloric pattern, each produces a different variant of the pattern, owing to the fact that a different subset of pyloric neurons contains receptors for each modulator (Nusbaum et al., 2001).

When modeling networks subject to neuromodulation, one must consider that although a modulator may act on only a subset of neurons in the network, neurons that are not direct targets of the modulator can likewise be affected and can influence the output of the system through their synapses within the network (for further discussion, see CRUSTACEAN STOMATO-GASTRIC SYSTEM).

Alteration of Synaptic Efficacy by Neuromodulators

The efficacy of both chemical and electrical synapses can be changed by neuromodulator actions. For chemical synapses, changes in the amount of transmitter released or in the responsiveness of postsynaptic neurons can contribute to modulation. In *Aplysia*, for example, changes in synaptic efficacy are largely responsible for the long-term changes that underlie learning (see INVERTEBRATE MODELS OF LEARNING: *APLYSIA* AND *HERMISENDA*). In the stomatogastric system, RPCH increases the efficacy of the synapses from a single presynaptic neuron onto its follower cells; this increase in synaptic efficacy is sufficient to cause a functional rewiring of two pattern generators and to provoke the generation of a novel rhythm (reviewed in Harris-Warrick and Marder, 1991).

Many synapses in invertebrates release transmitter not only in response to action potentials, but also as a graded function of membrane potential. It has recently been found that graded and spike-mediated transmitter release can be differentially modulated. Thus, for example, dopamine enhances graded synaptic transmission but decreases spike-evoked transmission at the LP-PD neuron synapse of the pyloric network (Harris-Warrick et al., 1998).

Electrical synapses are likewise subject to modulation. Serotonin, octopamine, and dopamine alter electrical coupling in the lobster pyloric system, with coupling at some synapses increased while at other synapses it is decreased. Because the same modulators can change the efficacies of both chemical and electrical synapses between the same neurons, the effective sign of a synapse can be changed. Dopamine, for example, alters synaptic efficacy between several pairs of pyloric neurons that are connected by dual synapses. Under control conditions, the electrical component dominates, and the synapses are largely excitatory. Under dopamine, however, the chemical component dominates, and the synapses are largely inhibitory (Harris-Warrick et al., 1998).

Modulation of Neuromuscular Junctions and Muscles

Neuromodulators in many systems exert effects on neuromuscular junctions or on muscles themselves. These effects are often consistent with central or sensory effects of the same modulators. Modulators, which are released both from motor neurons and from exogenous sources, can change the amplitude, duration, or speed of muscle contraction or relaxation. These effects result from changes at one or more of three levels: presynaptic effects resulting in altered transmitter release from motor terminals, electrical properties

and excitability of the muscle fibers themselves, and excitation-contraction coupling (Harris-Warrick and Marder, 1991).

These multiple steps have recently been incorporated into a model of the “neuromuscular transform” (NMT) by Brezina et al. (Brezina, Orekhova, and Weiss, 2000), who used a dynamical systems approach to examine modulation of feeding muscles in *Aplysia*, both theoretically and in this model neuromuscular system. These authors show that the NMT, which acts as a dynamic, nonlinear filter, limits the range of behaviors that can be produced. In particular, they find that many rhythmic behaviors break down at the level of the NMT when the neuronal cycle frequency driving the muscles increases beyond certain limits. However, neuromodulators in *Aplysia* modify a number of characteristics of the neuromuscular system, including the amplitude of contraction and rate of relaxation in response to neuronal stimulation, thereby altering the constraints on the system. Both intrinsic and extrinsic peripheral neuromodulation occur in this system, and the theoretical framework developed by these authors has shown that intrinsic modulation, which varies with the motor pattern itself, optimizes the performance of a single behavior. In contrast, extrinsic modulation, which is independent of the motor pattern, allows multiple contraction shapes to be generated and hence allows multiple behaviors to be produced using the same neuromuscular system.

Discussion

Modulation is prevalent in both invertebrate and vertebrate nervous systems, and it occurs at all levels: sensory, central, and motor. Neuromodulators alter the output of a system by changing membrane properties of neurons, synaptic interactions, and intracellular properties such as excitation-contraction coupling in muscles. Moreover, modulatory inputs can act on time scales ranging from seconds to days. Consequently, the properties of individual neurons, as well as the nature and extent of their interactions within networks, are dynamic rather than static, and so the temporal dynamics of such changing properties must be considered if we are to fully understand and appreciate the flexibility of the nervous system. Moreover, these dynamics must be incorporated into models if they are to fulfill their promise in illuminating principles of neuronal functioning. Additionally, the responses of a given system to a specific neuromodulator are not always the same and may depend on the state of the preparation when the modulator is applied or activated. Because a range of substances modulates invertebrate systems, this variability can have important consequences.

Moreover, because the neural circuits that are subject to modulation differ substantially, the same modulator can cause different effects on different systems. Serotonin, for example, enhances swimming behavior in leeches, *Tritonia*, and the pteropod *Clione*. In both leeches and *Tritonia*, one component of this increase is enhanced presynaptic transmitter release. In *Tritonia* and *Clione*, the excitability of pattern-generating neurons is enhanced. In the lobster STG, serotonin causes an overall activation of the pyloric pattern but an inhibition of a number of neurons within the network, so the circuit is effectively limited to three of the eight neuronal types (Leviton and Kaczmarek, 2002).

The specific effects of a given neuromodulator are determined by numerous factors, including (1) the array of neurons expressing receptors for that modulator, (2) the membrane channels (often voltage-dependent) that are altered by the modulator, (3) the membrane potentials of neurons in the circuit, and (4) the interactions of those neurons within the network.

At least partly because of the dynamic nature and complexity brought to nervous systems by modulation, models are being successfully used to test fundamental assumptions underlying mechanisms of neuronal and network function and modulation. Modeling studies have, for example, confirmed that neuromodulators

are able to increase and control the computational complexity of networks without increasing their structural complexity. However, it is important to consider neuromodulation as an integral part of models rather than simply as an "add-on" (Fellous and Linster, 1998). Moreover, models will ultimately need to incorporate the more complex aspects of neuromodulation that are now being found experimentally, including, for example, interactions amongst modulators, and the actions of modulators at multiple levels.

Road Map: Biological Networks

Related Reading: Crustacean Stomatogastric System; Neuromodulation in Mammalian Nervous System

References

- Brezina, V., Orekhova, I., and Weiss, K., 2000, Optimization of rhythmic behaviors by modulation of the neuromuscular transform, *J. Neurophysiol.*, 83:260–279.
- Clarac, C., Cattaert, D., and LeRay, D., 2000, Central control components of a "simple" stretch reflex, *Trends Neurosci.*, 23:199–208. ♦
- Dickinson, P. S., Fairfield, W. P., Hetling, J. R., and Hauptman, J., 1997, Neurotransmitter interactions in the stomatogastric system of the spiny lobster: One peptide alters the response of a central pattern generator to a second peptide, *J. Neurophysiol.*, 77:599–610.
- Dickinson, P. S., and Nagy, F., 1983, Control of a central pattern generator by an identified modulatory interneurone in Crustacea: II. Induction and modification of plateau properties in pyloric neurones, *J. Exp. Biol.*, 105:59–82.
- Fellous, J.-M., and Linster, C., 1998, Computational models of neuromodulation, *Neural Comput.*, 10:771–805. ♦
- Frost, W. N., Lieb, J., Jr., Tunstall, M. J., Mensh, B. D., and Katz, P. S., 1997, Integrate-and-fire simulations of two molluscan neural circuits, in *Neurons, Networks, and Motor Behavior* (P. G. Stein, S. Grillner, A. I. Selverston, and D. Stuart, Eds.), Cambridge, MA: MIT Press, pp. 173–179.
- Harris-Warrick, R. M., Johnson, B. R., Peck, J. H., Kloppenburg, P., Ayali, A., and Skarbinski, J., 1998, Distributed effects of dopamine modulation in the crustacean pyloric network, *Ann. N.Y. Acad. Sci.*, 860:155–167. ♦
- Harris-Warrick, R. M., and Marder, E., 1991, Modulation of neural networks for behavior, *Annu. Rev. Neurosci.*, 14:39–57. ♦
- Katz, P. S., 1998, Comparison of extrinsic and intrinsic neuromodulation in two central pattern generator circuits in invertebrates, *Exp. Physiol.*, 83:281–292. ♦
- Leviton, I., and Kaczmarek, L., 2002, *The Neuron: Cell and Molecular Biology*, Oxford, Engl.: Oxford University Press. ♦
- Marder, E., 1993, Modulating membrane properties of neurons: Role in information processing, in *Exploring Brain Functions: Models in Neuroscience* (T. A. Poggio and D. A. Glaser, Eds.), New York: John Wiley, pp. 27–42. ♦
- Marder, E., 1997, Computational dynamics in rhythmic neural circuits, *The Neuroscientist*, 3:295–302. ♦
- Nusbaum, M. P., Blitz, D. M., Swensen, A. M., Wood, D., and Marder, E., 2001, The roles of co-transmission in neural network modulation, *Trends Neurosci.*, 24:146–154. ♦
- Pasztor, V. M., 1989, Modulation of sensitivity in invertebrate sensory receptors, *Semin. Neurosci.*, 1:5–14. ♦
- Whim, M. D., Church, P. J., and Lloyd, P. E., 1994, Functional roles of peptide cotransmitters at neuromuscular synapses in *Aplysia*, *Molec. Neurobiol.*, 7:335–347. ♦

Neuromodulation in Mammalian Nervous Systems

Michael E. Hasselmo, Bradley P. Wyble, and Erik Fransen

Introduction

Neuromodulators change the way in which neural circuits process information. The term *neuromodulation* usually refers to the effect of neurochemicals such as acetylcholine (ACh), dopamine, norepinephrine, and serotonin, and other substances, including neuropeptides (see Hasselmo, 1995; Fellous and Linster, 1998; Katz, 1999; Doya, Dayan, and Hasselmo, 2002). The term neuromodulation does not refer to the rapid transmission of information through the nervous system by excitatory and inhibitory synaptic potentials. Rapid synaptic potentials are caused by neurotransmitters such as glutamate or γ -aminobutyric acid (GABA) acting on receptor proteins containing ion channels (ionotropic receptors), which cause fast changes in the conductance of the cell membrane to specific ions. In contrast, the neuromodulators primarily activate receptor proteins that do not contain an ion channel (metabotropic receptors). These receptors activate enzymes that change the internal concentration of substances called second messengers. Second messengers cause slower and longer-lasting changes in the physiological properties of neurons, resulting in changes in the processing characteristics of the neural circuit.

The effect of a neurochemical is receptor dependent (Table 1). Thus, a single neuromodulator such as serotonin can have dramatically different effects on different neurons, depending on the type of receptor it activates. Even the distinction between neurotransmitters and neuromodulators has exceptions based on receptor effects: glutamate and GABA can activate slower metabotropic receptor subtypes (mGluR and GABA_B), whereas some receptors for ACh and serotonin are ionotropic (nicotinic ACh and 5-HT₃).

Neural network models are important for understanding the function of neuromodulatory influences, because neuromodulation causes effects that may appear subtle and contradictory in recordings from single neurons but have a significant effect on dynamical properties when distributed throughout a network (Hasselmo, 1995; Fellous and Linster, 1998; Katz, 1999; Doya et al., 2002). The anatomical distribution of fibers releasing neuromodulatory substances in the brain is usually very diffuse, as shown in Figure 1. This allows the activity of a small number of neuromodulatory neurons to influence the functional properties of broad regions of the brain. Neuromodulatory effects are usually slower than effects at ionotropic receptors, causing longer-term changes in functional state.

This review falls into two main sections, the cellular effects of neuromodulators and the functional modeling of neuromodulation. The first section summarizes some of the physiological effects of neuromodulation, including effects on (1) the resting membrane potential of pyramidal cells and interneurons, (2) spike frequency adaptation, (3) synaptic transmission, and (4) long-term potentiation. The second section reviews several different theories of the function of modulatory influences in neural circuits, including (1) noradrenergic modulation of attentional processes, (2) dopaminergic modulation of working memory, (3) cholinergic modulation of input versus internal processing, and (4) modulation of oscillatory dynamics in cortex and thalamus.

Modeling Cellular Effects of Neuromodulation

A range of different neuromodulatory effects have been modeled in compartmental simulations of cortical pyramidal cells, including

Table 1. Some Receptor-Dependent Effects of a Subset of Neuromodulatory Substances

Neuromodulator	Receptor Subtype	Resting Potential		Spike Adaptation	Synaptic Transmission			Long-Term Potentiation
		Pyramidal Cell	Interneuron		Inhibition	Afferent Input	Feedback	
Acetylcholine	Muscarinic	↑	↑	↓	↓		↓	↑
	Nicotinic	↑	↑			↑		
Dopamine	D ₁		↑		↑		↑	
	D ₂				↓			
Norepinephrine	α				↓		↓	
	β		↑	↓			↑	↑
Serotonin	5-HT ₁	↓	↓					
	5-HT ₂			↓				
	5-HT ₃	↑	↑					
Opioids	μ		↓					

changes in resting membrane potential, spike frequency adaptation, synaptic transmission, and long-term potentiation. Many of these effects are described in more detail in reviews of neuromodulation (McCormick, Wang, and Huguenard, 1993; Hasselmo, 1995; Fellous and Linster, 1998; Katz, 1999).

Pyramidal Cell Membrane Potential

Some neuromodulatory agents cause slow changes in resting membrane potential owing to changes in the resting membrane conductance to individual ions. For example, activation of muscarinic ACh receptors causes a slow depolarization of pyramidal cells by decreasing the leak potassium conductance. In contrast, GABA activation of GABA_B receptors and serotonin activation of 5-HT_{1A} receptors causes a slow hyperpolarization by increasing potassium conductance. The time course of this GABA_B effect is modeled in SYNAPTIC INTERACTIONS (q.v.).

Inhibitory Interneuron Membrane Potential

Many neuromodulators have a strong effect on the membrane potential of inhibitory interneurons. Depolarization of interneurons as a result of activation of dopaminergic, noradrenergic, cholinergic, and serotonergic receptors has been detected, whereas opioid receptors have a strong hyperpolarizing effect on interneurons.

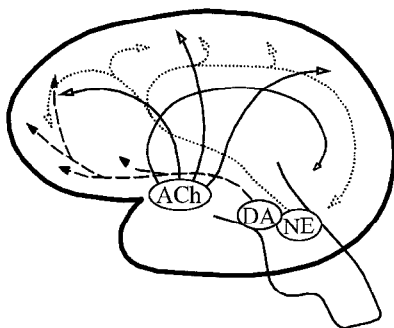


Figure 1. Example of the anatomy of neuromodulatory systems. Neurons producing acetylcholine (ACh) are clustered in nuclei of the basal forebrain. This relatively small population of neurons sends diffuse projections to a broad range of brain regions. Similarly, noradrenergic neurons (NE) are clustered in the locus coeruleus, and send diffuse connections throughout the brain. Dopaminergic neurons (DA) are in the ventral tegmental area and substantia nigra and project to nucleus accumbens, striatum, and frontal cortex.

Spike Frequency Adaptation

When injected with current, cortical pyramidal cells fire action potentials initially at high frequencies, but neurons show a rapid decrease in firing frequency, referred to as *accommodation* or *adaptation*. ACh, norepinephrine, and serotonin reduce spike frequency adaptation, allowing higher frequency firing. In models, spike frequency adaptation can be modeled with an increase in intracellular calcium, because of the voltage-dependent calcium influx caused by each action potential. The increasing intracellular calcium concentration activates calcium-dependent potassium currents (also known as $I_{K(AHP)}$), which are reduced by the above neuromodulators.

Synaptic Transmission

Many neuromodulatory substances activate presynaptic receptors, which alter the release of neurochemicals, including glutamate and GABA. Receptors that cause presynaptic inhibition at glutamatergic synapses include the α-adrenergic receptors (which are a subtype of norepinephrine receptors), muscarinic ACh receptors, GABA_B receptors, neuropeptide Y receptors, and adenosine receptors. As shown in Table 1, these effects can be selective for specific synapse types.

Long-Term Potentiation

Neuromodulators have been shown to induce long-term potentiation and to enhance long-term potentiation induced by synaptic stimulation. β-adrenergic receptors and muscarinic receptors appear to actively induce long-term potentiation at specific synaptic connections. Activation of dopamine receptors, muscarinic ACh receptors, metabotropic glutamate receptors, opioid receptors, and GABA_B receptors appears to influence the magnitude of long-term potentiation caused by synaptic stimulation.

Functional Modeling of Neuromodulation

The cellular effects of neuromodulators can dramatically alter the functional dynamics of neural circuits, in contradiction to the common notion that neuromodulation only slightly increases or decreases the normal function of the network (Hasselmo, 1995; Fellous and Linster, 1998; Katz, 1999). The essential functional role of neuromodulation is illustrated by the breakdown of normal cognitive function by high doses of drugs such as scopolamine (which blocks ACh receptors) and by the important clinical effects of selective serotonin reuptake inhibitors (SSRIs).

Modulator effects are often described in simple colloquial terms such as “memory,” “attention,” or “reward,” but the techniques of

computational neuroscience allow a more sophisticated assessment of the specific functional influence of modulators on neural circuits (Hasselmo, 1995; Fellous and Linster, 1998; Doya et al., 2002). This will allow data from specific behavioral tasks to be addressed directly in terms of the dynamics of neural circuits rather than as simple verbal hypotheses.

Norepinephrine

Drugs that enhance the release of norepinephrine and other monoamines have been shown to enhance performance on tests of sustained attention, such as the continuous performance task, in which subjects must detect an infrequent target stimulus in a long series of distractor stimuli. In recordings from monkeys, the enhancement of performance on these tasks has been linked to activity of noradrenergic neurons. The role of norepinephrine in this regard has been modeled using a hybrid model with detailed representation of the spiking activity of noradrenergic neurons in the locus coeruleus, coupled with a more abstract representation of cortical circuits performing the visual detection task (Usher et al., 1999). The cortical model regulates the behavioral response to individual stimuli using competing processing units with sigmoid input-output functions. The gain (slope) of these sigmoid units is increased by noradrenergic modulation. In this model, noradrenergic neurons have a low baseline firing rate, ensuring low gain in the cortical network, and a low false alarm rate. However, these neurons show a brief response to the target stimulus, which transiently increases the gain of the cortical units, increasing the likelihood of generating a response. The phasic level of noradrenergic neuron activity correlates with greater performance accuracy in the model and in experiments (Usher et al., 1999). A tonic increase in noradrenergic activity enhances the response to all stimuli, consistent with general arousal, which allows orienting to all stimuli but decreases visual detection performance.

The change in input-output functions described above could be a reasonable representation of the changes in circuit dynamics caused by norepinephrine. On a cellular level, norepinephrine activates β -adrenergic receptors, which decrease spike frequency adaptation and enhance postsynaptic responses (see Table 1). At the same time, norepinephrine activates presynaptic α -adrenergic receptors, which suppress excitatory transmission. In network simulations of piriform cortex, these effects enhance the response of pyramidal cells to afferent input while decreasing the background activity caused by excitatory transmission between pyramidal cells in the cortex (Hasselmo et al., 1997). Norepinephrine has the apparently contradictory effects of decreasing excitatory synaptic input to interneurons while directly depolarizing these same interneurons. These effects decrease the activity of local circuits in response to weak input but increase their activity in response to strong input, an effect that resembles the change in gain of sigmoid input-output functions utilized in the model described above.

Dopamine

Models of dopamine function include research on reinforcement learning as well as working memory. Dopamine has traditionally been viewed as a neuromodulator signaling reinforcement, but electrophysiological recording of the activity of dopaminergic neurons in the ventral tegmental area demonstrates activity dependent on the expectation of reward rather than on the reward itself. This pattern of activity appears similar to the error signal used in temporal difference learning. Models of the role of dopamine in reinforcement learning are described in more detail in REINFORCEMENT LEARNING and DOPAMINE, ROLES OF.

Experimental data also indicate a role for dopamine in working memory (the capacity to hold information in short-term memory

for performance of tasks). Working memory may involve sustained spiking activity in populations of neurons within the prefrontal cortex. This sustained spiking activity can be maintained by excitatory recurrent connectivity or by the intrinsic properties of individual neurons. Detailed biophysical modeling demonstrates how dopaminergic modulation could contribute to both the transition between different activity states, as well as the maintenance of stable spiking activity (Durstewitz, Seamans, and Sejnowski, 2000). In this model, dopamine enhances the maintenance of individual stored items through enhancement of persistent voltage-sensitive sodium currents and the NMDA current. At the same time, dopamine reduces AMPA currents and depolarizes inhibitory interneurons through activation of the D_1 subtype of receptor. In the model, this serves to prevent activation of other task-irrelevant memories. Other work from that group showed that the initial response to dopaminergic modulation involved a D_2 receptor-mediated suppression of inhibitory synaptic transmission, which could serve to allow activation of prefrontal cortex by multiple inputs, and exploration of the input space. This effect is transient, while the more persistent effects at D_1 receptors could allow selection and maintenance of a single memory for a more extended period.

Acetylcholine

The cellular effects of ACh play an important role in the function of cortical networks (Hasselmo, 1995). For example, muscarinic cholinergic antagonists such as scopolamine impair the encoding of new words but not the retrieval of previously encoded words. Models demonstrate how ACh could be important for enhancing encoding of new information in the cortex. As shown in Figure 2, computational modeling demonstrates how cholinergic activation of a calcium-sensitive cation current in layer II pyramidal cells in entorhinal cortex could cause sustained spiking activity such as that observed during the delay period of a delayed non-match-to-sample task (Fransen, Alonso, and Hasselmo, 2002). These mechanisms could allow maintenance of novel information for encoding and working memory.

Computational modeling also demonstrates how the cellular effects of ACh can enhance the encoding of new information patterns by selectively enhancing the response to external sensory stimuli

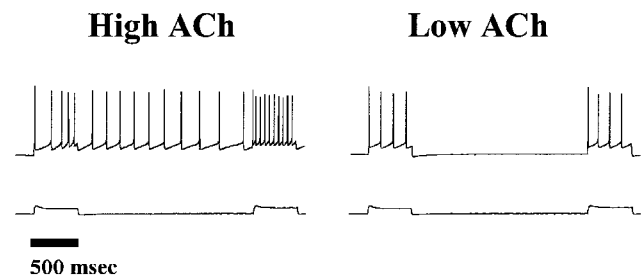


Figure 2. A biophysical simulation of an entorhinal cortex layer II pyramidal cell demonstrates that cholinergic enhancement of a calcium-sensitive cation current could allow maintenance of sustained spiking without afferent input. Bottom traces show the timing of depolarizing input, which causes spiking for 500-ms periods separated by a 2,500-ms delay. When cholinergic modulation is present (in a condition of high ACh, left side), the calcium influx during initial spiking causes activation of the cation current. This causes further depolarization, which causes additional spiking and calcium influx. This causes regenerative spiking activity even in the absence of external input. With lower ACh levels (right side), the current is not sufficiently activated and the neuron does not fire during the delay period.

versus internal retrieval (Hasselmo, 1995). This results from cholinergic suppression of glutamatergic transmission at feedback synapses but not at feedforward synapses, coupled with cellular depolarization, which makes neurons more likely to spike in response to afferent input. This causes activity to be clamped to the pattern of afferent input, allowing accurate storage of input patterns without interference from retrieval of previously stored patterns. Cholinergic enhancement of long-term potentiation offsets the reduction of synaptic input at excitatory intrinsic connections, allowing Hebbian synaptic modification of these connections for autoassociative storage of input patterns. On a faster time scale, similar transitions could occur in different phases of each cycle of hippocampal theta rhythm oscillations (Hasselmo, Bodelon, and Wyble, 2002). Theta rhythm appears when an animal is actively exploring the environment or attending to relevant stimuli. The phase with strong afferent input could allow encoding of new sequences without interference from retrieval, while a separate phase of dominant feedback connections could allow retrieval. Loss of this oscillatory modulation could underlie the learning impairments caused by fornix lesions, which damage the modulatory influences pacing theta rhythm.

Low ACh levels allow strong feedback transmission for consolidation. During quiet waking and slow-wave sleep, there is a dramatic decrease in ACh level, which is accompanied by the appearance of sharp wave events in the EEG (Buzsaki, 1989). During sharp waves, associations encoded in hippocampus are theorized to be reactivated, causing activity in hippocampus and neocortex that could mediate the formation of additional traces of the same memory (Buzsaki, 1989; Shen and McNaughton, 1996). The generation of sharp waves becomes more robust and drives neocortical activity more strongly when low levels of ACh release the suppression of excitatory feedback connections, allowing a shift from tonic theta rhythm oscillations to bursts of activity due to excitatory positive feedback.

Modulation and Network Oscillations

Changes in modulatory levels play an important role in setting the oscillatory properties of the EEG and functional properties important for behavior. Microdialysis of neuromodulators during different stages of waking and sleep show dramatic changes in their levels associated with changes in the cortical EEG. High levels of ACh, NE, and 5-HT are present during waking, very low levels of ACh are present during slow-wave sleep, and very low levels of NE and 5-HT coupled with high levels of ACh are present during REM sleep. Computational models have illustrated potential mechanisms for these modulatory influences on cortical dynamics.

Neuromodulation alters the way in which thalamic circuitry regulates low-frequency synchronous oscillations that appear in the neocortex during sleep, including spindle activity and delta waves. Modeling demonstrates that the fundamental frequency of spindle activity could result from the interaction of intrinsic currents in thalamic reticular neurons, which fire bursts due to a low-threshold calcium current and hyperpolarize due to a calcium-activated potassium current. These cells provide rhythmic inhibitory input to thalamocortical relay cells, which then repolarize because of the hyperpolarization-activated nonspecific cation current and fire bursts because of the activation of the low-threshold calcium current (Terman, Bose, and Kopell, 1996; also see SYNAPTIC INTERACTIONS). Cholinergic innervation of thalamic circuits could prevent the generation of spindle and delta wave oscillations by depolarizing neurons through blockade of potassium currents, thereby inactivating the low-threshold calcium current underlying bursting (McCormick et al., 1993). Decreases in cholinergic depolarization would initially cause spindles when both thalamo-

cortical neurons and thalamic reticular neurons are active, but as reticular neurons become less depolarized, their activity decreases, and thalamocortical neurons can oscillate synchronously at delta wave frequencies (Terman et al., 1996).

Neuromodulation plays an essential role in generating the hippocampal theta rhythm (see HIPPOCAMPAL RHYTHM GENERATION). Theta rhythm is primarily paced by rhythmic input from the medial septum, but it can also arise from intrinsic mechanisms in local circuits. In slice preparations of the hippocampus, cholinergic modulation has been demonstrated to induce theta rhythm oscillations (Tiesinga et al., 2001). Simulations demonstrate that the appearance of oscillations in the slice could result from slow depolarization of pyramidal cells, causing spiking activity, which is then synchronized by excitatory synaptic connections, with a time course dependent on the calcium-activated potassium current (Traub, Miles, and Buzsaki, 1992; Tiesinga et al., 2001). Computational modeling provides a means for understanding the role of neuromodulation in regulating a wide range of oscillatory dynamics.

Discussion

Neuromodulation plays an important role in brain function, as demonstrated by the dramatic behavioral effects of drugs that influence the release of neuromodulators or activate receptors for neuromodulators. Neuromodulators do not just cause slight quantitative changes in network function but can qualitatively alter neural circuits to a completely different functional state. Computational modeling helps us understand the functional role of modulatory effects that may appear contradictory at a cellular level (for example, many modulators simultaneously depolarize neurons while suppressing synaptic transmission). Computational modeling will prove essential to understanding the functional role of diffuse modulatory effects with complex effects on multiple components of neural circuits.

Road Map: Biological Networks

Related Reading: Dopamine, Roles of; Neuromodulation in Invertebrate Nervous Systems

References

- Buzsaki, G., 1989, Two-stage model of memory trace formation: A role for "noisy" brain states, *Neuroscience*, 31:551–570.
- Doya, K., Dayan, P., and Hasselmo, M. E., Eds., 2002, Special issue on neuromodulation, *Neural Netw.*, 15.
- Durstewitz, D., Seamans, J. K., and Sejnowski, T. J., 2000, Dopamine-mediated stabilization of delay-period activity in a network model of prefrontal cortex, *J. Neurophysiol.*, 83:1733–175.
- Fellous, J. M., and Linster, C., 1998, Computational models of neuromodulation, *Neural Computat.*, 10:771–805. ♦
- Fransen, E., Alonso, A. A., and Hasselmo, M. E., 2002, Simulations of the role of the muscarinic-activated calcium-sensitive non-specific cation current I(NCM) in entorhinal neuronal activity during delayed matching tasks, *J. Neurosci.*, 22:1081–1097.
- Hasselmo, M. E., 1995, Neuromodulation and cortical function: Modeling the physiological basis of behavior, *Behav. Brain Res.*, 67:1–27.
- Hasselmo, M. E., Bodelon, C., and Wyble, B. P., 2002, A proposed function for hippocampal theta rhythm: Separation of encoding and retrieval enhances reversal of prior learning, *Neural Computat.*, 14:793–817.
- Hasselmo, M. E., Linster, C., Ma, D., and Cekic, M., 1997, Noradrenergic suppression of synaptic transmission may influence cortical "signal-to-noise" ratio, *J. Neurophysiol.*, 77:3326–3339.
- Katz, P. S., 1999, *Beyond Neurotransmission: Neuromodulation and Its Importance for Information Processing*, New York: Oxford University Press. ♦

- McCormick, D. A., Wang, Z., and Huguenard, J., 1993, Neurotransmitter control of neocortical neuronal activity and excitability, *Cerebr. Cortex*, 3:387–398.
- Shen, B., and McNaughton, B. L., 1996, Modeling the spontaneous reactivation of experience-specific hippocampal cell assemblies during sleep, *Hippocampus*, 6:685–692.
- Terman, D., Bose, A., and Kopell, N., 1996, Functional reorganization in thalamocortical networks: Transition between spindling and delta sleep rhythms, *Proc. Natl. Acad. Sci. USA*, 93:15417–15422.

- Tiesinga, P. H., Fellous, J. M., Jose, J. V., and Sejnowski, T. J., 2001, Computational model of carbachol-induced delta, theta, and gamma oscillations in the hippocampus, *Hippocampus*, 11:251–274.
- Traub, R. D., Miles, R., and Buzsaki, G., 1992, Computer simulation of carbachol-driven rhythmic population oscillations in the CA3 region of the in vitro rat hippocampus, *J. Physiol. (Lond.)*, 451:653–672.
- Usher, M., Cohen, J. D., Servan-Schreiber, D., Rajkowski, J., and Aston-Jones, G., 1999, The role of locus coeruleus in the regulation of cognitive performance, *Science*, 283:549–554.

Neuromorphic VLSI Circuits and Systems

Stephen P. DeWeerth and Andreas G. Andreou

Introduction

Biological systems excel at sensory perception, motor control, and sensorimotor coordination by sustaining high computational throughput with minimal energy consumption. Research in electronic neuromorphic engineering (Mead, 1989) has had two intertwined objectives. The first objective is the use of very large-scale integrated (VLSI) technology as a modeling tool aimed at capturing the behavior of living neurons, networks of neurons, and the complex mechanical-electrical-chemical information processing present in biological systems. The second objective—the subject of this article—is the development of engineered systems based on abstractions of sensory, motor, and brain function.

Neuromorphic VLSI systems employ distributed and parallel representations and computation akin to those found in their biological counterparts. The hardware implementations combine analog and digital circuits to realize a variety of computational primitives and architectures. Unlike traditional analog computing, neuromorphic representations are not linearized a priori, but rather exploit the inherent nonlinearities and dynamics that arise from the physics of the devices and circuits. The bio-inspired approach to the engineering of VLSI microsystems results in the embodiment of computation in complex, *physical* systems that exploit biological inspiration and lie beyond digital computing and that have a wide range of industrial applications (Vittoz, 2002).

The high levels of system integration offered in VLSI technology make it attractive for the implementation of highly complex artificial neuronal systems, even though the physics of the liquid-crystalline state of biological structures is different from the physics of the solid-state silicon technologies. Silicon complementary metal oxide semiconductor (CMOS) transistors are used in their subthreshold region of operation (Vittoz, 1985; Mead, 1989) because in this regime the primary constraint of highly integrated microsystems, power consumption, is all but eliminated. Circuits operating in their subthreshold region also exhibit a diverse set of computational primitives that are continuous, analog functions of time, space, voltage, current, and charge.

In this article, we discuss neuromorphic VLSI viewed at three hierarchical levels: the *device* level, the *circuit/network* level, and the *systems* level. Given the space limitations, we focus primarily on the circuit/network level. We describe a design style that employs mixed-signal (analog/digital), current-mode CMOS circuits that have minimal complexity (Andreou et al., 1991). Our presentation is not a comprehensive overview of the field, even within the bounds of circuits and networks. Rather, it provides a basic foundation in device physics and presents a set of specific circuits that implement certain essential functions that exemplify the breadth possible within this design paradigm.

Devices

Neuromorphic VLSI circuits and systems are designed with two basic device building blocks available in CMOS technology—transistors and capacitors.

The transistors used in most neuromorphic implementations are devices operating in their *subthreshold* regime. In this regime, the current through the device is an *exact difference* of exponential functions of the drain and source voltages (Vittoz, 1985; Mead, 1989; Andreou et al., 1991). For an *n*MOS transistor, the current is given by

$$I_{DS} = I_{n0} \cdot S \cdot e^{\kappa_n V_{GB}} (e^{-V_{SB}} - e^{-V_{DB}}) \quad (1)$$

and for a *p*MOS transistor by

$$I_{SD} = I_{p0} \cdot S \cdot e^{-\kappa_p V_{GB}} (e^{V_{SB}} - e^{V_{DB}}) \quad (2)$$

The terminal voltages V_{GB} , V_{SB} , and V_{DB} , in these equations are normalized to the thermal voltage $U_t = kT/q$, which is approximately equal to 25 mV at room temperature. The constants I_{n0} and I_{p0} depend on the mobility of the carriers (electrons and holes) and other physical properties of the silicon, and are typically in the range of 10^{-15} A. The geometry factor, $S = W/L$, where W and L are the width and length of the device, respectively. The constants κ_n and κ_p take values between 0.6 and 0.9. All of these parameters are fixed by the fabrication process, and thus cannot be modified by the designer or user with the exception of the variable S , which can be modified by designing the geometry of the device, and U_t , which is a function of operational temperature.

The MOS transistor has excellent circuit properties as a voltage-input, current-output device (a *transconductance amplifier*) with good fan-out capabilities (high transconductance, $g_m = \partial I_{DS} / \partial V_{GS}$) and good fan-in capability (extremely high input impedance). Additionally, the exponential voltage-current relationships depicted in Equations 1 and 2 facilitate a powerful synthesis (and analysis) procedure, the *translinear principle* that has also been generalized for MOS circuits (Andreou and Boahen, 1996; Gilbert, 1996).

In addition to CMOS transistors, neuromorphic VLSI systems also utilize capacitors extensively. These capacitors, which are implemented as parallel-plate devices between fabrication layers (e.g., vertically adjacent layers of polysilicon), endow the systems with temporal properties based on the capacitor's ability to store charge (energy). The capacitor equation

$$I = C \frac{dV}{dt} \quad V = \frac{1}{C} \int I dt \quad (3)$$

demonstrates that the capacitor integrates input current and represents this integrated value as a voltage.

Circuits and Networks

The synthesis of computational structures begins at the circuit level and manifests itself as the emergence of *networks*. At the circuit level, conservation laws—the conservation of charge (Kirchoff's current law), $\sum_n I_n = 0$, and the conservation of energy (Kirchoff's voltage law), $\sum_n V_n = 0$ —are used to realize simple constraint equations. These laws, combined with the integration of charge on capacitors, provide a means of implementing functions of both space and time.

The important concept of *negative feedback* is also exploited to implement the inverse of natural functions in the technology and to trade off the gain in the active elements for precision and speed in the circuits. The simplest circuit that exploits the high-gain transconductance of the MOS transistor, its exponential characteristics, and negative feedback is the *diode-connected* transistor. This circuit uses negative feedback to invert the exponential characteristic of the transistor to create a logarithmic current-in, voltage-out configuration.

Current replication and scaling are two additional operations that are used in the implementation of many systems. Current replication is implemented using the *current mirror*, which uses an input transistor to convert an input current logarithmically to a voltage. This voltage is distributed to one or more output transistors, each of which converts this voltage exponentially back into a current that is equal (modulo fabrication offsets and second-order effects) to the input current. Scaling can be accomplished in multiple ways, including transistor sizing and using an additional parameter to control the source of a transistor.

In the remainder of this circuits section, we focus on three computational functions in space and time that are essential to and exemplify the breadth of neuromorphic microsystems: (1) signal aggregation, (2) normalization, and (3) signal quantization. These particular choices certainly do not represent a complete itemization of all of the important neuromorphic circuits/networks, but rather are a representative list. For each of these functions, we discuss both spatial and temporal implementations, and in some cases show circuit schematics and describe the corresponding operation. Finally, we present the issue of the representation and communication of signals among subsystems, which is essential in the implementation of large-scale systems. Throughout this section, we discuss systems that utilize these circuits, networks, and representations as fundamental elements in their implementation.

Signal Aggregation

Signal aggregation is the collection and processing of signals over space and/or time. Spatial aggregation can take many forms, including simple summation, computation of global attributes, and localized aggregation. Temporal aggregation is typically formulated as the integration of charge over time on a capacitor, and includes many temporal filtering functions. The representation of analog signals as currents facilitates the elegant implementation of both spatial and temporal aggregation as a direct result of Kirchoff's current law; thus, aggregation is implemented primarily using currents.

Spatial aggregation. One particularly useful circuit for local spatial aggregation is the resistive network (Mead, 1989, chap. 6). It performs linear addition of signals over a confined region of space, such as that observed throughout the nervous system. The basic resistive network depicted in Figure 1A employs voltages and currents. Its node equation is

$$I_j = G \cdot (V_i + V_k - 2V_j) \cong G \nabla^2 V \quad (4)$$

Note that the term $V_i + V_k - 2V_j$ is a first-order approximation to the Laplacian operator $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$, with the internode distance normalized to unity.

The resistive network in Figure 1A, although simple, is not amenable to compact VLSI integration because conductances G with a large linear range typically consume large amounts of both area and power. However, MOS transistors provide a natural way to exploit the underlying device physics to implement analog VLSI aggregation networks that can be applied to systems, including silicon retinas (Andreou et al., 1995; Vittoz, 2002).

The exponential functions of V_{SB} and V_{DB} in Equations 1 and 2 correspond to Boltzmann-distributed charges at the source and drain diffusing through the channel. For the n MOS transistor, the exponentials can be conveniently represented as dimensionless quantities of charge

$$Q_S \equiv e^{-V_{SB}} \quad Q_D \equiv e^{-V_{DB}} \quad (5)$$

and diffusivity

$$D \equiv S \cdot e^{V_{GB}} \quad (6)$$

so that the Equation 1 becomes

$$I_{DS} = I_{n0} \cdot D^K \cdot (Q_S - Q_D) \quad (7)$$

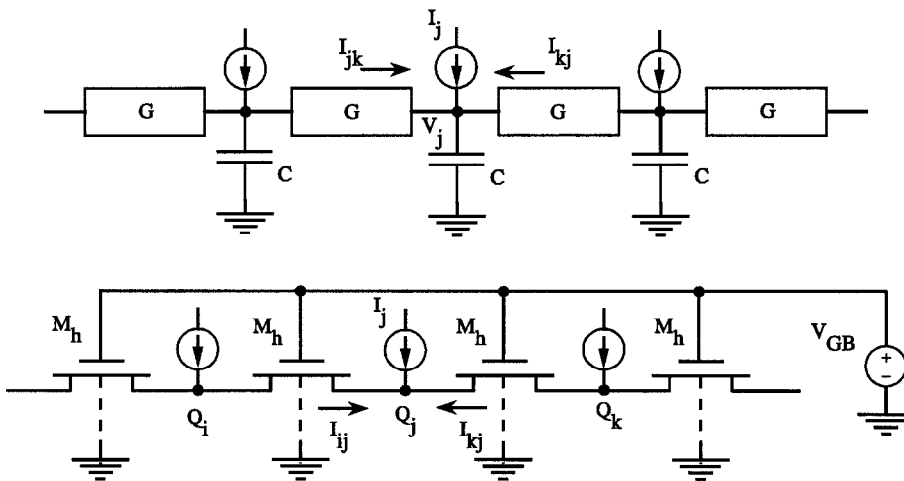


Figure 1. A, Signal aggregation in space using a linear resistive network. B, A network of non-linear conductances implemented using MOS transistors.

The charge-based representation depicted in Equation 10 suggests that the MOS transistor in subthreshold is a highly linear device in the charge domain. This property can be used to implement the resistive network using MOS transistors, as shown in Figure 1B. This network employs charges (positive) and currents through n MOS transistors operating in their subthreshold regime. The node equation is

$$I_j = I_{n0} \cdot D \cdot (Q_i + Q_k - 2Q_j) \cong I_{n0} D \nabla^2 Q \quad (8)$$

assuming that S and V_{GB} are identical for all transistors in the network. The diffusivity D can be controlled by setting the voltage at the gate of transistor M_h at the desired value.

Temporal integration. The aggregation of signals in time is performed using temporal integration circuits that combine a device or circuit that generates a current with a capacitor that integrates that current. The diode capacitor integrator shown in Figure 2 is one of the simplest and most useful of these circuits. The input to this circuit can take the form of a continuously varying current or of a pulse stream generated by quantization circuits such as those described in a subsequent section.

Given subthreshold bias voltages, the input current, I_1 , and output current, I_2 , are related as follows:

$$Q \frac{dI_2}{dt} = I_2 \cdot \left(I_1 - \frac{I_2}{A} \right) \quad (9)$$

where $Q = CU_f/\kappa$ and $A = e^{V_E/U_T}$. This equation demonstrates that the time constant of the circuit can be modified separately from the input current by changing V_E .

When a steady-state train of pulses of width w and frequency $f = 1/T$ is applied to this integrator, the nonlinear behavior of the diode capacitor integrator has the following interesting properties: (1) the steady-state current is proportional to pulse-stream frequency f , (2) the steady-state current is proportional to pulse width w , (3) the steady-state current ripple (error) is independent of current level, and (4) the steady state is reached, for a given precision, after a constant number of pulses.

The diode capacitor integrator has been used in many systems, including silicon retinas that compute motion (Sarpeshkar et al., 1996) and the receiver sections for multichip neuromorphic systems (Boahen, 2000).

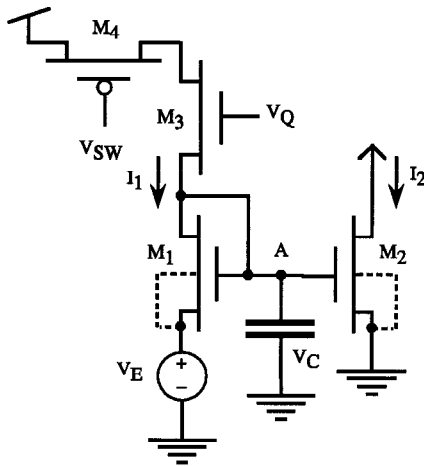


Figure 2. Diode capacitor integrator. The input and output signals are currents. The circuit is basically a current mirror with scaling, with the integrating capacitor at node A providing a time history of the input signals.

Gain Control

Gain control is an important function in biological systems. Many biological signals, especially those in the sensory periphery (e.g., visual and auditory signals), exhibit orders of magnitude of dynamic range. In order to process these signals, it is necessary to modify the dynamic range while not diminishing the information embedded in the signal. This type of normalization occurs in both space and time. Spatial normalization is used to bring a spatially distributed set of signals into a common or prescribed range that is matched to further processing. Temporal normalization typically takes the form of adaptation in which a slow variable that sets the gain of a circuit or network is modified over time to adapt to changes in signal magnitude.

Spatial normalization. Spatial normalization can be divided into *global* and *local* operations. Global normalization modifies the gain of all of the signals in a network, serving to shift the global level of these signals. Local normalization serves to perform similar processing but controls gain based on the activity in a local region.

One example of global normalization is the linear, current-mode normalizing circuit (Gilbert, 1996). Using the translinear principle, it can be shown that the output currents of this circuit are related to its input currents through the following expression:

$$I_{On} = \frac{I_T}{\sum I_{In}} \cdot I_{In} \quad (10)$$

Thus an output current I_{On} in the array is proportional to the corresponding input current I_{In} normalized to the sum of all input currents. The current I_T is a scaling parameter that can be controlled externally. This circuit demonstrates the basic formulation of global normalization: the gain of individual elements is modified based on a global aggregate, in this case the sum (or average) of the input currents. Similar circuits have been created to compute a wide variety of nonlinear normalization functions using the same principle. Local normalization can also be implemented via the addition of resistive elements between the individual nodes, resulting in localized sums of input currents against which the input signals are compared.

Temporal adaptation. Normalization in time is accomplished through the combination of “fast” input variables normalized by “slow” adaptation variables. As in the case of spatial normalization, the basic premise of temporal adaptation is the averaging of the input signal—in this case, averaged over time using the slow variable—and the scaling of the input based on this average.

One of the simplest and most widely used adaptive circuits that implements temporal gain control is the adaptive photoreceptor (Delbruck and Mead, 1996). This circuit provides an analog output that has low gain for static signals and high gain for transient signals that vary about an operating point. The circuit’s adaptive function allows the output to represent a large dynamic range of absolute intensities while retaining sensitivity to small inputs. The adaptive photoreceptor has been used extensively in vision chips, including those that compute motion (Sarpeshkar et al., 1996) and visual attention (Morris and DeWeerth, 1999).

Signal Quantization

Signal quantization is seen throughout the nervous systems of animals. Information is transmitted using action potentials, which are temporally quantized forms of the analog membrane potentials from which they are derived. Spatial quantization is also widespread, taking forms such as the *place codings* found in motor maps in areas including colliculus and primary motor cortex. Such quantization is essential to system operation because multidimensional

signals must be represented by sets of individual neurons, each of which encodes one piece of a quantized signal.

Spatial quantization. One example of spatial quantization is the current-mode winner-take-all circuit (Lazzaro et al., 1989; Andreou et al., 1991), a variation of which is shown in Figure 3. In this circuit, pairs of MOS transistors (M_{1n} and M_{2n}) configured as *current conveyers* compete for current supplied to a common line. Each current conveyor sees a voltage V_T at its common node. Consequently, for conveyor n , if $I_n < I_0 Se^{kV_T}$, M_{1n} enters its linear region ($V_{DS} < 4$), turning M_{2n} off at that element. This condition occurs in all but one of the conveyers. At the element with the largest input, negative feedback in transistor M_{2n} adjusts V_T so that $I_n = I_0 Se^{kV_T}$. Thus, the conveyor with the largest input sets the voltage on the common node and conveys the common current I_T to its output. The result is a network that, for normal operation, selects the largest input by generating a non-zero output at only that element. Variants of the basic winner-take-all circuit have been used extensively in systems such as attention-based imagers (Morris and DeWeerth, 1999) and large classifier arrays (Pouliquen et al., 1997).

Temporal quantization. Temporal quantization is exemplified by the creation of neural “spikes” (action potentials) as a function of a continuous input current (Mead, 1989, chap. 12). For example, integrate-and-fire neurons convert continuous-value representations of signals into a discrete-value representation. Spiking neuron circuits are used widely in interface circuits to communicate information among different subsystems and is an essential component of the address event representation discussed in the next section. They have also been used extensively as the primary output devices in the pulse-modulated control of motor systems.

Representation and Communication

To construct systems from the circuits and networks described previously, we must have a paradigm for representing and communicating signals over long distances (including between VLSI chips) with potentially significant fan-in and fan-out. The massive connectivity of the brain is impossible to implement directly using VLSI because of wiring limitations within and between microchips. We can, however, exploit the temporally sparse nature of spike codes and the high bandwidth of VLSI systems in order to overcome this connectivity problem by time-multiplexing signals from many connections onto a single high-speed data bus. The resulting

address-event representation (AER) (Lazzaro et al., 1993) has emerged as a general paradigm for communicating large amounts of data in distributed spatiotemporal architectures. The core paradigm represents individual local activity that is encoded by neural spikes as *events*. Each of these events represents the origin/destination and timing information of a single spike, and is communicated on high-speed data buses throughout the network.

Because AER was originally formulated to emulate the optic nerve and the auditory nerve (Lazzaro et al., 1993), it implements a one-to-one connection topology. To implement more complex neural circuits, convergent and divergent connections are required. For example, architectures have been developed for emulating short and long connections along the spinal cord (DeWeerth et al., 1997), and for memory-based projective field mappings that enable the projection of an address event to multiple receiver locations (Boahen, 2000).

Discussion

Neuromorphic engineering represents a design paradigm that combines biological inspiration with microsystems technology to facilitate the design and implementation of systems that address a wide variety of tasks that are presently beyond the abilities of today’s computer systems but appear accessible and even trivial to the nervous systems of the simplest animals. The field is based on the premise that it is the nonlinearities and dynamics, the circuit architectures, and the basic computational paradigms present in these biological systems—most of which are foreign to traditional engineered systems—that make them successful. The ultimate goal of this field is to abstract these organizational principles by implementing them in real, physical systems in order to create artificial systems that exploit the power of biological computation.

In this article we have discussed neuromorphic electronic systems at three levels of organization: devices, circuits/networks, and systems, with an emphasis on the intermediate level. At the device level, we presented the basic devices—transistors and capacitors—that form the foundation for the other levels. At the circuit/network level, we described a set of essential circuits and networks that are employed as building blocks for a wide variety of systems. A full appreciation of the value of these circuits can be attained only by studying their use in larger systems. For more detailed coverage of this rapidly developing field, readers are referred for such excellent journals as *IEEE Transactions on Circuits and Systems*, *IEEE Transactions on Neural Networks*, *IEEE Journal of Solid-State Circuits*, *Analog Integrated Circuits and Signal Processing*, *Neural*

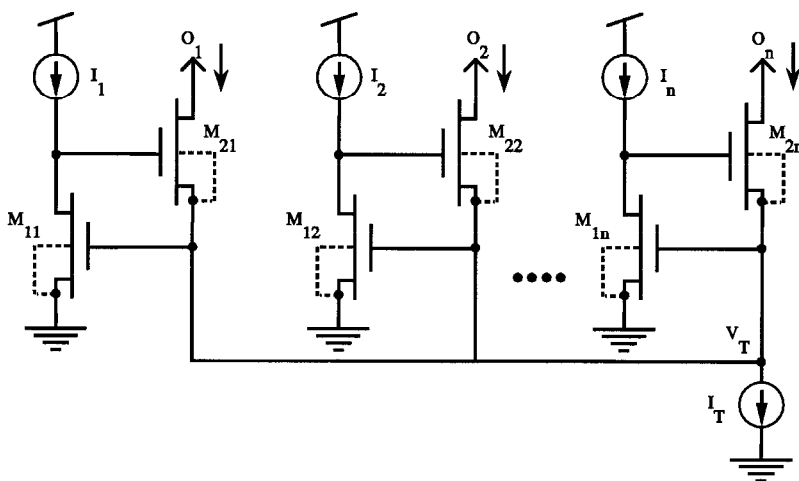


Figure 3. Winner-take-all circuit with current input and current output signals.

Computation, and Neural Networks, which carry regular articles and have special issues on hardware implementations of neuromorphic microsystems.

Road Map: Implementation and Analysis

Related Reading: Analog VLSI Implementation of Neural Networks; Digital VLSI for Neural Networks; Photonic Implementations of Neurobiologically Inspired Networks

References

- Andreou, A., and Boahen, K., 1996, Translinear circuits in subthreshold CMOS, *Analog Integr. Circuits Sign. Process.*, 9:141–166.
- Andreou, A. G., and Boahen, K. A., 1991, Current-mode subthreshold MOS circuits for analog VLSI neural systems, *IEEE Trans. Neural Netw.*, 2:205–213.
- Andreou, A. G., and Meitzler, R. C., 1995, Analog VLSI neuromorphic image acquisition and pre-processing systems, *Neural Netw.*, 8:1323–1347.
- Boahen, K. A., 2000, Point-to-point connectivity between neuromorphic chips using address events, *IEEE Trans. Circuits Syst. II Analog Digital Sign. Process.*, 47:416–434.
- Delbruck, T., and Mead, C. A., 1996, *Analog VLSI Phototransduction by Continuous-Time, Adaptive, Logarithmic Photoreceptor Circuits*, technical report, Caltech CNS Memo No. 30.
- DeWeerth, S. P., Patel, G. N., Simoni, M., Schimmel, D., and Calabrese, R., 1997, A VLSI architecture for modeling intersegmental coordination, in *Proceedings of the 17th Conference on Advanced Research in VLSI*, pp. 182–200, Los Alamitos, CA.
- Gilbert, B., 1996, Translinear circuits: An historical overview, *Analog Integr. Circuits Sign. Process.*, 9:95–118.
- Lazzaro, J., Ryskebusch, S., Mahowald, M. A., and Mead, C. A., 1989, Winner-take-all circuits, in *Advances in Neural Information Processing Systems*, vol. 1 (D. S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 703–711.
- Lazzaro, J., Wawrzyniek, J., Mahowald, M., Sivilotti, M., and Gillespie, D., 1993, Silicon auditory processors as computer peripherals, *IEEE Trans. Neural Netw.*, 4:523–528. ♦
- Mead, C. A., 1989, *Analog VLSI and Neural Systems*, Reading, MA: Addison-Wesley. ♦
- Morris, T. G., and DeWeerth, S. P., 1999, A smart-scanning analog VLSI visual-attention system, *Int. J. Analog Integr. Circuits Sign. Process.*, 21:67–78.
- Pouliquen, P. O., Andreou, A., and Strohbehn, K., 1997, Winner-takes-all associative memory: A hamming distance vector quantizer, *Analog Integr. Circuits Sign. Process.*, 13:211–222.
- Sarpeshkar, R., Kramer, J., Indiveri, G., and Koch, C., 1996, Analog VLSI architectures for motion processing: From fundamental limits to system applications, *Proc. IEEE*, 84:969–987. ♦
- Vittoz, E. A., 1985, The design of high-performance analog circuits on digital CMOS chips, *IEEE J. Solid State Circuits*, 20:657–665.
- Vittoz, E. A., 2002, Present and future industrial applications of bio-inspired VLSI systems, *Analog Integr. Circuits Sign. Process.*, 30:173–184.

NEURON Simulation Environment

Michael L. Hines and N. Ted Carnevale

Introduction

NEURON is designed to be a convenient and efficient environment for simulating models of biological and artificial neurons, individually and in networks. Great care has been exercised at every point in its development to achieve computational efficiency and robustness while helping users maintain conceptual clarity, i.e., the knowledge that what has been instantiated in the computer is an accurate implementation of one's conceptual model. NEURON's application domain extends beyond continuous system simulations of models of individual neurons with complex anatomical and biophysical properties, and includes discrete-event and hybrid simulations that combine biological and artificial neuronal models. Here we review features of NEURON that are of special interest to prospective users.

Who Uses NEURON, and Why?

Presently, more than 300 papers have reported research performed with NEURON (see <http://www.neuron.yale.edu/neuron/hib/usednrm.html>). Among the research topics are descriptions of models of individual neurons and networks of neurons with properties such as complex branching morphology, multiple channel types, inhomogeneous channel distribution, ionic diffusion and buffering, active transport, second messengers, and use-dependent synaptic plasticity. At the cellular level, NEURON has been used to investigate pre- and postsynaptic mechanisms involved in synaptic transmission, the roles of dendritic architecture and active membrane properties in synaptic integration, spike initiation and propagation in dendrites and axons, the effects of developmental changes of anatomy and biophysics, the functional genomics of ion channels, and extracellular stimulation and recording, among other topics.

Network models implemented with NEURON have been used to address issues such as the origin of cortical and thalamic oscillations, the role of gap junctions in neuronal synchrony, information encoding in biological networks, visual orientation selectivity, mechanisms of epilepsy, and the actions of anticonvulsant drugs.

NEURON is also being used in neuroscience education at the undergraduate and graduate level at numerous universities across the United States and around the world. Many of these courses are completely homegrown, but one lab manual with exercises has already appeared in print (Moore and Stuart, 2000), and we are involved in a collaboration to develop another set of laboratory exercises for publication. NEURON is particularly well suited to educational applications, since special expertise in numerical methods or programming is not required for its productive use. Furthermore, NEURON runs under MacOS, MSWindows, and UNIX/Linux, and can execute research-quality simulations with reasonable run times on entry-level hardware.

How Does NEURON Work?

Historically, NEURON's primary domain of application was in simulating empirically based models of biological neurons with extended geometry and biophysical mechanisms that are spatially nonuniform and kinetically complex. In the past decade its functionality was enhanced to include extracellular fields, linear circuits to emulate the effects of nonideal instrumentation, models of artificial (integrate-and-fire) neurons, and networks that can involve any combination of artificial and biological neuron models. The following sections outline how these capabilities have been implemented so as to achieve computational efficiency while maintaining conceptual clarity, i.e., the knowledge that what has been instan-

tiated in the computer model is an accurate representation of the user's conceptual model.

Representing Biological Neurons

Information processing in the nervous system involves the spread and interaction of electrical and chemical signals within and between neurons and glia. These signals are continuous functions of time and space and are described by the diffusion equation and the closely related cable equation (Rall, 1977; Crank, 1979). To simulate the operation of biological neurons, NEURON uses the tactic of discretizing time and space, approximating these partial differential equations by a set of algebraic difference equations that can be solved numerically (numerical integration) (Hines and Carnevale, 1997).

Discretization is often couched in terms of compartmentalization, but it is perhaps better to regard it as an approximation of the original continuous system by another system that is discontinuous in time and space. Simulating a discretized model results in computation of the values of spatiotemporally continuous variables over a set of discrete points in space ("nodes") for a finite number of instants in time. If NEURON's second-order-correct integration method is used, these values are a piecewise linear approximation to the continuous system, so that second-order-accurate estimates of continuous variables at intermediate locations can be found by linear interpolation (Hines and Carnevale, 2001).

In one form or another, spatial discretization lies at the core of all simulators used to model biological neurons, e.g., GENESIS SIMULATION SYSTEM (q.v.). Unlike other simulators, however, NEURON does not force users to deal with compartments. Instead, NEURON's basic building block is the section, an unbranched, continuous cable whose anatomical and biophysical properties can vary continuously along its length. The branched architecture of a cell is reconstructed by connecting sections together, each section having its own anatomical dimensions, biophysical properties, and discretization parameter *nseg*, which specifies the number of nodes at which solutions are computed. This strategy makes it easier to manage anatomically detailed models, since each section in the model is a direct counterpart to a branch of the original cell (Figure 1). Furthermore, neuroscientists naturally tend to think in terms of axonal or dendritic branches rather than compartments.

But even in topologically simple cases, there is still the problem of how to treat variables that are continuous functions of space. Thinking in terms of compartments leads to representations that require users to keep track of which compartments correspond to which anatomical locations. If we change the size or number of compartments, e.g., in order to see whether spatial discretization is adequate for numerical accuracy, we must also abandon the old mapping between compartments and locations in favor of a completely new one.

This is the motivation for another strategy that helps NEURON users maintain conceptual clarity: *range* and *range variables*. Range variables are continuous functions of position along a branch of a cell, e.g., diameter, membrane potential, or ion channel density. NEURON deals with range variables in terms of arc length (normalized distance) along the centroid of each section. This normalized distance, which is called *range*, is a continuous parameter that varies from 0 at one end to 1 at the other. In NEURON's programming language *hoc*, the membrane potential at a point 700 μm down the length of a 1,000- μm -long axon would be called `axon.v(0.7)`, regardless of the value of the axon's discretization parameter *nseg*. Range and range variables allow NEURON itself to take care of the correspondence between nodes and anatomical location. This avoids the tendency of compartmental approaches to confound representation of the physical properties of neurons, which are biologically relevant, with implementational

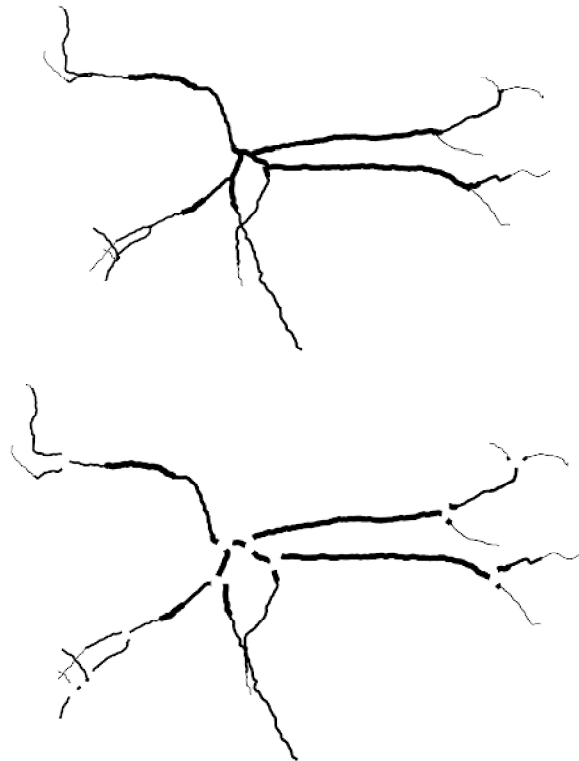


Figure 1. *Top*, Morphometric reconstruction of a hippocampal interneuron (data from A. I. Gulyás). *Bottom*, An "exploded" view, in which individual, unbranched neurites have been separated from each other at branch points.

details such as compartment size, which are mere artifacts of having to use a digital computer to emulate the behavior of a distributed physical system that is continuous in time and space.

Representing Artificial Neurons

In NEURON, the basic difference between biological and artificial neuron models is that the former may have arbitrarily complex anatomical and biophysical complexity, while the latter have no spatial extent and employ highly simplified kinetics. Indeed, the three built-in classes of artificial spiking neurons are so simple that they are simulated using a discrete-event method, which executes hundreds of times faster than numerical integration methods. If an event occurs at time t_1 , all state variables are computed from the state values and time t_0 of the previous event. Since computations are performed only when an event occurs, total computation time is proportional to the number of events delivered and independent of the number of cells, number of connections, or problem time. Thus, handling 100,000 spikes in 1 hour for 100 cells requires the same time as handling 100,000 spikes in 1 s for one cell. This takes advantage of NEURON's event delivery system, which was originally implemented to facilitate efficient network simulations of biological neurons (see following section).

Three different classes of integrate-and-fire models are built into NEURON. The simplest is IntFire1, a leaky integrator that treats input events as weighted delta functions. When an IntFire1 cell receives an input event of weight w , its "membrane potential" state m jumps instantaneously by an amount equal to w and thereafter resumes its decay toward 0 with time constant τ_m .

A step closer to the behavior of a biological neuron is the IntFire2 mechanism, which differs in that m integrates a net syn-

aptic current i . An input to an IntFire2 cell makes the synaptic current jump by an amount equal to the synaptic weight, after which i continues to decay toward a steady level i_b with its own time constant τ_s , where $\tau_s > \tau_m$. Thus a single input event produces a gradual change in m with a delayed peak, and cell firing does not obliterate all traces of prior synaptic activation. The firing rate is $\sim i/\tau_m$ if $i \gg 1$ and $\tau_s \gg \tau_m$.

Although IntFire2 can emulate a wide range of relationships between input pattern and firing rate, all inputs produce responses with the same kinetics, regardless of whether they are excitatory or inhibitory. The fact that synaptic excitation in biological neurons is generally faster than inhibition inspired the design of IntFire4. IntFire4 integrates two synaptic current components that have different dynamics, depending on whether the input event is excitatory or inhibitory (Figure 2). Excitatory inputs add instantaneously to an excitatory synaptic current e , which otherwise decays toward 0 with a single time constant τ_e ; this is analogous to IntFire2 with $i_b = 0$. However, the inhibitory synaptic current i_2 is described by the reaction scheme



where an inhibitory input (weight $w < 0$) adds instantaneously to i_1 , so that i_2 follows a biexponential course (a slow rise followed by an even slower decay).

These are not the only kinds of artificial neurons that can be simulated using discrete events. The only prerequisite for discrete event simulations is that all state variables of a model cell can be

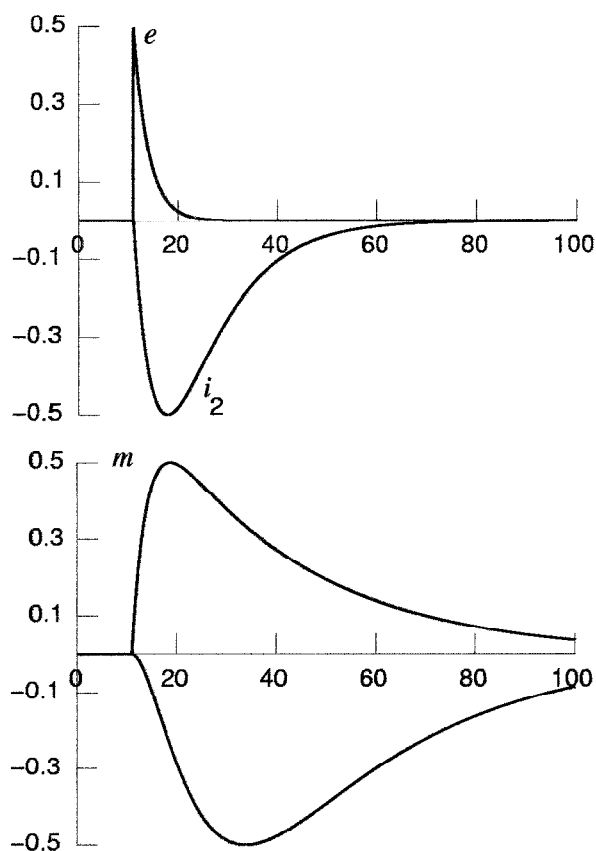


Figure 2. *Top*, Current generated by a single input event to an IntFire4 cell with weight 0.5 (e) or -0.5 (i_2). *Bottom*, The corresponding response of m . Parameters are $\tau_e = 3$, $\tau_{i1} = 5$, $\tau_{i2} = 10$, and $\tau_m = 30$ ms.

computed analytically from a new set of initial conditions. Users who have special needs can add other kinds of artificial neuron classes to NEURON with the NMODL language (discussed later).

Representing Networks

NEURON can handle networks that involve gap junctions or graded transmitter release, but this discussion is restricted to spiking networks. NEURON's NetCon class and event delivery system are used to manage synaptic communication between any combination of artificial and biological model neurons in a network. The event delivery system can also serve other purposes, such as parameter changes on the fly, and it is a key part of the implementation of the built-in integrate-and-fire models, but these topics are beyond the scope of this article.

NEURON's strategy for dealing with synaptic connections emerged from techniques initially developed by Destexhe, Mainen, and Sejnowski (1994) and Lytton (1996). This strategy is based on a very simple conceptual model of synaptic transmission: arrival of a spike at the presynaptic terminal causes transmitter release, which in turn perturbs some mechanism in the postsynaptic cell (e.g., a membrane conductance or second messenger) that is described by a differential equation or kinetic scheme. All that matters is whether or not a spike has occurred in the presynaptic cell; mechanistic details in the presynaptic and postsynaptic cells do not affect transmitter release. This conceptual model separates the specification of the connections between cells from the specification of the postsynaptic mechanisms that the connections activate.

A presynaptic spike triggers an event that, after a delay to account for conduction along the axon, transmitter release, and diffusion time, is delivered to the postsynaptic mechanism, where it causes a change in some variable (e.g., a conductance). Event delivery is computationally efficient because of how synaptic divergence ("fan out") and convergence ("fan in") are handled. Synaptic divergence from model biological neurons is efficient because threshold detection is performed on a per source basis rather than a per connection basis. That is, if a neuron projects to multiple targets, the presynaptic variable is checked only once per time step, and when it crosses threshold an event is generated for each of the targets. Fan-out from artificial neurons is also very efficient, since their discrete event mechanisms do not have to be checked at each time step. However, the greatest computational savings are offered by synaptic convergence onto model biological neurons. Suppose a neuron receives multiple synaptic inputs that are close to each other and of the same type, i.e., each synapse has the same kind of postsynaptic mechanism. Then the total effect of all these synapses can be represented by a single kinetic scheme or set of equations driven by multiple input streams, each of which has its own weight.

The implementation of the event delivery service in NEURON takes into account the fact that the delay between initiation and delivery of events is different for different streams. Consequently, the order in which events are generated by a set of sources is rarely the order in which they are received by a set of targets. Furthermore, the delay may be 0 or 10^9 or anything in between.

As noted earlier, NEURON separates specification of the connections between cells from specification of the postsynaptic mechanisms that the connections activate. This separation means that NEURON models are compatible with other event delivery systems, such as the parallel discrete event delivery system used by NeoSim (Goddard et al., 2001).

Integration Methods

Users can choose among several different integration methods, but the "best" method for any given problem depends on many factors and may require empirical testing. Every choice involves trade-offs

between accuracy, on the one hand, and stability and/or run time, on the other. For more extensive treatments of this topic, see Hines and Carnevale (1995, 1997, 2001).

Two methods use fixed time steps: backward Euler, and a Crank-Nicholson variant. NEURON's default integrator is backward Euler, which is first-order accurate in time, inherently stable, and generally produces good qualitative results even with large time steps. Used with extremely large Δt , it will find the steady-state solution of a linear ("passive") model in one step, and quickly converge to a steady state for nonlinear models. The Crank-Nicholson (CN) variant employs a staggered time step in order to provide second-order accuracy without having to iterate nonlinear equations. The computational cost of a single time step is practically the same as for backward Euler, but much shorter run times are possible with CN because it can use a larger Δt for a given degree of accuracy. However, CN cannot be used with models that involve purely algebraic relations between states, and it can produce spurious oscillations if Δt is too large or if the model includes a fast voltage clamp.

Models that work with CN are generally also amenable to CVODE (Cohen and Hindmarsh, 1984), the variable order, variable time step method offered by NEURON. For a given run time, CVODE often yields greater accuracy than CN. CVODE is usually the best choice for network models that involve artificial neurons. NEURON also offers a local variable order, variable time step method in which each cell has its own time step. This can be advantageous for network models in which most cells are silent most of the time.

Development Environment

Constructing and managing models and controlling simulations can be accomplished with an object-oriented interpreter, a set of GUI tools, or a combination of both. Most common tasks can be performed with the GUI tools, which are especially convenient for exploratory simulations during model development. Where the GUI is inadequate, users can resort to the interpreter, which is based on hoc (Kernighan and Pike, 1984). The interpreter is also appropriate for noninteractive simulations, such as production runs that generate large amounts of data for later analysis. Even so, several of the GUI tools are quite powerful in their own right, offering functionality that would require significant effort for users to recreate in hoc. This is particularly true of the optimization and electrotonic analysis tools. Thus, the most flexible and productive use of NEURON is to combine the GUI and hoc programming, taking advantage of the strengths of both.

Because of the ever-growing number and diversity of ligand- and voltage-gated ionic currents, pumps, buffers, etc., NEURON has a special facility for expanding its library of biophysical mechanisms (Hines and Carnevale, 2000). A user can write a text file that contains a description of the mechanism in NMODL, a programming language whose syntax for expressing nonlinear algebraic equations, differential equations, and kinetic reaction schemes closely resembles familiar notation. This file is then converted into C by a translator that automatically generates code for handling details such as mass balance for each ionic species. The translator output, which includes code that is suitable for each of NEURON's integration methods, is then compiled for computational efficiency. This achieves tremendous conceptual leverage and savings of effort because the high-level mechanism specification in NMODL is much easier to understand and far more compact than the equivalent C code, and the user is not bothered with low-level programming issues like how to interface the code with other mechanisms and with NEURON itself.

NEURON runs under MacOS, MSWindows, and UNIX/Linux, with a similar X11-based look and feel on all platforms. Further-

more, the same hoc and NMODL code works under all these operating systems without modification. This facilitates collaborations in a heterogeneous computing environment.

Distribution, Documentation, and Support

NEURON is available free of charge from <http://www.neuron.yale.edu/>, along with extensive documentation and tutorials. The UNIX/Linux distribution includes full source code; the MSWin and MacOS distributions employ an identical computational engine and come with the hoc code that implements the GUI and the NMODL definitions of the built-in biophysical mechanisms and artificial neuron classes.

The web site also has a sign-up page for joining the NEURON Users' Group, a moderated mailing list for questions and answers, and pertinent announcements such as program updates and courses on NEURON. For the past several years we have presented "executive summary" and intensive hands-on courses at sites in the United States and Europe, and we plan to continue this in the future. NEURON is actively supported by a development team that responds to bug reports and questions about program usage. Indeed, much of the program's current functionality has been stimulated by requests and suggestions from users, and we are grateful to them for their continued interest and encouragement.

Discussion

The level of detail included in NEURON models can extend from a single compartment with linear membrane, to intricate extended architectures with membrane and cytoplasm that have complex biophysical properties. There are also several classes of artificial spiking neurons. Networks can involve biological neurons, artificial neurons, or both. Models can be simulated with fixed time step or with variable order, variable time step methods. With the variable time step integrator, the built-in artificial neurons are simulated as discrete event models, executing orders of magnitude faster than models of biological neurons do. Users can add new kinds of biophysical mechanisms and artificial neuron classes to NEURON's built-in library. NEURON runs on a wide variety of platforms, is actively supported, and is under continuous development, with revisions and updates that address the evolving needs of users. Because of these features, NEURON is employed in research on topics that range from the biophysical basis of neuronal function at the subcellular level to the operation of large-scale networks involved in consciousness, perception, learning, and memory. It is also increasingly being adopted for neuroscience education.

Road Map: Implementation and Analysis

Related Reading: Neurosimulation: Tools and Resources; Perspective on Neuron Model Complexity

References

- Cohen, S. D., and Hindmarsh, A. C., 1984, *CVODE User Guide*, Livermore, CA: Lawrence Livermore National Laboratory.
- Crank, J., 1979, *The Mathematics of Diffusion*, 2nd ed., London: Oxford University Press. ◆
- Destexhe, A., Mainen, Z. F., and Sejnowski, T. J., 1994, An efficient method for computing synaptic conductances based on a kinetic model of receptor binding, *Neural Computat.*, 6:14–18.
- Goddard, N., Hood, G., Howell, F., Hines, M., and De Schutter, E., 2001, NEOSIM: Portable large-scale plug and play modelling, *Neurocomputing*, 38:1657–1661. ◆
- Hines, M., and Carnevale, N. T., 1995, Computer modeling methods for neurons, in *Handbook of Brain Theory and Neural Networks* (M. A. Arbib, Ed.), Cambridge, MA: MIT Press, pp. 226–230. ◆
- Hines, M. L., and Carnevale, N. T., 1997, The NEURON simulation environment, *Neural Computat.*, 9:1179–1209. ◆

- Hines, M. L., and Carnevale, N. T., 2000, Expanding NEURON's repertoire of mechanisms with NMODL, *Neural Computat.*, 12:839–851. ♦
- Hines, M. L., and Carnevale, N. T., 2001, NEURON: A tool for neuroscientists, *Neuroscientist*, 7:123–135. ♦
- Kernighan, B. W., and Pike, R., 1984, Appendix 2: Hoc manual, in *The UNIX Programming Environment*, Englewood Cliffs, NJ: Prentice-Hall, pp. 329–333. ♦

- Lytton, W. W., 1996, Optimizing synaptic conductance calculation for network simulations, *Neural Computat.*, 8:501–509.
- Moore, J. W., and Stuart, A. E., 2000, *Neurons in Action: Computer Simulations with NeuroLab*, Sunderland, MA: Sinauer.
- Rall, W., 1977, Core conductor theory and cable properties of neurons, in *Handbook of Physiology*, vol. 1, part 1, *The Nervous System* (E. R. Kandel, Ed.), Bethesda, MD: American Physiological Society, pp. 39–98. ♦

Neuropsychological Impairments

Martha J. Farah

Why Model Neuropsychological Impairments?

Neuropsychological impairments are an important source of evidence about the organization of cognition in the normal brain. For example, the finding that amnesic patients retain the ability to learn certain kinds of implicit information led to the idea of multiple memory systems and the distinction between explicit and implicit learning, both key insights in modern memory research (Squire, 1987). However, the inferences that link a neuropsychological impairment to a particular theory in cognitive neuroscience are not as direct as one might first assume. A patient's behavior following brain damage is not necessarily determined by a simple subtraction of one or more components of the mind or brain, with those that remain functioning normally. The brain is a distributed and highly interactive system, such that local damage to one part can unleash new modes of functioning in the remaining parts of the system. Thus, the link between neuropsychological impairments and models of the normal system must take into account not only the subtraction of one of more components of that system, but also changes in the functioning of other components that had previously been influenced by the missing components.

Neural network models of cognition and the brain provide a framework for reasoning about the effects of local lesions in distributed, interactive systems (Farah, 1994). Computer simulations of such models allow us to test hypotheses concerning the normal cognitive system using data from neurological patients, by simulating the candidate systems. The simulations of normal systems can be "lesioned," for example by removing their neuron-like units, or connections between units, and the behavior of the lesioned system can be compared with the behavior of patients.

In many cases a model's behavior after lesioning is somewhat counterintuitive. Indeed, it is often very different from what one would expect by reasoning in terms of simple deletion of parts from a normal system, with minimal interactions among the parts. For this reason, the use of neural networks when interpreting neuropsychological impairments can lead to very different interpretations regarding the nature of the normal system.

Examples of Neural Network Models in Neuropsychology

The remainder of this article will back up these statements with some specific examples of well-known neuropsychological impairments whose interpretations vis-à-vis the normal brain have been changed dramatically by neural network modeling. In each example the neural networks are highly simplified models of real brain tissue, in the tradition of parallel distributed processing (PDP; Rumelhart and McClelland, 1986).

PDP systems consist of a large number of highly interconnected neuron-like units. These units are connected to one another by

weighted connections that determine how much activation from one unit flows to another. Each part of the network functions locally and in parallel with the other parts; hence the first P in PDP. Representations consist of the pattern of activation distributed over a population of units, and long-term memory knowledge is encoded in the pattern of connection strengths distributed among a population of units; hence the D. There are many types of PDP networks with different computational properties. Among the features that determine network type are the activation rule, connectivity, and the learning rule.

PDP models differ from real neural networks, including the human brain, in numerous ways. Even the biggest PDP networks are tiny compared with the brain. PDP models have just one kind of unit, compared with a variety of types of neurons, and just one kind of activation (which can act excitatorily or inhibitorily) rather than a multitude of different neurotransmitters, and so on. Of course, all models are simplifications of reality and possess both theory-relevant and theory-irrelevant features. Among the theory-relevant features of PDP models are the use of distributed representations, the large number of inputs to and outputs from each unit, the modifiable connections between units, the existence of both inhibitory and excitatory connections, summation rules, bounded activations, and thresholds. PDP models allow us to find out what aspects of behavior, normal and pathological, can be explained by this set of theory-relevant attributes. The three examples that follow demonstrate the explanatory work that can be done in neuropsychology with such models.

Interpreting Error Types: The Case of Deep Dyslexia

Neuropsychologists have long assumed that the nature of the damaged component could be inferred from the kind of errors made. In a syndrome known as deep dyslexia, patients make two kinds of errors. They make semantic errors, that is, errors that bear a semantic similarity to the correct word, such as reading *cat* as *dog*. They also make visual errors, that is, errors that bear a visual similarity to the correct word, such as reading *cat* as *cot*. The most straightforward interpretation would seem to be that deep dyslexics have at least two lesions, with one affecting the visual system and another affecting semantic knowledge. However, a consideration of the effects of single lesions in a neural network with attractor states suggests that a single lesion is sufficient to account for these patients' errors. Furthermore, it suggests that mixtures of error types will be the rule rather than the exception when the system that has been damaged normally functions to transform the stimulus representation from one form that has one set of similarity relations (e.g., visual, in which *cot* and *cat* are similar) to another form with different similarity relations (e.g., semantic, in which *cot* and *bed* are similar).

Hinton and Shallice (1991) trained the recurrent network shown in Figure 1 to produce semantic representations of a set of words, given their printed orthography as input. The grapheme-to-“sememe” (their term for elements of semantic representation) mapping is carried out with the aid of hidden units, and the sememes are interconnected among themselves and connected to a final layer of semantic representation that connects, recurrently, back to the sememes. This pattern of connectivity in the semantic layers creates attractor states for the network. The input to the semantic layers need not be perfectly on target for the semantics of a particular word; as long as it is sufficiently similar to the correct semantics, which is an attractor state, it will be pulled in (i.e., as long as it falls in a region of activation space that slopes downward to the correct activation pattern, it will be transformed into that pattern). Damage to the network, from the removal of units or connections, distorts the shape of the activation space. Figure 2 illustrates the normal attractor structure of a region of activation space containing *cot*, *cat*, and *bed*, and the altered structure following damage to semantics. Whereas before damage, “cat” fell into the *cat* basin of attraction, after damage the edges of the basins have shifted and “cat” falls into the *cot* basin of attraction. Thus, one need not hypothesize damage to visual representations to account for the visual errors in deep dyslexia.

Plaut and Shallice (1993) have demonstrated the generality of Hinton and Shallice’s account by replicating their simulation results with a variety of different networks, with different patterns of connectivity and different training procedures. As long as there are attractors that serve to transform input patterns whose similarity relations are based on visual appearance into semantic representations whose similarity relations are based on meaning, the landscape of the activation space will be organized by both visual and semantic similarity, and distortions of that landscape due to network damage will result in both visual and semantic errors.

Determining a Processing Sequence: The Example of Neglect Dyslexia

Patients with left visual neglect omit or misidentify letters on the left side of letter strings. When the letter string is a word, this pattern of performance is termed neglect dyslexia. Surprisingly, neglect dyslexics are more likely to report the initial letters of a

word than of a nonword letter string, even when the initial letters of the word cannot simply be guessed on the basis of the end of the word. This seems to have a fairly specific, though surprising, implication for the order in which word recognition and spatial attention occur in the brain. If lexical status (word versus nonword) affects the allocation of attention, then it would seem that word recognition occurs before attention is allocated.

The concept of attractors is helpful here too, in understanding how an impairment in a prelexical attentional process could nevertheless show a lexicality effect. Mozer and Behrmann (1990) simulated neglect dyslexia by damaging the attentional mechanism in a computational model of printed word recognition so that attention was distributed asymmetrically over letter strings. In their model, attention *preceded* word recognition. In fact, it gated the flow of information out of early visual feature maps. Neglect therefore resulted in full information from the right side of a letter string, but only partial information from the left, being transmitted to word representations.

According to this model, the errors that occur with nonword letter strings result from partial visual information about the letter features on the left side of the string, which is not sufficient to identify precisely which letters are present. In contrast, the same partial information about the initial letters of a word, with good-quality information about the remaining letters of a word, will result in an activation pattern that is similar to the activation pattern for that word. Because known words are attractors, the network will settle into the pattern of the word, complete with initial letters. In this way, it is possible to explain why neglect dyslexics read words better than nonwords, without giving up the hypothesis that neglect is a disorder of visual perception that affects stimulus processing prior to the recognition stage.

It is worth noting that computational models make predictions that can be tested empirically. According to this model of neglect dyslexia, if the asymmetry of attention is too extreme, no information about the initial letters will get through to word representations, and the resulting activation state will not fall within the basin of attraction for the word. Behrmann et al. (1990) tested this prediction with a patient who had severe neglect. As predicted, he did not show better perception of the initial letters of words than nonwords. Furthermore, when his attention was drawn to the left, and the attentional asymmetry thereby made less extreme, he

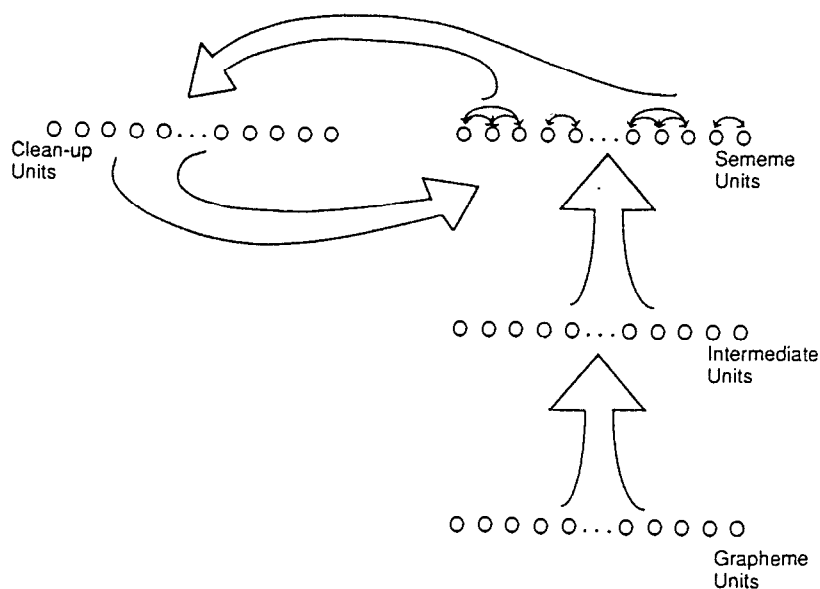
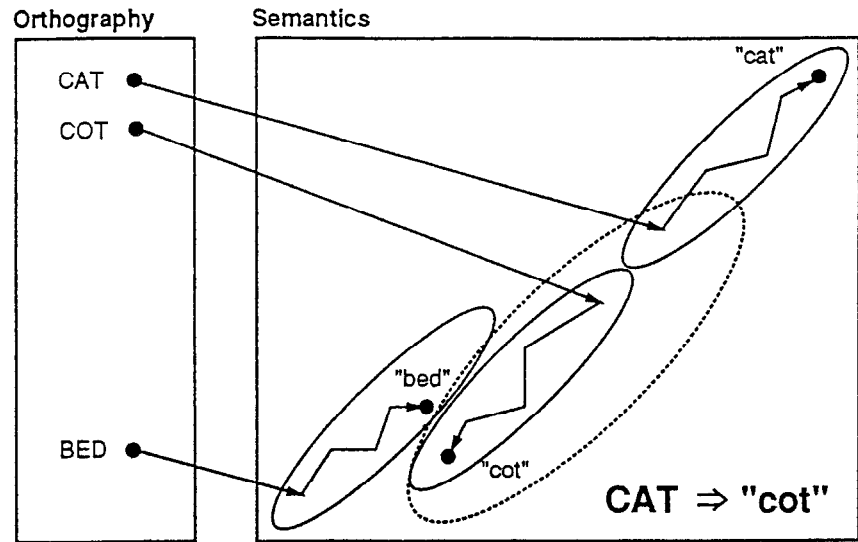


Figure 1. Hinton and Shallice’s (1991) PDP model of reading, in which visual graphemic representations are associated with semantic representations. Single lesions in this model produce a mixture of visual and semantic errors.

Figure 2. Part of the activation space of the Hinton and Shallice model as represented by Plaut and Shallice (1993), showing attractors for three words. After damage to semantic units, the basins of attraction shift from those shown in solid lines to those shown in dotted lines, resulting in visual errors.



showed the usual difference between word and nonword letter strings. Conversely, a patient who normally showed this difference between words and nonwords was stopped from doing so by attentional manipulations that increased his attentional asymmetry.

Dissociation Without Separate Systems: The Example of Covert Face Recognition

Prosopagnosia is an impairment of face recognition that can occur relatively independently of impairments in object recognition. The behavior of some prosopagnosic patients (described below) seems to suggest that recognition and awareness depend on dissociable and distinct brain systems (Figure 3). My colleagues and I built a computer simulation that is able to account for covert recognition in a number of different tasks (Farah, O'Reilly, and Vecera, 1993; O'Reilly and Farah, 1999). The network is shown in Figure 4 and consists of face recognition units, semantic knowledge units, and name units (embodying knowledge of people's facial appearance, general information about them, and their names, respectively). Hidden units were interposed between these layers to assist the network in learning to associate faces and names by way of semantic information. No part of the network is dedicated to awareness.

The first finding to be simulated was that some prosopagnosics can learn to associate facial photographs with names faster when the pairings are true (e.g., Harrison Ford's face with Harrison Ford's name) than when they are false (e.g., Harrison Ford's face with Michael Douglas's name; De Haan, Young, and Newcombe, 1987). This result was initially taken to imply that these patients were recognizing the faces normally, and that the breakdown in processing lay downstream from vision, as shown in Figure 3. However, when some of the face units were eliminated from our model, thus simulating a lesion in the visual system, the network also relearned old face-name pairings faster than new ones. Why should this be? We can think of learning as a process of moving through weight space. After damage, the network is in a high-error region of weight space for both old face-name pairings and new ones, and therefore the network cannot overtly associate any faces with any names. However, that region of weight space is closer to a low-error region for the old pairings than for the new ones, because the residual weights (connecting intact units) have the correct values for the old pairings, and the learning process is therefore shorter.

A second finding, that previously familiar faces are perceived more quickly in the context of a same/different matching task, has also been interpreted as evidence for intact visual face processing (De Haan et al., 1987). However, after lesions to the face units in our model, the remaining face units settled into a stable state faster for previously familiar face patterns. This can be understood in terms of the distortion of the network's attractor structure after damage. The original structure was designed to take familiar face patterns as input and settle quickly to a stable state. After damage,

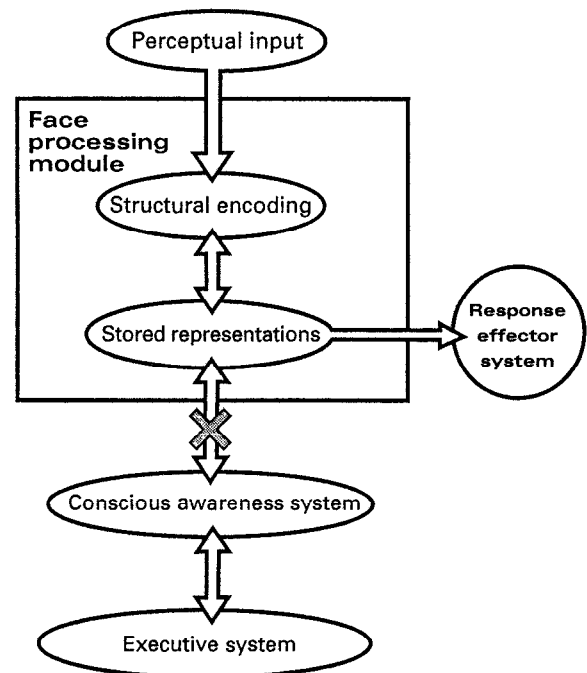


Figure 3. A model proposed by De Haan, Bauer, and Greve (1992) to account for covert face recognition in prosopagnosia. A separate mechanism for conscious awareness is hypothesized, distinct from the mechanisms of face recognition, and covert recognition is explained by a lesion at location X, disconnecting the two parts of the model.

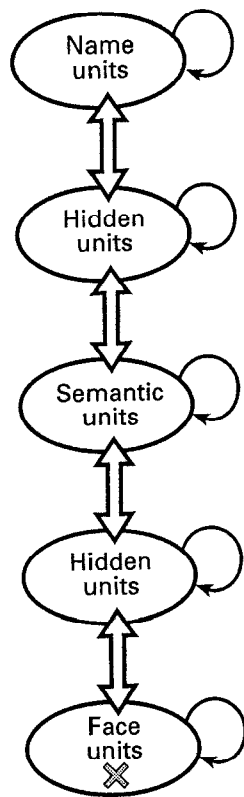


Figure 4. A model proposed by Farah, O'Reilly, and Vecera (1993) to account for covert face recognition in prosopagnosia. The dissociation between overt and covert face recognition emerges when the face recognition system is damaged.

these patterns will still find themselves on downward-sloping parts of the energy landscape more often than novel patterns, even if the energy minima into which they roll have changed.

In yet another task, one that requires classifying a printed name as belonging to an actor or a politician, both normal subjects and prosopagnosics are influenced by a face from the opposite occupation category shown in the background, again implying that the face is recognized despite prosopagnosia (De Haan et al., 1987).

To simulate this finding, we removed face units until the network's overt performance at classifying faces according to occupation was as poor as the patient's. At this level of damage, wrong-category faces slowed performance in the name classification task. This can be understood in terms of the distributed nature of representation in neural networks, which allows for partial representation of information when some but not all units representing a face have been eliminated. The partial information generally raises the activation of the appropriate downstream occupation units, thus biasing their responses to the printed names, but is not generally able to raise their activations above threshold to allow an explicit response to faces.

Road Map: Cognitive Neuroscience

Related Reading: Cognitive Development; Developmental Disorders; Face Recognition: Psychology and Connectionism; Reading

References

- Behrmann, M., Moscovitch, M., Black, S., and Mozer, M., 1990, Perceptual and conceptual mechanisms in neglect dyslexia: Two contrasting case studies, *Brain*, 113:1163–1183.
- De Haan, E. H. F., Bauer, R. M., and Greve, K. W., 1992, Behavioral and physiological evidence for covert recognition in a prosopagnosic patient, *Cortex*, 28:77–95. ♦
- De Haan, E. H. F., Young, A. W., and Newcombe, F., 1987, Face recognition without awareness, *Cogn. Neuropsychol.*, 4:385–415.
- Farah, M. J., 1994, Neuropsychological inference with an interactive brain: A critique of the locality assumption, *Behav. Brain Sci.*, 17:43–61.
- Farah, M. J., O'Reilly, R. C., and Vecera, S. P., 1993, Dissociated overt and covert recognition as an emergent property of lesioned attractor networks, *Psychol. Rev.*, 100:571–588. ♦
- Hinton, G. E., and Shallice, T., 1991, Lesioning an attractor network: Investigations of acquired dyslexia, *Psychol. Rev.*, 98:96–121.
- Mozer, M. C., and Behrmann, M., 1990, On the interaction of selective attention and lexical knowledge: A connectionist account of neglect dyslexia, *J. Cogn. Neurosci.*, 2:96–123.
- O'Reilly, R. C., and Farah, M. J., 1999, Simulation and explanation in neuropsychology and beyond, *Cogn. Neuropsychol.*, 16:49–72.
- Plaut, D. C., and Shallice, T., 1993, Deep dyslexia: A case study of connectionist neuropsychology, *Cogn. Neuropsychol.*, 10:377–500.
- Rumelhart, D. E., and McClelland, J. L., 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA: MIT Press.
- Sieroff, E., Pollatsek, A., and Posner, M. I., 1988, Recognition of visual letter strings following injury to the posterior visual spatial attention system, *Cogn. Neuropsychol.*, 5:427–449. ♦
- Squire, L. R., 1987, *Memory and Brain*, New York: Oxford University Press.

Neurosimulation: Tools and Resources

Randall D. Hayes, John H. Byrne, and Douglas A. Baxter

Introduction

In all scientific fields, including neuroscience, experimental hypotheses are based on models. Often, these models only qualitatively describe the relationships among the elements of a system, such as the molecules that make up a second-messenger cascade or the neurons that make up a network. Mathematical models provide a more rigorous framework within which investigators can organize large amounts of empirical data, test whether current data can account for the behavior of the system, identify critical features that warrant additional experimental investigation, and discover dynamic properties that are not intuitively obvious.

This article reviews neurosimulators, that is, programs designed to reduce the time and effort required to build models of neurons and neural networks. We include programs for modeling networks of spiking neurons as well as programs for kinetic modeling of intracellular signaling cascades and regulatory genetic networks. A comprehensive description of all neurosimulators is beyond the scope of this chapter (see also GENESIS SIMULATION SYSTEM, NEURON SIMULATION ENVIRONMENT, and NSL NEURAL SIMULATION LANGUAGE). Instead, we provide a general picture of the capabilities of several neurosimulators, highlighting some of the best features of the various programs, and refer the reader to more

specific information (Bower and Bolouri, 2002; Skrzypek, 1994; Koch and Segev, 1998; De Schutter, 2001). We do not list connectionist simulators in this chapter (see Murre, 1995).

This article also describes ongoing efforts to increase compatibility among the various programs, which serve two purposes. First, compatibility allows models built with one neurosimulator to be independently evaluated and extended by investigators using different programs, thereby reducing duplication of effort. Second, compatibility allows for models describing different levels of complexity (i.e., molecular, cellular, network) to be related to one another.

General Considerations

General considerations in selecting a simulator program include the hardware capabilities and programming expertise of the lab. Complicated models require fast workstations or networks of desktop computers running portions of the model in parallel (Hammarlund and Ekeberg, 1998). For investigators with limited resources, it may be necessary to simplify a model so that a less powerful computer can run a simulation in a reasonable amount of time. Building a model takes more time than simulating one, so using a simple program to build a simple model can speed up the process even further.

Hardware Issues

Platform requirements for various neurosimulators are listed in Tables 1 to 3. We include a few programs that can run on older, less expensive computers (e.g., *Nodus*, *MetaModel*). Programs written in Java or Perl (e.g., *CATACOMB*, *NSL*, *SNNAP*, *StochSim*) can run on multiple platforms without modification. Another solution to the problem of computer compatibility is to separate the simulator code, which is less platform sensitive, from the user interface code (e.g., *NEST/SYNOD*, *CONICAL*). For some simulators, the user can log onto a centralized network server through a web browser (e.g., *iCell*, *CMISS*, *Vcell*).

Level of Tech Support

An important issue is how much support the developers provide to the first-time user (see Tables 1 to 3). Most programs distribute manuals, which vary greatly in how complete and up-to-date they are. Buyers of commercial packages may receive personal instruction (e.g., *In Silico Cell*, *ModelMaker*, *SABER*). Another option is to attend short courses at workshops or national meetings (e.g., *GENESIS*, *Mcell*, *NEURON*, *SNNAP*). Many developers maintain e-mail lists or bulletin boards, where users can ask questions. Es-

pecially useful to the beginner are templates of possible conductances, cells, or circuits. These allow the user to start modeling immediately by modifying an existing file rather than beginning from scratch.

Numerical Methods

Discrete events. Connectionist simulators generally use continuous scalar variables to represent firing rate and synaptic weight (see SINGLE-CELL MODELS). The integrate-and-fire units used by discrete event neurosimulators (e.g., *NeuroImitator*, *SpikeNET*) are more realistic. They have a membrane potential that sums the cell's inputs over time; a spiking threshold, which determines when the cell will fire; and often a refractory hyperpolarization after the spike. When threshold potential is exceeded, a time-stamped spike event (not an action potential waveform) is generated, and the membrane potential is reset. After some fixed delay representing axonal conduction, the spike event triggers synaptic conductances in postsynaptic target cells. Note that most of the programs that calculate a waveform for each action potential by one of the methods described below also default to fixed axonal delays, rather than propagating the action potential (see AXONAL MODELING), to save computing time.

By adjusting the free parameters that control threshold and refractory period, the user can reproduce the firing rates and spike frequency adaptation of a wide range of biological neurons. Because individual conductances contributing to the membrane potential are ignored, discrete event simulators are suitable for studying network behaviors but not the intracellular mechanisms that produce them (see INTEGRATE-AND-FIRE NEURONS AND NETWORKS).

Ordinary differential equations (ODE). Most simulators represent individual neurons as systems of ODEs, which represent the average change in a dependent variable (usually membrane voltage or concentration) in a well-mixed spatial compartment over one time step of the simulation (Cobelli and Foster, 1998). The dependent variable is recalculated at regular intervals, with the intervals chosen to capture the fastest dynamics of the model (often tens of microseconds). In addition, some simulators (e.g., *BIOQ*, *NEURON*, *Surf-Hippo*, *XNBC*) provide variable time-step algorithms, which recalculate the dependent variable more often when it is changing rapidly and less often when it is changing slowly. These algorithms can shorten the computer time required to simulate large networks.

Monte Carlo methods. Other simulators explicitly follow each molecule in the simulation over time (e.g., *BIOQ*, *CKS*, *MCELL*, *StochSim*). At each time step, random numbers are chosen for the

Table 1. Programs Suitable for Demonstration and Teaching

Name	Topics	Control	Platform	Support	Web Site
<i>ArtMem/MemPot/MemCable</i> \$	P I A C	G	W, DOS	M	http://www.med.unsw.edu.au/PHBSoft/
<i>cLabs</i> \$	P I A C	G W	W		http://www.clabs.de/clabs.htm
<i>Computational Neuroscience</i> \$	P I A C N	G	M, W	T M	http://www.compneuro.org/
<i>iCell</i>	A C	W	M, W		http://ssd1.bme.memphis.edu/icell/
<i>NerveWorks</i>	P I A C N	S G	M, U, W	T M	http://nerve-works.com/
<i>NeuroDynamix</i> \$	P I A C N	G		T M	http://www.people.virginia.edu/~wof/pub
<i>NeuroSim</i> \$	P I A C N	G	W	T M	http://biology.st-and.ac.uk/sites/neurosim/
<i>Vclamp/Cclamp</i>	A C	C	W, DOS	T M	http://tonto.stanford.edu/~john/
<i>Electrophysiology o/t Neuron</i> \$	P I A C N	G	M, W	T M	http://tonto.stanford.edu/eotn/

\$ indicates that software must be purchased. Abbreviations under Topics: P = Passive membrane, I = Ion channel, A = Action potential, C = Cell, N = Network of multiple neurons. Abbreviations under Control: C = Command line, S = Scripts, G = Graphical user interface, W = Web browser. Abbreviations under Platform: M = MacOS, U = Unix, W = Windows. Abbreviations under Support: T = Templates, C = Courses, M = Manual, L = List.

Table 2. Packages Specialized for Neurobiological Modeling

Name	Spiking Neurons	Chemical Kinetics	Control	Platform	Support	Web Site
<i>BioPSE</i>	ODE		G W	U	T M L	http://www.sci.utah.edu/ncrr/software/biopse.html
<i>BIOSIM</i>	ODE		G	U, W	T M	ftp://ftp.uni-kl.de/pub/bio/neurobio/
<i>CATACOMB</i>	ODE	ODE	G	M, U, W	T	http://www.compneuro.org/catacomb/index.shtml
<i>CONICAL</i>	ODE		C S	M, U, DOS		http://www.strout.net/conical/
<i>GENESIS</i>	ODE	ODE	C S G	U	T C M L	http://www.genesis-sim.org/GENESIS/
<i>Maxsim</i>	ODE		C S G	U	M	http://ibcmgs6.unil.ch/staff/tettoni/maxsim/
<i>Mcell</i>	MC	MC	C S	U	T C M L	http://www.mcell.cnl.salk.edu/
<i>NeMoSys</i>	ODE	ODE	S G	U	T M	http://cns.montana.edu/research/nemosys
<i>NEST/SYNOD</i>	ODE InF		C S	U	T M	http://www.synod.uni-freiburg.de/
<i>Neurolmitator \$</i>	InF		G	W	T M	http://www.cellmc.com/ni/ni.html
<i>NEURON</i>	ODE InF	ODE	C S G	M, U, W	T C M L	http://www.neuron.yale.edu/
<i>NeuronC</i>	ODE	ODE	G	U, DOS		http://retina.anatomy.upenn.edu/~rob/neuronc.html
<i>Nodus</i>	ODE		G	M	M	http://bbf-www.uia.ac.be/SOFT/downloads.shtml
<i>NSL</i>	ODE InF		S G	U, W	T M	http://www-hbp.usc.edu/Projects/nsl.htm
<i>Neurosys</i>	ODE		C S G	M, U, W	T	http://nexus.cs.usfca.edu/neurosys/
<i>SEE</i>	ODE	ODE	S G	U, Cray		http://debian.nada.kth.se/sans.php?cont=tools
<i>SNNAP</i>	ODE InF	ODE	S G	M, U, W	T C M	http://snnap.uth.tmc.edu/
<i>SONN</i>	ODE InF	ODE	G	U, W	M	http://www.lis.huji.ac.il/~litvak/Sonn/sonn.html
<i>SpikeNET</i>	InF		S	U	M	http://www.cnl.salk.edu/~arno/SpikeNET
<i>Surf-Hippo</i>	ODE InF	ODE	C S G	U	T M L	http://www.cnrs-gif.fr/iaf/iaf9/surf-hippo.html
<i>XNBC</i>	ODE InF	ODE	G	U, W	M	http://www.u444.jussieu.fr/xnbc/
<i>BIOQ</i>		ODE MC	G	U, W	M	http://www.lis.huji.ac.il/~litvak/Bioq/bioq.html
<i>CKS</i>		MC	G	M, W, OS/2	T M	http://www.almaden.ibm.com/st/msim/index.html
<i>DBSolve</i>		ODE	C G W	U, W	T	http://websites.ntl.com/~igor.goryanin/
<i>Gepasi</i>		ODE	G	W	T C M	http://www.gepasi.org/
<i>In Silico Cell \$</i>		ODE	G	W	M	http://www.physiome.com/
<i>Jarnac/Indigo</i>		ODE	C S G	W	T M L	http://www.sys-bio.org
<i>KinSim/FitSim</i>		ODE	S G	M, U, W	M	http://www.biochem.wustl.edu/cflab/message.html
<i>MacKinetics</i>		ODE	S	M, W	M	http://members.dca.net/leipold/mk/advert.html
<i>MetaModel</i>		ODE	C	DOS		http://bip.cnrs-mrs.fr/bip_10/modeling.htm
<i>StochSim</i>		MC	G	U, W	M	http://www.zoo.cam.ac.uk/comp-cell/StochSim.html
<i>Vcell</i>	ODE	ODE	G W	M, U, W	M L	http://www.nrcam.uchc.edu/index.html

Packages that simulate the electrical activity of cells are in the top half of the table. Packages that simulate chemical reactions are in the bottom half of the table. Methods abbreviated under Spiking Neurons and Chemical Kinetics: ODE = Ordinary differential equations, MC = Monte Carlo, InF = Integrate-and-fire. Abbreviations under Control, Platform, and Support are identical to those in Table 1.

direction and distance each individual molecule diffuses (Stiles and Bartol, 2001). If two or more molecules happen to collide, another random number is compared to a rate constant to determine whether a chemical reaction occurs. These methods are more accurate than ODEs when well-mixed assumptions are violated, such as when a simulation involves small numbers of molecules per spatial compartment. Monte Carlo methods can also be faster than ODEs when complicated three-dimensional structure requires a large number of compartments to meet the well-mixed assumptions of ODEs. The major disadvantage of Monte Carlo methods is the amount of memory required to track every element of a simulation.

Progression of Complexity

In our experience, users generally prefer to start out using a program through a graphical user interface (GUI), in which the number of choices is restricted and approachable. As they become more proficient, users often dispense with the GUI and begin to interact with the program at the more flexible command-line level or through scripts (lists of commands saved in text files). This allows them to automate common tasks that would otherwise have to be performed step by step. In keeping with this progression, we divide the simulators into three levels of complexity: demonstration pro-

Table 3. General Modeling Tools for Detailed Analysis

Name	Control	Platform	Support	Web Site
<i>CMISS</i>	C S G W	U	T M	http://www.cmiss.org
<i>CONTENT \$</i>	G	U, W	M	http://www.cwi.nl/ftp/CONTENT/
<i>JSIM/XSIM</i>	G	U, W	T C M	http://nsr.bioeng.washington.edu/index.html
<i>Madonna \$</i>	S G	M, W	M	http://www.berkeleymadonna.com/
<i>MatLab \$</i>	C S	M, U, W	C M	http://www.mathworks.com/products/
<i>ModelMaker \$</i>	G	W	T C M	http://www.modelkinetix.com/
<i>SAAM \$</i>	G	W	T M	http://www.saam.com/software/saam2
<i>SABER \$</i>	G	U, W	C M	http://www.analogy.com/products/simulation/
<i>ScoP \$</i>	C S G	U, W, DOS	T C M	http://www.simresinc.com/
<i>SPICE</i>	C S G	U	M	http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/
<i>XPP-AUT</i>	C S G	U, W	M L	http://www.math.pitt.edu/~bard/xpp/xpp.html

Abbreviations under Control, Platform, and Support are identical to those in Table 1.

grams, modeling packages specialized for neurobiology, and generalized modeling tools. Each level has its advantages, as described in more detail below.

Demonstration Programs

Demonstration programs are particularly appropriate for students, for researchers who are approaching modeling for the first time, and for quick explorations of experimental hypotheses. They are limited to a few general examples (see Table 1), although those examples can be rich, with parameters for temperature and other common experimental variables (Friesen and Friesen, 1994; Huguenard and McCormick, 1994). Several demonstration programs use virtual experimental rigs, including electronic components such as amplifiers and analog-to-digital converters, perfusion pumps for drugs, and oscilloscopes; this allows users to become comfortable with using the equipment before proceeding to simulated or real experiments (e.g., *cLabs*, *NerveWorks*). Some programs allow the preset simulations to be modified (e.g., *NeuroSim*, *VClamp/CClamp*). For example, *NeuroSim* allows the teacher to simplify simulations by hiding irrelevant parameters; this feature can also hide the values of parameters so that students have to deduce them experimentally. A drawback for experimenters is that these demonstration programs generally do not include automated tools to analyze their output or to save their output so that third-party software could analyze it (but see GENESIS SIMULATION SYSTEM and NEURON SIMULATION ENVIRONMENT).

Packages Specialized for Simulating Spiking Neurons

A number of specialized packages are available to simulate the electrical activity of neurons and neural circuits. These packages are more flexible than the demonstration programs discussed above. The trade-off for this richness is an increase in complexity of the software used to build models. For example, *SNNAP* uses a hierarchical parameter tree composed of 26 parameter file types, including conductances (voltage-gated or ligand-gated), modulators, and intracellular pools of ions (Ziv, Baxter, and Byrne, 1994). Each parameter file consists of a single equation, chosen from a list of possibilities. Simulations are controlled through the GUI, although the text of the parameter files can be accessed through an ASCII text editor, a faster and more flexible option for experienced users.

Individual cells. Packages vary in the detail with which individual neurons are modeled. The simplest ODE package (*SONN*) models each cell as a single-compartment isopotential sphere, using a fixed set of four differential equations, sufficient to describe the minimal soma capable of bursting behavior. Most other neurosimulators divide a cell into many compartments, each with its own parameters. For example, a dendritic compartment might be electrically passive, whereas an axonal compartment would contain the Na^+ and K^+ conductances necessary to support spiking. Certain packages can automate creation of multicompartmental three-dimensional structures to some extent by importing anatomical files, either with built-in tools (e.g., *Maxsim*, *NeMoSys*, *Surf-Hippo*) or through the use of third-party applications (e.g., *GENESIS*, *NEURON*).

Packages vary in their analytical capabilities, as well. Most packages can run batches of simulations to systematically vary parameters, saving the results in files to be examined by other tools. Some can also optimize model parameters to fit experimental data (e.g., *GENESIS*, *NEURON*, *NeuronC*, *Surf-Hippo*). *Surf-Hippo* runs scripts that detect spikes or other events in the output of a cell. *XNBC* has menu-based tools for time series analysis.

Small circuits. Most packages, whether based on discrete events, ODEs, or Monte Carlo, come with built-in synaptic conductances

that can be used to connect neurons into a circuit. The most common “alpha function” synapse has a maximal conductance, a reversal potential, and an exponential time course (see SYNAPTIC INTERACTIONS). In many packages, synaptic conductances can be modulated by intracellular second messengers to produce various profiles of synaptic plasticity (see the “Chemical Kinetics” column in Table 2). Built-in electrical synapses that simulate gap junctions are relatively uncommon (however, see *NeuronC*, *SNNAP*). *NeuronC* is unusual in that it models the synapse as a series of filters that determine how presynaptic voltage affects postsynaptic voltage to study information transfer across the synapse. In addition to graphs of membrane voltage over time in individual neurons, several packages allow the user to visualize the entire network as a grid of icons in order to examine spatial patterns of activity, such as waves that sweep through the network (e.g., *BioSim*, *Neuro-Imitator*, *XNBC*).

Large networks. Some packages allow for abstraction beyond integrate-and-fire neurons to reduce the number of network elements. For example, a group of neurons whose interactions form an oscillator or a filter function can be represented as a single unit (e.g., *NSL*, *SpikeNET*). In some packages (e.g., *SEE*, *SpikeNET*, *XNBC*), the user can define populations of identical neurons that project topographically to other populations, which reduces effort in wiring up large networks.

Detailed compartmental modeling of many thousands of neurons requires a package optimized for that purpose, usually running on parallel processors, because of the large number of calculations involved (e.g., *GENESIS*, *NEST/SYNOD*, *NEURON*, *Parallel Neurosys*, *SEE*). A program for simulation at even larger scales is *BioPSE*, which models electrical field potentials in biological tissue.

Specialized Biochemical Packages

Many of the packages described above can simulate intracellular processes such as diffusion of ions and second-messenger cascades (see the top half of Table 2). Other packages, designed to study metabolic or genetic networks (see the bottom half of Table 2), have additional analytical capabilities that may benefit neural researchers. For example, some (e.g., *Gepasi*, *Jarnac/Indigo*) are capable of metabolic control analysis. This sensitivity analysis measures the relative control exerted by each enzyme on the system’s fluxes and metabolite concentrations (Wildermuth, 2000). A convenient feature of *BIOQ* allows the user to check closed-form reactions, such as the opening and closing of an ion channel, for violations of the law of conservation of energy.

Most of these biochemical modeling packages can compile a kinetic description of a reaction into differential equations. Some packages incorporate a GUI that allows the user to build a “biochemical network” using icons of molecules and reaction schemes. The equations are then generated by the program and presented for the user’s inspection.

General Modeling Tools

The powerful programs described in this section can potentially be used to simulate any dynamical system (Table 3). However, they are less likely to have built-in model templates relevant to a particular neurobiological system. For example, *SABER* and *SPICE* are electrical circuit simulators that have been used to simulate biological neurons (Bove et al., 1994). *MatLab* is commonly used by connectionist modelers but has no official support for compartmental neuron simulations. With the advent of model databases, this may be less of a restriction on the usefulness of general mod-

eling tools, because individual users of a particular package may have models to share (see DATABASES FOR NEUROSCIENCE).

One advantage of these general tools is their analytical power. For instance, *MatLab* has many functions for analyzing time series data such as spike trains. *XXP-AUT* and *CONTENT* are specialized for analyzing dynamical phenomena, such as nullclines, singularity points, bifurcations, and steady states (Ermentrout, 2002; see PHASE-PLANE ANALYSIS OF NEURAL NETS). Currently, perhaps the best compromise is to analyze models built in more specialized packages with these general tools. Ready examples of this type of synergy between programs are *GENESIS* and the biochemical simulator *Jarnac/Indigo*, which can export their output in *MatLab* format.

Increasing Cooperation among Modelers

Some of the most exciting developments in neurosimulation technology will allow modelers to compare and modify models, verify one another's simulations, and extend models with their own tools. We examine two of these enabling technologies below (see also DATABASES FOR NEUROSCIENCE).

Brokering Agents

One way to coordinate the actions of multiple simulation packages is through software called a brokering agent. For example, a model can be built by using a biophysically detailed neuron model from one simulator, an integrate-and-fire network from a different simulator, and a number of individualized Java or C++ components for visualization or analysis. Each component runs independently, receiving input from the broker and sending back its output.

A brokering agent can also support the distribution of a single model onto multiple processors without the need to modify the model description. This last point is crucial, as some parallel simulation tools (such as *GENESIS*) require the model itself to be a parallel program, which is not a trivial modification. This "transparent parallelism" is possible if the model is specified as a number of entities that communicate using discrete events, a strategy that is employed to some extent by many neurosimulators.

Three brokering agents are currently under development for use in neural simulation. *Bio/Spice* (<http://www.darpa.mil/ipto/research/biocomp/index.html>) and the *Systems Biology Workbench* (http://www.cds.caltech.edu/erato/the_project.html) interface between biochemical simulators. *NEOSIM* (<http://www.neosim.org/>) is designed to work at the level of neuronal networks. An existing example of the possibilities for a brokering agent, *ISYS*, provides interfaces between genomic databases (Siepel et al., 2001).

Open-Source Development

Another approach to collaborative modeling is taken by five cooperating laboratories in producing *NEST/SYNOD* (Diesmann and Gewaltig, in press). In an open-source project, registered users can modify the software. A centralized server automatically updates the software and distributes the latest version to users. The large number of reviewers ensures that bugs will be found and repaired quickly. This is unlike most neurosimulators, for which a single laboratory controls the development of the software, albeit with suggestions from users, and individual users are responsible for their own updates. It remains to be seen whether the open-source approach will spread to other neurosimulator packages.

Discussion

Understanding the brain requires an enormous amount of data about its individual components and their interactions. Because of the sheer volume of data and the nonlinear nature of the interactions, it becomes impossible to understand the system without computational modeling.

No single monolithic simulation package can fulfill all of the diverse requirements of individual neuroscientists doing quantitative simulations. At the same time, developers are challenged to encourage collaboration by increasing interoperability among the many neurosimulators and by building tools to manage the ever-increasing number of models and simulations. As these challenges are met, models that span levels of neural organization, ranging from genetic regulatory networks to large-scale brain structures, will become possible.

Road Map: Implementation and Analysis

Related Reading: Databases for Neuroscience; Neuroinformatics; Perspective on Neuron Model Complexity

References

- Bove, M., Massobrio, G., Martinoia, S., and Grattarola, M., 1994, Realistic simulations of neurons by means of an ad hoc modified version of SPICE, *Biol. Cybern.*, 71(2):137–145. ♦
- Bower, J. M., and Bolouri, H., 2002, *Computational Modeling of Genetic and Biochemical Networks*, Cambridge, MA: MIT Press.
- Cobelli, C., and Foster, D. M., 1998, Compartmental models: Theory and practice using the SAAM II software system, *Adv. Exp. Med. Biol.*, 445:79–101.
- De Schutter, E. (Ed.), 2001, *Computational Neuroscience: Realistic Modeling for Experimentalists*, Boca Raton, FL: CRC Press. ♦
- Diesmann, M., and Gewaltig, M.-O., in press, NEST: An environment for neural systems simulations, in *Forschung und wissenschaftliches Rechnen* (V. Macho, Ed.), Göttingen, Germany: Gesellschaft für Wissenschaftliche Datenverarbeitung.
- Ermentrout, B., 2002, *Simulating, Analyzing and Animating Dynamical Systems: a Guide to XPPAUT for Researchers and Students*, Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Friesen, W. O., and Friesen, J. A., 1994, *NeuroDynamix: Computer Models for Neurophysiology*, New York: Oxford University Press.
- Hammarlund, P., and Ekeberg, O., 1998, Large neural network simulations on multiple hardware platforms, *J. Comput. Neurosci.*, 5(4):443–459. ♦
- Huguenard, J., and McCormick, D., 1994, *Electrophysiology of the Neuron: An Interactive Tutorial*, Oxford, Engl: Oxford University Press.
- Koch, C., and Segev, I. (Eds.), 1998, *Methods in Neuronal Modeling: From Ions to Networks*, Cambridge, MA: MIT Press.
- Murre, J. M. J., 1995, Neurosimulators, in *Handbook of Brain Theory and Neural Networks* (M. A. Arbib, Ed.), Cambridge, MA: MIT Press, pp. 634–639.
- Siepel, A., Farmer, A., Tolopko, A., Zhuang, M., Mendes, P., Beavis, W., and Sobral, B., 2001, ISYS: A decentralized, component-based approach to the integration of heterogeneous bioinformatics resources, *Bioinformatics*, 17(1):83–94.
- Skrzypek, J. (Ed.), 1994, *Neural Network Simulation Environments*, Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Stiles, J. R., and Bartol, T. M., 2001, Monte Carlo methods for simulating realistic synaptic microphysiology using Mcell, in *Computational Neuroscience: Realistic Modeling for Experimentalists* (E. De Schutter, Ed.), Boca Raton, FL: CRC Press, pp. 87–127.
- Wildermuth, M. C., 2000, Metabolic control analysis: Biological applications and insights, *Genome Biol.*, 1(6):REVIEWS1031. ♦
- Ziv, I., Baxter, D. A., and Byrne, J. H., 1994, Simulator for neural networks and action potentials: Description and application, *J. Neurophysiol.*, 71(1):294–308.

NMDA Receptors: Synaptic, Cellular, and Network Models

Michel Baudry, Jean-Marie Bouteiller, Jim-Shih Liaw, and Theodore W. Berger

Introduction

NMDA receptors are subtypes of receptors for the excitatory neurotransmitter, glutamate, that are selectively activated by the agonist *N*-methyl-D-aspartate (NMDA), a glutamate analog. They are involved in diverse physiological as well as pathological processes such as visual perception, motor pattern generation, learning and memory, and epilepsy- or stroke-induced neuronal damage (Dingledine et al., 1999).

NMDA receptors have three unique properties that distinguish them from other ligand-gated channels (channels activated by molecules that bind to them). First, they mediate a relatively "slow" excitatory postsynaptic potential (EPSP). Second, their activation requires not only the binding of an agonist (a molecule that mimics the effect of the endogenous neurotransmitter), but also the depolarization of the postsynaptic membrane to remove the voltage-dependent blockade of the channel by Mg^{2+} ions. Thus, NMDA receptors act as coincidence-detectors of presynaptic and postsynaptic activity. Third, NMDA receptors are 10 times more permeable to Ca^{2+} ions than to Na^{+} or K^{+} . They often are colocalized with other receptor channels that conduct "fast" EPSPs (e.g., the α -amino-3-hydroxy-5-methyl-4-isoxazole propionic acid, or AMPA, receptor). The interactions between the slow NMDA-mediated and fast AMPA-mediated currents provide the basis for a range of interesting dynamic properties, which contribute to a diversity of neuronal processes.

NMDA receptors have attracted a lot of interest in neuroscience because of their role in learning and memory. Their properties of coincidence detectors make them an ideal molecular device for producing Hebbian synapses, that is, synapses whose strength is modified depending on the correlation of presynaptic and postsynaptic activity. Furthermore, the influx of Ca^{2+} ions through the NMDA receptor channel triggers cascades of molecular processes that lead to various forms of synaptic plasticity, including short-term potentiation, long-term potentiation, and long-term depression (Malenka and Nicoll, 1999), as well as to pathological processes, including neuronal death. As a result, they have been a subject of intense investigation by both experimentalists interested in understanding the features of the protein that result in its functional properties and theorists/modelers attempting to incorporate NMDA receptors into models of synapses, neurons, or circuitries. We will briefly review here data related to the biological characteristics of NMDA receptors and to models that have been used to describe their function in isolated membrane patches, in neurons, and in complex circuits.

Biological Characteristics of NMDA Receptors

Cloning techniques have provided evidence for the existence of a multiplicity of NMDA receptor subunits conferring distinct properties for the NMDA receptors (see Cull-Candy, Brickley, and Farrant, 2001, for a review). The NR1 subunits is required in combination with at least one member of the NR2 family (four distinct genes), and NR1/NR2 complexes may also co-assemble with a member of the NR3 family (two genes). NMDA receptor activation requires not only the presence of glutamate, but also the presence of another amino acid, glycine, which therefore has been called a co-agonist and binds to NR1 subunits, while glutamate binds to NR2 subunits. A number of other binding sites for Mg^{++} , Zn^{++} ,

and polyamines are also present on the subunits, and their occupation modifies receptor properties. It is generally assumed that the NMDA receptors are multimeric entities (i.e., composed of several subunits), possibly pentameric proteins by analogy with the acetylcholine nicotinic receptor. Moreover, as different combinations of receptor subunits produce receptors with different physiological and pharmacological properties, it is likely that several functional classes of NMDA receptors exist in adult neurons, and a major challenge remains to determine their cellular distributions and the mechanisms regulating subunit expression and receptor assembly and turnover.

A major avenue of research over the last five years has focused on the mechanisms underlying NMDA receptor targeting and anchoring in postsynaptic sites. Thus a very large complex of proteins, including receptor molecules, PSD-95, cytoskeletal proteins, and associated protein kinases has been shown to be localized in postsynaptic densities and to play critical roles in mediating multiple effects of NMDA receptor activation (Kennedy, 1998).

Kinetic Models of NMDA Receptors

The behavior of a receptor can be described by the kinetic scheme of its transition between various discrete states representing the occupancy of the different binding sites and the functional states of the channel (a simple example is $A + R \leftrightarrow AR \leftrightarrow AR^*$, where A, R, AR, and AR^* represent agonist, receptor, and receptor-agonist complex in closed state and open state, respectively; additional desensitized states of the receptors correspond to receptor occupied by agonists but with closed channels). Several kinetic models of the NMDA receptors have been developed to study various aspects of the nature of the receptor and the dynamics of its behavior.

Glycine Binding

Glycine is a necessary co-agonist of the receptor, and various models of NMDA receptors have been developed to study the number of glycine binding sites and the interaction between glutamate and glycine on the receptors. Recordings in outside-out patches from cultured hippocampal neurons were used to measure the rate constants for agonist binding, open/close transitions, and desensitization of the NMDA receptor. The measured rate constants were used in models assuming one, two, or three agonist binding sites. For both glutamate and glycine binding, a two-site model provided a superior fit for the time course of NMDA channel activation. Desensitization was also studied in excised outside-out patches and was interpreted as an interaction between the glutamate and the glycine site such that glycine binding produced a decreased affinity of glutamate for its binding site (Lester, Tong, and Jahr, 1993). Note that binding experiments also suggest a strong positive allosteric effect between glutamate and glycine binding (Marvizon and Baudry, 1993).

Number of States of the NMDA Receptor

Jahr and Stevens (1990) developed a model of NMDA receptor-channel kinetics to address the issue of the number of states of the receptor. They assumed that the NMDA receptor exists in three states: closed (C), open (O), and blocked (B). The kinetic behavior

of the NMDA receptor-channel was characterized by four experimentally measured quantities: open time (T_o), interruption time (T_i), number of interruptions (N), and burst length (T_b). The predictions made by the model failed to match several key experimental data, and these shortcomings suggested an extension to include a second blocked state in addition to the Mg^{2+} block. The four-state model could describe NMDA receptor behavior in all conditions except the low-amplitude, second-exponential component in T_o , which occurred in low Mg^{2+} concentrations and positive voltage. A theory based on the four-state model postulates that the interruptions could be the result of a voltage-dependent conformational change, which could be facilitated by the binding of Mg ions to some sites on the NMDA receptor.

NMDA Receptors in Models of Neurons

NMDA receptor models have been incorporated in several models of synapses/neurons developed to answer questions related to short-term as well as long-term synaptic plasticity and to mechanisms of synaptic integration. As was mentioned in the introduction, a major focus is on the role of the voltage-dependent Mg^{2+} blockade, which gives rise to the property of coincidence detection and Ca^{2+} influx through the NMDA receptor, in initiating the molecular events leading to changes in synaptic strength. Furthermore, the interplay between the slow NMDA-mediated and fast AMPA-mediated currents leads to important features in the timing of synaptic inputs required to induce long-term potentiation (LTP)

Short-Term Synaptic Plasticity

Pongrácz et al. (1992) used a compartmental model to study short-term changes in excitability of a hippocampal pyramidal neuron. The pyramidal neuron was represented by five compartments for the apical and basal dendrites, one compartment for the soma, and a layer representing extracellular K^+ concentration. The model included intrinsic membrane conductances and excitatory (NMDA and AMPA) and inhibitory ($GABA_A$ and $GABA_B$) synaptic conductances distributed in the apical dendrite compartment. The voltage dependency of NMDA receptors (due to the Mg^{2+} blockade of the channel) resulted in a weak conductance by single stimulation, but the conductance increased with increasing number and intensity of repeated stimulation. The model suggested that such cumulative activation of NMDA-mediated synaptic conductances contributed to the frequency-dependent EPSP potentiation, a form of short-term plasticity, of hippocampal neurons.

Long-Term Synaptic Plasticity

Holmes and Levy (1990) developed a compartmental model of a granule cell from the dentate gyrus to study the role of Ca^{2+} and the subsequent biochemical events involved in triggering LTP. In particular, they were interested in understanding the mechanisms amplifying the calcium signal and the relative timing of presynaptic and postsynaptic activation. An 11-compartment model was constructed to represent a spine and a small patch of the neighboring dendrite. Calcium dynamics, including Ca^{2+} influx, buffering, pumping, and diffusion were computed over this domain. One glutamate binding site and the voltage-dependent Mg^{2+} block were included in the NMDA receptor kinetics. The amplitude of the peak intracellular-free Ca^{2+} concentration was regarded as the critical parameter for the induction of LTP at a particular synapse. When few synapses were activated, Ca^{2+} influx was small, even with high input frequency. When a large number of synapses were activated simultaneously, a steep rise in Ca^{2+} influx was seen with increasing frequency due to the voltage dependency of the NMDA-mediated conductance. However, total Ca^{2+} influx never increased by more

than fourfold, which is too small an amount to account for the selective induction of LTP. The threefold to fourfold increase could be amplified 20- to 30-fold by transient saturation of the fast Ca^{2+} buffering system. When a weak input was paired with a strong one, the largest increase in peak $[Ca^{2+}]_i$ was seen in cases in which the weak stimulation preceded the strong input by 1–8 ms, because of the slow rate constant of NMDA receptor kinetics. De Schutter and Bower (1993) extended this model to evaluate the effect of Ca^{2+} permeability of the NMDA receptor channel. Maximum amplification of $[Ca^{2+}]_i$ was obtained at permeability close to values reported in the literature and decreased significantly when permeability was reduced by more than 50%. Furthermore, simulations showed that $[Ca^{2+}]_i$ was up to 80% higher at distal spines than at proximal ones.

Synaptic Integration

In addition to synaptic plasticity, NMDA receptors are also involved in the integration of spatiotemporal patterns of inputs by a neuron. Fox and Daw (1992) developed an electrical model of a neuron in area 17 of the visual cortex to study neuronal responses to visual stimuli. The model was composed of two compartments: a somatic component and a dendritic compartment with NMDA and AMPA receptors. The model showed that instead of switching on only at a higher level of contrast, the NMDA receptor-mediated conductance contributed to the response to visual stimuli in a graded fashion, all the way from near threshold to saturation. This property of the NMDA receptor could not be accounted for solely by its voltage dependency. In addition, the higher affinity (binding rate) of NMDA receptors for glutamate, in comparison to the non-NMDA receptor, was involved.

A compartmental model was developed to study the integrative behavior of a complex dendritic tree with particular focus on the role of NMDA receptors in the generation of neuronal responses (Mel, 1993). An anatomically characterized cortical pyramidal cell was represented by a model consisting of 903 electrically coupled compartments. A glutamatergic synapse was placed on the distal end of a spine containing both AMPA and NMDA receptors. The major finding from the simulation of the model was that when the NMDA receptors constituted a large portion of the synapse, the neuron responded preferentially to spatially clustered, rather than randomly distributed activated synapses. This was due to the voltage dependency of the NMDA receptors that were more effective when activated in group than individually. As a result of activity-dependent synaptic modifications, synapses on a dendritic tree were organized in such a way that stimuli that activated a similar set of synapses as those activated by patterns presented during the learning period had a higher probability of eliciting a neuronal response. Therefore, manipulating the spatial ordering of afferent activation of a dendritic tree provides a biological strategy for storing and classifying patterned information.

Metaplasticity

Considerable attention has been devoted to understanding mechanisms that could account for changes in rate of learning, a phenomenon referred to as metaplasticity. Several models of neurons or of networks of neurons have introduced various parameters to incorporate metaplasticity. The most popular model is the so-called modification threshold or sliding rule introduced by Bienenstock, Cooper, and Munro (1982). In their model of synaptic plasticity, activity-dependent changes in NMDA receptors could account for such a rule.

NMDA Receptors in Models of Neuronal Circuits

The role of NMDA receptors in neuronal networks has been studied in various systems. In this section, we briefly review their roles in

generating evoked field potentials, oscillatory or epileptiform activity in neuronal networks, in working memory and learning of temporal sequences. In all these cases, the interactions between excitatory and inhibitory synapses shape the dynamics of the neural network.

NMDA Receptors and Evoked Field Potentials

A recent compartment model of a hippocampal network comprising pyramidal neurons, inhibitory interneurons, and feedback and feed-forward inhibition explored the contribution of NMDA receptor-mediated synaptic currents to evoked field potentials (Wang et al., *in press*). As predicted from the kinetics of the receptors, the NMDA receptors contribute significantly to the late phase of the evoked potentials, and their influence becomes more important with repeated stimulation as in paired-pulse or burst of stimulation. This effect is illustrated in Figure 1, which clearly indicates that NMDA receptors contribute significantly to burst-evoked synaptic depolarization. Likewise, compartment model simulation indicated that the amplitude of NMDA receptor-mediated calcium signals is greatly increased with increased frequency of stimulation. Furthermore, several experimental and model studies have now shown that the presence of NMDA receptors at synapses could account for low-pass temporal frequency tuning in several sensory pathways (Krukowski and Miller, 2001).

Role of NMDA Receptors in Oscillators

A network of interneurons conformed to experimentally identified cell types was constructed to simulate the spinal locomotor pattern generation in lamprey (Träven et al., 1993). Excitatory synapses displayed both NMDA and AMPA receptors, while the inhibitory

synaptic transmission was glycinergic and mediated by chloride ions. The NMDA receptor current was modeled as a product of channel conductance, the difference of the membrane potential and the equilibrium potential and a state variable, which accounted for the voltage-dependent Mg^{2+} block of the channel. Oscillatory bursts could be evoked in a postsynaptic cell driven by NMDA receptor-mediated synaptic currents, but the presynaptic neuron had little effect on oscillation frequency. The presynaptic control of oscillation frequency increased when AMPA receptors were added. A continuous range of network burst rate could be produced by the NMDA and AMPA receptor-mediated conductances. The simulations suggested that spinal locomotor network could be modulated by controlling the balance between NMDA and AMPA receptor-mediated synaptic input. The NMDA receptor-containing synapses mainly served to stabilize the rhythmic motor output, whereas the AMPA receptor-containing synapses provided direct phasic control of the burst pattern.

NMDA Receptors and Epileptiform Activity

Traub, Miles, and Jefferys (1993) developed a computer model of hippocampal CA3 region consisting of pyramidal neurons and inhibitory interneurons. Each pyramidal neuron was composed of 19 compartments with six voltage-dependent ionic conductances. Each pyramidal neuron was randomly connected to 20 other pyramidal neurons via excitatory (NMDA and AMPA) receptors and to 20 interneurons via inhibitory ($GABA_A$ and $GABA_B$) receptors. The computation of NMDA receptor-mediated current involved a scaling factor, a synaptic conductance term with a slow decay time constant, and a term representing the voltage-dependent Mg^{2+} blockade. The simulation suggested the following conditions for the occurrence of population oscillations: (1) The strength of excitatory synapses falls within a limited range, (2) the after-hyperpolarization conductance is significantly reduced, (3) the inhibitory postsynaptic potentials are blocked, and (4) the apical dendrites of the pyramidal neurons are depolarized. The NMDA receptor conductance was not necessary for the population oscillation. The model generated synchronized population bursts that resemble experimental data obtained from hippocampal slices perfused with a $GABA_A$ receptor blocker and predicted that dendritic calcium spikes occurred during each secondary burst generated by the AMPA receptor current. However, with sufficiently high NMDA receptor conductance, synchronized bursts could occur in the absence of AMPA receptor current.

NMDA Receptors and Memory

Experimental evidence has strongly implicated NMDA receptors in various forms of learning and memory. Although it is generally assumed that this is a consequence of the role of NMDA receptors in activity-dependent changes in synaptic transmission in various networks, the specific roles of NMDA receptors in memory has also been more formally evaluated in various simulations of network activity and dynamics. In particular, several groups have shown that the properties of NMDA receptors are well suited for learning of sequences of information. This is particular true for networks exhibiting both theta and gamma activities, allowing recall of stored sequence by presentation of the initial element of the sequence (Jensen and Lisman, 1996).

Another interesting role of NMDA receptors has been recently proposed to account for the persistent activity in prefrontal cortex, which is assumed to be the basis for working memory (Compte et al., 2000). In this case, stable and persistent activity could be observed when recurrent synaptic excitation was mediated principally by NMDA receptor-mediated currents.

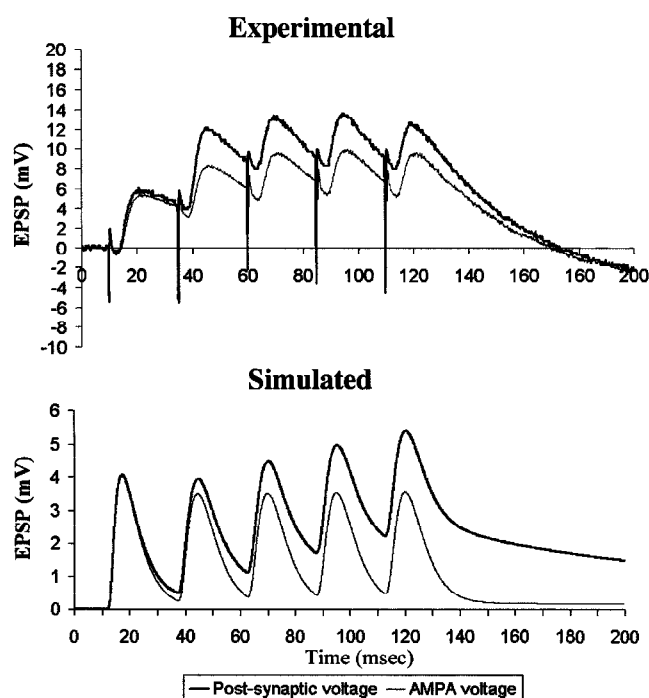


Figure 1. Contribution of NMDA receptors to postsynaptic depolarization elicited by a train of five pulses (40 Hz) in field CA1 (*Top*) and in a simulated computational model (*Bottom*). In both cases, the top trace represents total depolarization, and the bottom trace represents depolarization in the presence of the NMDA receptor blocker, APV.

Discussion

This review of the various models of NMDA receptors at the synaptic, cellular, and network levels illustrates that the three unique properties of these receptors (i.e., slow conductance, voltage and transmitter dependency, and calcium permeability) provide the basis for their involvement in a large variety of fundamental dynamic properties of synaptic transmission, including not only short-term and long-term plasticity at individual synapses, but also complex network properties such as synaptic integration, motor pattern generation, and epileptiform activity.

The review also indicates the usefulness of this approach to investigate the contribution of different characteristics of the receptors at the functional level. In particular, the recent suggestion that metaplasticity could be accounted for by changes in NMDA receptor subunit composition and function needs to be emphasized. A number of issues still remain unresolved, in part because of the limited knowledge concerning the exact number of binding sites for the various effectors of the receptors, the mechanisms underlying desensitization, and the anatomical distribution of the different types of receptors. Interestingly, developmental switches in the subunit composition of NMDA receptors have been shown to dramatically alter network properties and information processing in these networks. Therefore, as the understanding of the characteristics and properties of NMDA receptors continue to be resolved in greater details, new models will need to be generated to capture these properties and to evaluate their contributions to the computational properties of individual neurons as well as to complex circuitries.

Road Map: Neural Plasticity

Related Reading: Biophysical Mosaic of the Neuron; Hebbian Synaptic Plasticity; Temporal Dynamics of Biological Synapses; Ion Channels: Keys to Neuronal Specialization

References

- Bienenstock, E. L., Cooper, L. N., and Munro, P. W., 1982, Theory for the development of neuron selectivity: Orientation, specificity, and binocular interaction in visual cortex, *J. Neurosci.*, 2:32–48.
- Compte, A., Brunel, N., Goldman-Rakic, P. S., and Wang, X. J., 2000, Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model, *Cereb. Cortex*, 10:910–923.
- Cull-Candy, S., Brickley, S., and Farrant, M., 2001, NMDA receptor subunits: Diversity, development and disease, *Curr. Opin. Neurobiol.*, 11:327–335.
- De Schutter, E., and Bower, J. M., 1993, Sensitivity of synaptic plasticity to the Ca^{2+} permeability of NMDA channels: A model of long-term potentiation in hippocampal neurons, *Neural Computation*, 5:681–694.
- Dingledine, R., Borges, K., Bowie, D., and Traynelis, S. F., 1999, The glutamate receptor ion channels, *Pharmacol. Rev.*, 51:7–61.
- Fox, K., and Daw, N., 1992, A model for the action of NMDA conductances in the visual cortex, *Neural Comput.*, 4:59–83.
- Holmes, W. R., and Levy, W., 1990, Insights into associative long-term potentiation from computational models of NMDA receptor-mediated calcium influx and intracellular calcium concentration changes, *J. Neurophysiol.*, 63:1148–1168.
- Jahr, C. E., and Stevens, C. F., 1990, A quantitative description of NMDA receptor-channel kinetic behavior, *J. Neurosci.*, 10:1830–1837.
- Jensen, O., and Lisman, J. E., 1996, Theta/gamma networks with slow NMDA channels learn sequences and encode episodic memory: Role of NMDA channels in recall, *Learn. Mem.*, 3:264–278.
- Kennedy, M. B., 1998, Signal transduction molecules at the glutamatergic postsynaptic membrane, *Brain Res. Rev.*, 26:243–257.
- Krukowski, A. E., and Miller U. D., 2002, Thalamo-cortical NMDA conductances and intracortical inhibition can explain cortical temporal tuning, *Nature Neurosci.*, 4:429–430.
- Lester, R. J. A., Tong, G., and Jahr, C. E., 1993, Interaction between the glycine and glutamate binding sites of the NMDA receptor, *J. Neurosci.*, 17:1088–1098.
- Malenka, R. C., and Nicoll, R. A., 1999, Long-term potentiation: A decade of progress?, *Science*, 285:1870–1874.
- Marvizon, J. C., and Baudry, M., 1993, Receptor activation by two agonists: Analysis by nonlinear regression and application to *N*-Methyl-D-Aspartate receptors, *Anal. Biochem.*, 213:3–11.
- Mel, B. W., 1993, NMDA-based pattern discrimination in a modeled cortical neuron, *Neural Comput.*, 4:502–516.
- Pongrácz, F., Poolos, N. P., Kocsis, J. D., and Shepherd, G. M., 1992, A model of NMDA receptor-mediated activity in dendrites of hippocampal CA1 pyramidal neurons, *J. Neurophysiol.*, 68:2248–2259.
- Traub, R. D., Miles, R., and Jefferys, J. G. R., 1993, Synaptic and intrinsic conductances shape picrotoxin-induced synchronized after-discharges in the guinea-pig hippocampal slice, *J. Physiol.*, 461:525–547.
- Träven, H., Brodin, L., Lansner, A., Ekeberg, Ö., Wallén, P., and Grillner, S., 1993, Computer simulations of NMDA and non-NMDA receptor-mediated synaptic drive: Sensory and supraspinal modulation of neurons and small network, *J. Neurophysiol.*, 70:695–709.
- Wang, Z., Song, D., and Berger, T. W., 2002, Contribution of NMDA receptor channels to the expression of LTP in hippocampal dentate gyrus, *Hippocampus* (in press).

NSL Neural Simulation Language

Alfredo Weitzenfeld

Introduction

Neural simulation plays an essential role in understanding the brain. While many neural simulators exist today (see NEUROSIMULATION: TOOLS AND RESOURCES for a listing of the most important ones), design considerations can be quite different. For example, systems supporting very detailed neural elements can simulate only a few neurons at a time (see NEURON SIMULATION ENVIRONMENT and GENESIS SIMULATION SYSTEM), while systems supporting coarser elements can usually simulate larger neural populations. In this article, we describe the Neural Simulation Language (NSL) (Weitzenfeld, Arbib, and Alexander, 2002), an *object-oriented* system (Wegner, 1990) primarily designed to support simulation of large neural networks. The system addresses the needs of a wide range of users, from novice users requiring friendly

user interfaces to advanced users requiring advanced programming and integration to other systems. Two versions of the system exist today, one in Java (Gosling et al., 2000) and the other in C++ (Stroustrup, 2000). Both of these can run on a wide range of computer platforms, making the system quite independent from the actual computing environment.

Modularity in Neural Systems

A particular aspect that distinguishes NSL from comparable simulators is its special focus on *modularity*, a well-known software development strategy in dealing with large and complex systems. As neural models become large and complex, they become hard to manage. Moreover, modularization of biological neural networks

is further motivated by taking into consideration the way we analyze the brain as a set of different brain regions, as seen by the example shown in Figure 1.

The general methodology for understanding a complex neural system involves two basic approaches. One is to focus on some particular brain region or module and carry out studies of that region in detail. The other is to step back and look at higher levels of organization in which the details of particular modules are hidden. Full understanding comes as we cycle back and forth between different levels of detail in analyzing different subsystems, sometimes simulating modules in isolation, at other times designing computer experiments that help us follow the dynamics of the interactions between the various modules.

Modeling in NSL

There are two ways to describe a model in NSL: (1) by direct programming in NSLM, the NSL (compiled) modeling language, and (2) by using the Schematic Capture System (SCS), a visual programming interface to NSLM supporting the description of module assemblages. In general, NSL supports the two levels of modeling, *modules* and *neural networks*, as described next.

Modules

Modules in NSL are hierarchical structures organized in a tree fashion having a root module, the *model*, and multiple levels of *module assemblages*. Modules may be implemented in different ways and independently from each other in a top-down and a bottom-up fashion, an important benefit from modular design. In particular, *neural* modules are implemented with neural networks, corresponding to leaves on a tree. In general, the external interface to a module is

described by a set of unidirectional input and output *data ports*, representing module entry or exit points, where data are sent or received, usually in the form of numerical values with varying dimension, that is, a single scalar, a one-dimensional array of values (*vector*), a two-dimensional array (*matrix*), or higher ones. To communicate, modules require interconnections among ports belonging to different modules. The following is sample NSLM code describing module assemblages:

```
nsIModel Model ()
{
    private StimulusModule stimulus ();
    private MainModule main ();
    private OutputModule output ();
    public void makeConn () {
        nsIConnect (stimulus.sout, main.in);
        nsIConnect (stimulus.sout, output.sin);
        nsIConnect (main.out, output.oin);
    }
}
```

The description is analogous to a class specification in object-oriented programming. The attribute section describes a three-module assemblage consisting of a “stimulus,” “main,” and “output” modules, while the *makeConn* method specifies module interconnections using the *nsIConnect* statement (see Weitzenfeld et al., 2002, for a more extensive description of all NSLM commands.) This sample NSLM code could be automatically generated from SCS as well. Figure 2 shows sample schematics for a module assemblage within a higher-level module.

Neural Networks

Modules representing brain regions can be anatomically or physiologically divided to obtain *neural modules*, modules described by

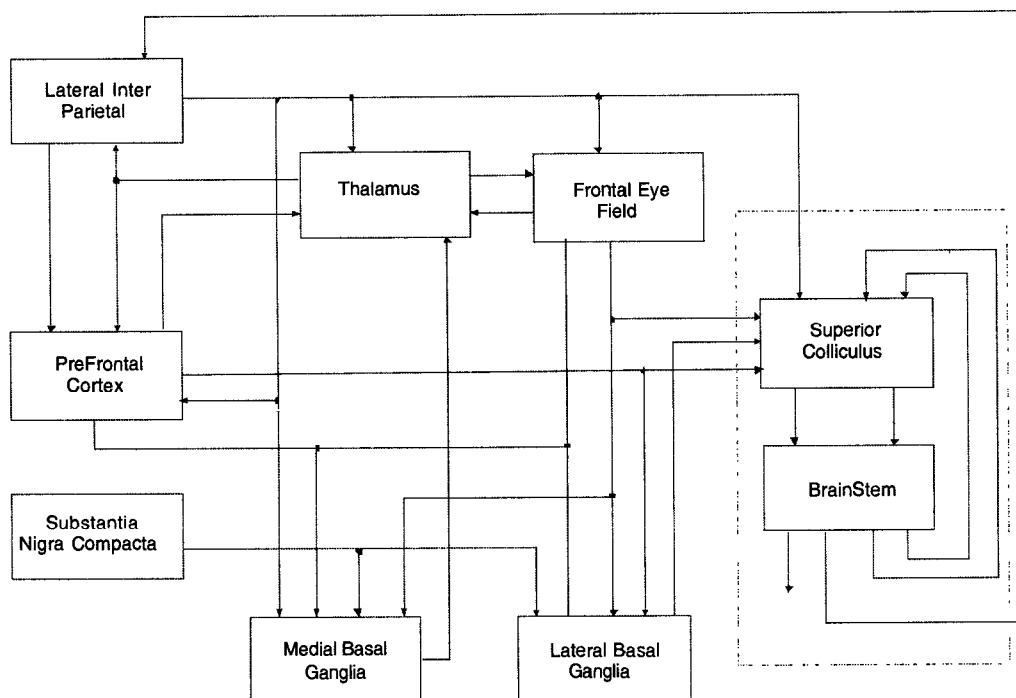


Figure 1. The smaller outlined diagram shows a basic model for control of eye movements consisting of a superior colliculus {XE “Superior Colliculus”} (sc) and brainstem {XE “Brainstem”} modules, each representing a single brain region, responsible for generation of saccades (see COLLICULAR VISUOMOTOR TRANSFORMATIONS FOR GAZE CONTROL). As an ex-

ample of the benefits of modularization, the SC and Brainstem modules can be embedded into the much larger and far more complex model of interacting brain regions, such as the Crowley-Arbib model of BASAL GANGLIA (Crowley, Oztop, and Mármol, 2002).

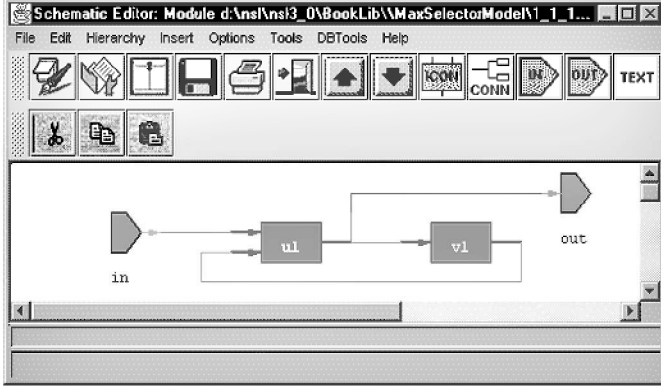


Figure 2. The window shows the Schematic Capture System (SCS) view of the schematics of a sample module consisting of a two-module assemblage and two data ports. Modules are represented by rectangles, while entry (left) and exit (right) ports are represented by pentagon-shaped icons.

neural arrays. To model a complete neural network, it is necessary to describe (1) the particular neuron model, that is, the desired neural level of detail, (2) the neurons making up the network, (3) the set of interconnections among neurons, and (4) network parameters, such as inputs and connection weights. Without disregarding the importance of other neural models, we focus here on the *leaky integrator* (Arbib, 1989) neuron model, a single-compartment neuron having one output and many inputs. The internal state of the neuron is described by a single scalar quantity, its membrane potential mp , which depends on the neuron's inputs and past history. The output is described by another single scalar quantity, its firing rate mf , and may serve as input to multiple neurons, including itself. As the input to a neuron varies, the membrane potential and firing rate vary as well.

In NSL, two numerical structures (NslDouble0 data type) are required to represent such a neuron, one corresponding to the membrane potential and the other one to its firing rate:

```
private NslDouble0 mf ();
private NslDouble0 mp ();
```

In many cases, we may want the value of mf to be communicated to other modules. If such is the case, the declaration for mf should be modified from a private variable to a public output port (note the *Dout* keyword):

```
public NslDoutDouble0 mf();
```

The *membrane potential* for mp is described by a first-order differential equation with dependence on its previous history and input s_m :

$$\tau_m \frac{dmp(t)}{dt} = f(s_m, mp, t)$$

Variable τ_m represents the time constant, while the choice of f defines the particular neural model utilized. The *leaky integrator* model is described by $f(s_m, mp, t) = -mp(t) + s_m(t)$, or

$$\tau_m \frac{dmp(t)}{dt} = -mp(t) + s_m(t)$$

In addition to the membrane potential and firing rate descriptions, we also need to specify the input to the neuron, s_m , internal to the module or obtained from another module. In the latter case, input s_m would be specified as an input port (note the “Din” keyword):

```
public NslDinDouble0 sm();
```

where sm holds a weighted spatial summation of all input to the corresponding neuron.

While neural networks are continuous in their nature, their simulated state is approximated by discrete-time computations. For this reason, we must specify an integration or approximation method to generate as faithfully as possible the corresponding neural state. The dynamics for mp are described by the following statement:

$$mp = \text{nslDiff}(mp, \tau_m, -mp + sm);$$

Function *nslDiff* defines a first-degree differential equation equal to “ $-mp + sm$ ” as described by the leaky integrator model. Different methods can be used to approximate the differential equation, such as Euler and Runge-Kutta. The choice of method may affect both the computation time and its precision. The specific method to use is chosen during simulation and not as part of the model architecture.

The firing rate mf , the output of the neuron, is obtained by applying a *threshold*, typically a *ramp*, *step*, *saturation*, or *sigmoidal* function, to the neuron's membrane potential:

$$mf(t) = \sigma(mp(t))$$

where σ is usually a nonlinear function.

For example, if σ is set to a *step* threshold function, the equation for the firing rate mf would be described by

$$mf = \text{nslStep}(mp);$$

where *nslStep* is the corresponding NSL *step* threshold function.

The previous definition specifies a single neuron without any interconnections. An actual neural network is made of a number of interconnected neurons in which the output of one neuron serves as input to the others. In the leaky integrator neural model, interconnections are very simple structures. On the other hand, *synapses*, the links among neurons, are—in biological systems—complex electrochemical systems and may be modeled in exquisite detail. However, many models have succeeded with a very simple synaptic model in which each synapse carries a connection weight that describes how neurons affect each other. The most common formula for the input sv to a neuron v is given by

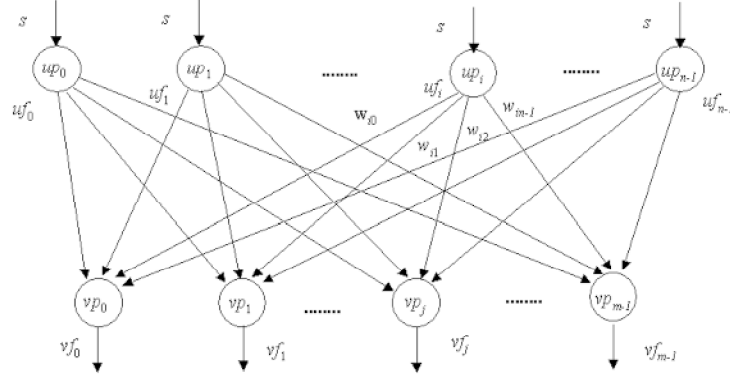
$$sv_j = \sum_{i=0}^{n-1} w_{ji} uf_i$$

where uf_i is the firing of neuron u_i whose output is connected to the j th input line of neuron v_j and w_{ji} is the weight for that link, as shown in Figure 3 (up and vp are analogous to mp , while uf and vf are analogous to mf).

Expanding the summation, input to neuron v_j (identified by its corresponding membrane potential vp_j) is given by sv_j , which is defined as

$$sv_j = w_{j0} uf_0 + w_{j1} uf_1 + w_{j2} uf_2 + \cdots + w_{jn-1} uf_{n-1}$$

Figure 3. The diagram shows a sample two-layer fully connected neural organization (see BACKPROPAGATION: GENERAL PRINCIPLES for an example of networks using such architectures). Each neuron is described by a single compartment represented by a value up or vp , its membrane potential, and a value uf or vf , to its firing, the output from the neuron. Input to the first neural layer is represented by s . Additionally, weights w have been added to the different connections.



While module interconnections are specified in NSL via a *nslConnect* method call, doing this with neurons would in general be prohibitively expensive, considering that there may be thousands or millions of neurons and even more connections in a single neural network. Instead, we use mathematical expressions similar to those used for their representation. For example, the input to neuron v_j , represented by sv_j , would be the sum for all outputs of neuron uf_i multiplied (using the $*$ operator) by connection weight w_{ji} , correspondingly, as shown next:

$$sv_j = w_{j0} * uf_0 + w_{j1} * uf_1 + w_{j2} * uf_2 + \dots;$$

Note that there exist m such equations in the network. We could describe each membrane potential and firing rate individually, or else we could make all u_i and v_j neuron vector structures. The first approach would be very long, inefficient, and prone to typing errors; therefore, we present the second approach, using *neuron arrays* and *connection masks* representing spatial arrangements among homogeneous neurons and their connections, respectively. We consider uf_i the output from a single neuron in an array of neurons and sv_j the input to a single neuron in another array of neurons.

If mask w_{jk} (for $-d \leq k \leq d$) represents the synaptic weights from the uf_{j+k} (for $-d \leq k \leq d$) elements to v_j , for every j , we then have

$$sv_j = \sum_{k=-d}^d w_{jk} uf_{j+k}$$

where the same mask w is applied to the output of each neuron uf_{i+k} to obtain input sv_j . In NSL, the *convolution* operation is described by a single symbol '@':

$$sv = w @ uf;$$

This kind of representation results in great conciseness, an important concern in working with large numbers of interconnected neurons. Note that this is possible as long as connections are regular. Otherwise, single neurons would still need to be connected separately, one by one. This also suggests that the operation is best defined when the number of v and u neurons matches, although a nonmatching number of units can be processed by using a more complex notation.

Simulation in NSL

Simulation involves interactively specifying aspects of the model that tend to change, in particular parameter values, input patterns, simulation control, and visualization. It is important not only to design a good model, but also to design good graphical interfaces, both input and output. In terms of input, NSL offers a number of

approaches: (1) by interactively writing code in NSLS, the NSL (interpreted) scripting language, (2) by loading NSLS scripts stored in files, and (3) by designing custom input interfaces. In terms of output, NSL enables the user to specify various forms of graphical and textual output, including temporal and spatial 2D and 3D graphics (see Weitzenfeld et al., 2002, for input and output visualization examples as well as more extensive description of NSLS commands).

Discussion

In this article we have overviewed modeling and simulation using NSL, a system primarily designed to simulate modular neural systems, both biological and artificial (see BACKPROPAGATION: GENERAL PRINCIPLES for an example of artificial neural networks). The underlying NSL computational model is based on the Abstract Schema Language (ASL) (Weitzenfeld, 1993) inspired by the work on SCHEMA THEORY (q.v.), on actors (Agha, 1986), and more generally on object-based concurrent programming (Yonezawa and Tokoro, 1987).

There are a number of issues worth discussing from our experience with NSL. While user interactivity plays an essential role while creating and testing new neural models, as model becomes more stable, simulation efficiency becomes a primary concern. This is a very important issue if we consider that neural network execution can consume extensive amounts of processing time, possibly hours or even days depending on the size and architecture of the network. For example, we have processed in NSL "simple" biological network, such as the retina model (see RETINA) involving 10,000 neurons, consuming just a few seconds. On the other hand, a more complex network such as the one described in Figure 1 could take several minutes if implemented by "faithful" neural components. The general solution to this problem is to use parallelism and distributed computing facilities in speeding up computation. While a number of neural systems have been ported to supercomputers, we are currently developing a distributed simulation environment to run on networks of low-cost computers (Weitzenfeld, Peguero, and Gutiérrez, 2000). In general, the client-server distributed architecture has become quite pervasive, thanks to the Internet.

A web-based simulation interface (Alexander, Arbib, and Weitzenfeld, 1999) brings additional possibilities to the process of neural modeling. For example, users could be offered shared model repositories in creating new models or in addressing experimental data linked to it. These two thrusts are part of a project known as Brain Models on the Web (BMW), a model repository in which model assumptions, empirical data, and simulation results are stored (see DATABASES FOR NEUROSCIENCE).

An important issue arising from the sharing of module libraries is how to reuse portions of different models in creating new ones. An important consideration is to provide a general module interconnection specification to be followed by all modelers. This specification should deal with issues such as “edges” in the block diagrams as the one shown in Figure 1, where module interconnections and the corresponding ports are designed to deal only with primitive data without any temporal considerations. Additionally, the specification could address the relationship to the particular experimental protocol on which the model is based. These aspects need to be defined and then specified as a “meta-level” that will separate the internal module characteristics from the external ones.

Another consideration is the extensibility the system. Since not all users use similar simulation systems, it is important to offer interoperability of data and model descriptions to be shared by multiple simulation systems and applications in general. Additionally, integration with simulated or real-time REACTIVE ROBOTIC SYSTEMS (q.v.) is of particular interest, in particular BIOLOGICALLY INSPIRED ROBOTICS (q.v.), as exemplified by a number of NSL-based neural architectures developed to control mobile robots (Fagg et al., 1992; Weitzenfeld, 2000). Since many approaches exist today to mobile robotics, it is an interesting challenge to design new architectures integrating nonneural and neural-based approaches (Arkin et al., 2000).

Road Map: Implementation and Analysis

Related Reading: Neurosimulation: Tools and Resources; Schema Theory; Single-Cell Models

References

- Agha, G., 1986, *Actors: A Model of Concurrent Computation in Distributed Systems*, Cambridge, MA: MIT Press.
- Alexander, A., Arbib, M. A., and Weitzenfeld, A., 1999, Web simulation of brain models, in *Proceedings of SCS 1999 International Conference on Web-Based Modelling and Simulation*, January 17–20, San Francisco, CA.
- Arbib, M. A., 1989, *The Metaphorical Brain 2: Neural Networks and Beyond*, New York: Wiley. ♦
- Arkin, R. C., Ali, K., Weitzenfeld, A., and Cervates-Perez, F., 2000, Behavioral models of the praying mantis as a basis for robotic behavior, *J. Robotics and Autonomous Systems*, 32(1):39–60.
- Crowley, M., Oztog, E., and Mármol, S., 2002, Crowley-Arbib saccade model, in *The Neural Simulation Language: A System for Brain Modeling* (A. Weitzenfeld, M. Arbib, and A. Alexander, Eds.), Cambridge, MA: MIT Press.
- Fagg, A. H., King, I. K., Lewis, M. A., Liaw, J. S., and Weitzenfeld, A., 1992, A neural network based testbed for modeling sensorimotor integration in robotics applications, in *Proceedings of IJCNN '92*, Baltimore, MD.
- Gosling, J., Joy, B., Steele, G., and Bracha, G., 2000, *The Java Language Specification*, 2nd ed., Reading, MA: Addison-Wesley.
- Stroustrup, B., 2000, *The C++ Programming Language*, Special Edition, Reading, MA: Addison-Wesley.
- Wegner, P., 1990, Concepts and paradigms of object-oriented programming, *SIGPLAN OOPS Messenger*, 1(1):7–87. ♦
- Weitzenfeld, A., 1993, ASL: Hierarchy, composition, heterogeneity, and multi-granularity in concurrent object-oriented programming, in *Proceedings on Neural Architectures and Distributed AI: From Schema Assemblages to Neural Networks Workshop*, Oct. 19–20, Center for Neural Engineering, USC, Los Angeles, CA.
- Weitzenfeld, A., 2000, A multi-level approach to biologically inspired robotic systems, in *Proceedings of NNW 2000 10th International Conference on Artificial Neural Networks and Intelligent Systems*, Prague, Czech Republic, July 9–12.
- Weitzenfeld, A., Arbib, M. A., and Alexander, A., 2002, *The Neural Simulation Language, A System for Brain Modeling*, Cambridge, MA: MIT Press. ♦
- Weitzenfeld, A., Peguero, O., and Gutiérrez, S., 2000, NSL/ASL: Distributed simulation of modular neural networks, in *Proceedings of MICAI 2000: Advances on Artificial Intelligence*, Acapulco, Mexico, April 10–14, LNCS 1796.
- Yonezawa, A., and Tokoro, M. (Eds.), 1987, *Object-Oriented Concurrent Programming*, Cambridge, MA: MIT Press.

Object Recognition

Bosco S. Tjan

Introduction

Visual object recognition is a classification task that assigns a behaviorally relevant label to a region of an image. It is one of the most important functions carried out by the human visual system. Understanding how it is achieved is almost as great a task as understanding visual processing in its entirety (Marr, 1982; Ullman, 1997; Rolls and Deco, 2002). Object recognition is a special case of PATTERN RECOGNITION (q.v.) in which the pattern to be classified is a two-dimensional (2D) projection of a three-dimensional (3D) structure. The 3D structure in question can be a single object (e.g., a car), an arrangement of objects (a convoy of cars), or even a space surrounding the observer (the interior of a car). For the same object, the assigned label depends on the task. A car may be recognized as “an obstacle” (along with garbage cans, lampposts) when one tries to maneuver through parked cars or “an approaching hazard” when one tries to cross the street. The level of classification also varies. An object can be recognized at what cognitive psychologists call the *basic* level (e.g., “a car”), a coarser *superordinate* level (“a vehicle”), or a finer *subordinate* level (“VW Beetle”). In general, object recognition is a mapping, $G: \{x\} \rightarrow \{c\}$, from a set of input images $\{x\}$ to a set of task-dependent class labels $\{c\}$.

Ideally, the mapping should be achieved by selecting the class label $c \in \{c\}$ that *best* explains the image in a probabilistic sense (i.e., by maximizing the posterior probability $\Pr(c|x)$; see BAYESIAN NETWORKS). The challenge lies in the fact that the 2D image of a 3D structure changes dramatically under different *imaging conditions*: variations in lighting, position of the observer, presence or absence of other objects in the foreground and background, etc. A theory of object recognition must therefore explain (1) how a system may retain constancy in object classification in spite of large image variability, and (2) how such a system may learn and generalize. Among all the varying imaging parameters, the one that has received most attention is *viewpoint*, i.e., viewing position and direction of the observer relative to an object.

In this article, we review a number of computational theories of object recognition. The review, which is neither exhaustive nor comprehensive, is meant to show a broad range of the approaches. Space limitations prevent us from discussing the psychological validity of each approach—a difficult issue that is hotly debated. We emphasize computational theories over their neurological implementations, relying on other articles in this *Handbook* to bridge the two (see OBJECT RECOGNITION, NEUROPHYSIOLOGY; OBJECT STRUCTURE, VISUAL PROCESSING). Our discussion of the compu-

tational theories of object recognition leads to a language of decision complexity, which we will use to characterize the trade-offs chosen by each theory. We will argue that a general-purpose object recognition system, such as the human brain, must be able to represent an object in multiple ways, and we outline a computational framework for such a system.

Matching: A Core Operation

Object recognition is about matching an input *image* (2D array of luminance and chromaticity values) to a stored *representation* of an object. A very rudimentary form of object recognition is *template matching* performed at (or near) the image level, in which an input image is matched pixel to pixel to a set of stored images. The input image may be preprocessed to reduce the variability in luminance, size, 2D position, and/or image-plate rotation. A degree of mismatch is computed with some knowledge about the noise processes in image acquisition and preprocessing. For example, the commonly used mean-square error, or L2-norm, between an image and a stored template is appropriate if the imaging noise is an identically and independently distributed Gaussian in pixel value. A match is found when the degree of mismatch is less than some threshold.

The obvious advantage of image-level template matching is that little preprocessing is required, and few assumptions about the utility and detectability of certain image features are needed to transform the input image into a template. On the other hand, small variations in imaging conditions, such as changes in viewpoint or illumination direction, can produce a large mismatch between an input image and a restored template. As a result, image-level template matching methods are of little use for general-purpose object recognition.

On the other hand, it is not difficult to see that template matching is at the core of every object recognition theory. Superficially, the final decision stage that matches a processed input to a stored representation of an object is a template matcher operating at the level of the representation. A more fundamental role of template matching can be seen as follows. Let $v_c(\mathbf{g})$ be a view of object c in some representation, where \mathbf{g} is a vector of imaging parameters (i.e., viewpoint, illumination, occlusion, etc.). The posterior probability, $\text{Pr}(c|x)$, of object c being present in the image x can be expressed in terms of all the possible views of c by integrating over the generic variable \mathbf{g} . With Bayes's rule, we have:

$$\text{Pr}(c|x) = \int_{\mathbf{g}} p(v_c(\mathbf{g})|x) d\mathbf{g} = \int_{\mathbf{g}} \frac{p(x|v_c(\mathbf{g}))p(v_c(\mathbf{g}))}{p(x)} d\mathbf{g} \quad (1)$$

where $p(v_c(\mathbf{g}))$ is prior probability density of a view, $p(x|v_c(\mathbf{g}))$ is the likelihood of $v_c(\mathbf{g})$ being the cause of the input image x , and $p(x)$ is the probability density of seeing the input image x , which, for our purposes, is simply a normalization constant independent of c and \mathbf{g} . The decision rule that is most accurate on average is to select c that maximizes $\text{Pr}(c|x)$. That is,

$$\arg \max_c \text{Pr}(c|x) = \arg \max_c \int_{\mathbf{g}} p(x|v_c(\mathbf{g}))p(v_c(\mathbf{g})) d\mathbf{g} \quad (2)$$

If the view $v_c(\mathbf{g})$ is represented as an N -pixel image, the likelihood function $p(x|v_c(\mathbf{g}))$ is simply the probability density function describing the imaging noise. For example, if the imaging noise is an identically and independently distributed Gaussian pixel noise, then $p(x|v_c(\mathbf{g})) = 1/(\sigma\sqrt{2\pi})^N \exp(-\|x - v_c(\mathbf{g})\|^2/2\sigma^2)$. The L2-norm term $\|x - v_c(\mathbf{g})\|^2$ in the exponent represents image-level template matching. In other words, object constancy could in theory be achieved with image-level template matching, provided one has the space and time to store and compare the input image against all views of all objects of interest (or according to Tjan and Legge

[1998], a large but finite set of random views would suffice). Moreover, the decision rule of Equation 2 evaluated at the image level, with unlimited memory for object views, achieves the theoretical maximum in average accuracy.

Theories of Object Recognition

Theoretical ideals notwithstanding, if object constancy is to be achieved practically, a visual system must explore regularities in the mapping between an input image and the intended output label. This is often done in two steps. First, photometric and geometric regularities inherent in the image-formation process are explored by extracting image features that remain constant (or nearly so) with respect to changes in imaging conditions. The extracted features then form a *representation* of the input. Second, this representation of the input is compared to the stored representations of different objects to identify a match. This decision process explores the statistical regularities inherent in the mapping between the representation (as opposed to the input image) and the output label. Theories of object recognition differ in their respective emphases on the representation or decision step. In their succinct discussion of the computational theories of object recognition, Trucco and Verri (1998) mentioned two general approaches to object recognition. If a system constructs a representation that does not vary (i.e., is *invariant*) with respect to imaging parameters but is sufficiently discriminating to tell one object from another, then the decision step will be relatively trivial. This type of approach is referred to as being *invariant based*. However, finding features that are both invariant and discriminatory is often difficult. The alternative is to rely less on an invariant representation and more on statistical inference to implicitly capture the regularities between the stimuli and the output labels. Because such an approach often makes object decisions using a representation closely resembling the input image, it is termed *appearance based*. A third type of approach, which does not fit neatly into either category described by Trucco and Verri, is to store a 3D model for each object. Recognition proceeds by trying to align a 3D model to match the input image. We will refer to this type of approach as *model based*.

Invariant-Based Object Recognition

For an invariant-based object recognition system, the primary objective of visual processing is to extract and construct features that are constant relative to imaging conditions. For example, the presence of an *edge* often signals a discontinuity in surface orientation, mostly independent of lighting conditions. Hence, edge detection and contour formation are often proposed as the first steps of visual processing. The position, orientation, length, and curvature of a contour, however, vary with respect to an observer's viewpoint. Quantization is sometimes used to gain an additional degree of invariance. Within sensory acuity, many edges appear straight. Classifying the curvature of an edge as "straight" or "curved" makes the (qualitative) curvature of an edge invariant to viewpoint. Quantization represents a trade-off between discriminability and invariance. For example, a circle could not be discriminated from an ellipse if edges were classified solely as either straight or curved.

Another important means to attain invariance is to form compound features from elementary ones. For example, if there exist four identifiable points, P_1, \dots, P_4 , on a straight edge (perhaps due to surface markings), then the *cross ratio*, defined as $(|P_1 - P_2| \cdot |P_3 - P_4|) / (|P_1 - P_3| \cdot |P_2 - P_4|)$, is a quantity invariant to viewpoint under perspective projection (Duda and Hart, 1973). The cross-ratio of four collinear points can therefore be treated as a discriminating "feature" invariant to viewpoint. Other invariants based on configurations of points, lines, and conics have been proposed for the purpose of object recognition (Mundy and Zisserman,

1992). Like feature quantization, discriminability in object shape is often reduced when object identification relies on invariant features. For example, if shapes are represented by cross-ratios alone or any invariant derived from them, objects that are an affine (linear) transformation away from one another will be indistinguishable. In particular, a cube would be indistinguishable from a rectangular box or a parallelepiped.

Human observers are unaware of most of the geometric invariants, such as cross-ratios. However, a few spatial arrangements of features, such as collinearity of edges, parallelism, closure, or symmetry, are perceptually conspicuous. It has been proposed that such feature arrangements are ecologically significant, useful for segmenting an object from a cluttered environment. Lowe (1987) demonstrated an object recognition system that relies on perceptual organization of features to achieve invariant object recognition in a cluttered scene. Specifically, line segments were hypothesized to belong to the same object if they could be grouped based on proximity, parallelism, or collinearity.

Another well-known example of achieving invariance by feature grouping is the recognition by components theory (RBC; Beiderman, 1987). Beiderman argued that some 50 elementary 3D volumes (called *geons*) could be uniquely identified from their 2D images over a range of viewpoints by expressing edge configurations in terms of their qualitative (quantized) properties: straight versus curve, parallel versus converging, and type of intersection (“fork,” “arrow,” “L,” or “T”). General-purpose object recognition proceeded by first identifying the geons then the spatial relationships between the geons. Spatial relationships were also identified in quantized terms, such as “on,” “next-to,” “left-of,” etc. In short, RBC proposes that an object is represented by the qualitative spatial relationship between its parts, and the parts are represented as geons.

In general, an invariant-based object recognition theory proposes to construct, through stages of feature extraction, combination, and quantization, a set of higher-order features that are largely unaffected by imaging conditions. The final set of invariant features is then compared to those stored in memory as object models. Recognition results when a reasonably good match is found between the invariant representation of the input and that of a known object.

Model-Based Object Recognition

Instead of combining features or quantizing feature attributes to achieve invariance, a visual system can factor out image variability by making explicit use of the 3D structure of an object. Lowe (1987) and later Huttenlocher and Ullman (1990) proposed the alignment method for object recognition, in which a small number (three or more) of identifiable features (parallel line segments, corners, junction types) were first hypothesized to be in correspondence with the compatible features of a stored 3D object model. The hypothesized correspondence was used to compute an orientation of the object (called a *pose*) that could bring the 3D model features into alignment with the corresponding 2D image features. The validity of the proposed object model was determined by the degree of match between the rest of the image and the projection of the model onto the image using the computed pose.

Common to most model-based approaches is an explicitly 3D representation of the objects, which allows the appearance of an object to be determined from any viewpoint, thus achieving invariance. Acquiring such a representation, however, can be challenging, especially if the 3D representation is to be derived from 2D inputs. Fortunately, an object’s 3D structure is implicitly captured in its 2D images. Techniques have since been developed to exploit the three-dimensionality using image-to-image operations, bypassing the need to explicitly construct or store any 3D object model. Such techniques are often referred to as being appearance based.

Appearance-Based Object Recognition

The most basic (and least plausible) form of appearance-based object recognition is the image-level template matching described earlier. It typifies one prominent feature of the appearance-based object recognition theories, namely, that decisions are made based on representations that closely resemble the raw input image. This is in sharp contrast to the invariant-based approach, which relies on higher-order features designed to be invariant to imaging conditions. Making object decisions near the image level relieves the system from depending on the complicated and often noisy feature extraction and construction processes, which results in higher tolerance to input noise and better discrimination. In addition, operations needed for learning new objects (model acquisition) are usually more robust and straightforward. The obvious challenge, however, is to achieve a reasonable level of viewpoint invariance for the technique to be useful.

An N -pixel image view $\mathbf{v}_c(\mathbf{g})$ can be expressed as a vector of N pixel values. This vector denotes a point in the *image space*. Over all the continuously varying imaging parameters (e.g., viewpoint, illumination), the set of points $\mathbf{v}_c(\cdot)$ form an *object manifold* embedded in the image space, which implicitly captures the 3D structure of the object. The process of object recognition in the image space can be thought of as determining which object manifold $\mathbf{v}_c(\cdot)$ is the “closest to” the input image. Because the input is a single point in the image space, but a manifold is a set of points, the distance between the two can take many forms. Equation 2 expresses one such form, which maximizes the average recognition accuracy. A brute-force implementation of Equation 2, however, would require storing too many images and would take too long to compute to be practical. Most of the appearance-based theories try to approximate Equation 2 by exploiting various regularities of the object manifolds, in order to significantly reduce storage and computation requirements.

An object manifold $\mathbf{v}_c(\cdot)$ can be arbitrarily complex and not necessarily smooth, even if the imaging parameters vary smoothly. There exist conditions, however, where the object manifold has a simple analytic form with parameters that can be determined from a sparse set of images. For example, if the only allowable variations in imaging conditions are the direction and intensity of illumination, then the object manifold will be piecewise linear (ignoring cast shadow). This means that images of the same object over a range of illumination changes can be synthesized by linearly combining three known images. Moreover, recent analysis has shown that the entire object manifold under illumination changes is “flat,” residing close to a nine-dimensional subspace embedded in the image space of a few million dimensions. A practical appearance-based recognition system needs to store only nine images per object to achieve invariance over lighting variations (Basri and Jacobs, 2001; Lee, Ho, and Kriegman, 2001).

Another important case is when the image $\mathbf{v}_c(\mathbf{g})$ is not a vector of luminance values but a list of 2D coordinates of feature points. Furthermore, it is assumed that the correspondence between feature points across different views has been established. Object manifolds in this *correspondence space* (as opposed to the image space) are piecewise linear, if the viewing distance is large compared to object size. Ullman and Basri (1991) explored this property and pointed out that the views of an object over a range of viewpoints could be synthesized by linearly combining the coordinates of corresponding features between two views, thus avoiding the need for storing a large number of views. Other means of correspondence-space view interpolation, such as one using RADIAL BASIS FUNCTION NETWORKS (q.v.) (Poggio and Edelman, 1990), have also shown promise.

If the analytical form of an object manifold is not known, as is often the case, a visual system can still exploit the regularity of the

manifold implicitly. For example, it could (1) use an inexact matching operation, (2) reduce the dimensionality of the manifolds, or (3) increase the distance between manifolds by mapping them to an even higher-dimensional space.

Von der Malsburg and colleagues (Lades et al., 1993) adapted DYNAMIC LINK ARCHITECTURE (q.v.) to perform elastic graph matching for object recognition. An object from a given viewpoint was represented by a grid of sparsely sampled image features, each being a vector (called a “jet”) representing the local spatial frequency spectrum. To match an input image, the grid was allowed to deform mildly in search of the best-matched feature points. Recognition can be invariant over a modest range of viewpoints because a small change in viewpoint tends to displace feature points without significantly altering their local spatial frequency spectrum.

The set of images an appearance-based system can possibly store for a given object is usually a very sparse sample of the object manifold. Therefore, it is often advantageous to reduce the dimensionality of the object manifold to make the stored images more representative. The eigenspace method (Murase and Nayar, 1995; see PRINCIPAL COMPONENT ANALYSIS and INDEPENDENT COMPONENT ANALYSIS) is a principled way of achieving this. The idea is to express a set of n known images of an object in terms of a weighted sum of their principal components $\mathbf{e}_1 \dots \mathbf{e}_n$. This amounts to a coordinate transformation from the image space (of pixel values) to an eigenspace via rigid rotation. If the principal components are arranged in descending order of their associated eigenvalues, then the weighted sum of the first k principal components ($k < n$) is the “best” k -dimensional approximation of the image set, in the sense of having the minimum mean-square error. That is, if $\mathbf{v}_c(\mathbf{g})$ is an image of the object c in some imaging condition \mathbf{g} , then

$$\begin{aligned} \mathbf{v}_c(\mathbf{g}) &= w_1 \mathbf{e}_1 + w_2 \mathbf{e}_2 + \dots + w_k \mathbf{e}_k + \dots + w_n \mathbf{e}_n \\ &\approx w_1 \mathbf{e}_1 + w_2 \mathbf{e}_2 + \dots + w_k \mathbf{e}_k = [\mathbf{e}_1, \dots, \mathbf{e}_k] \mathbf{w}_c(\mathbf{g}) \end{aligned} \quad (3)$$

where $\mathbf{w}_c(\mathbf{g}) = [w_1, w_2, \dots, w_k]^T$. In practice, k can often be much smaller than n . This means that an image $\mathbf{v}_c(\mathbf{g})$ in a high-dimensional image space can be approximated with a k -dimensional weight vector $\mathbf{w}_c(\mathbf{g})$ in an eigenspace. Furthermore, the only images that need to be stored are the first k eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_n$, which jointly define the eigenspace.

The eigenspace method seeks to reduce the dimensionality of an object manifold. Some recent appearance-based theories, however, take exactly the opposite approach, realizing that if the manifolds of different objects can be made far apart from each other, there will be no need to represent the manifolds in any detail. A simple hyperplane (a plane in a high-dimensional space) will be sufficient to partition the decision space into regions, with each containing at most one object manifold. SUPPORT VECTOR MACHINES (q.v.), for example, first map a set of known images to a high-dimensional *feature* space by passing them through a nonlinear transformation ψ . The dimensionality of this feature space can be arbitrarily high, making it possible for a hyperplane to separate one object’s manifold from another. For each object, a support vector machine finds the *optimal* hyperplane in the feature space by maximizing the distance between the hyperplane and the object manifolds it tries to separate. Because such a hyperplane is fully determined by the points on the object manifolds that are closest to it, only those points, which are simply images, need to be stored. These images that determined the optimal hyperplanes are called *support vectors*. Conceptually, object recognition proceeds by first mapping an input image to a point in the feature space via ψ , and then deciding on which sides of the hyperplanes the input resides. Mathematically, this two-step process is equivalent to comparing the input image with the set of stored images (support vectors) using an *inner product kernel* function K , which relates to the nonlinear feature map ψ as $K(\mathbf{v}, \mathbf{v}_s) = \psi(\mathbf{v})^T \psi(\mathbf{v}_s)$. The inner product kernel $K(\mathbf{v}, \mathbf{v}_s)$ returns a scalar and can be thought of as a generalized similarity

measure in the feature space between the unknown input image \mathbf{v} and a set of stored image \mathbf{v}_s .

Two key features of an appearance-based theory distinguish it from the invariant-based approach. First, the stored representation of an object is often very close to the images of the object and does not possess a high degree of invariance. Second, invariant object decisions are achieved with some distance measure between the unknown input image and a sizable set of image-like representations that jointly define an object.

Decision Complexity and Representations

Each theory reviewed in the preceding section proposes one “final” representation used for object recognition. The theories differ in the amount of details the final representation retains. There is generally a trade-off between invariance (with an abstract representation) and discriminability (with an image-specific representation).

Regardless of the approach, an object recognition system has to decide which object manifold an input image belongs to. The decision task is easy if the object manifolds are (1) simple and (2) far apart. Historically, most object recognition theories, especially the invariant-based ones, have pursued the first factor. If successful, such an approach would have produced a general-purpose object representation that is appropriate regardless of the task or objects involved. However, if the manifolds for different objects intertwine, an attempt to reduce the manifolds’ dimensionality can also make them indistinguishable from one another. Furthermore, the ever-present noise in the input can perturb the transformation processes required to map the input image to its final representation, leading to misidentification.

We refer to the extent to which object manifolds intertwine in the decision space as the *decision complexity*. Decision complexity is a function of the recognition task and the *set* of objects to be recognized. The decision complexity of telling two chickens apart is clearly different from that of recognizing a chicken from a fish. Tjan and Legge (1998) used a random sampling method to estimate the decision complexity in the image space for different sets of objects. Specifically, they measured the number of unprocessed raw images that an *ideal observer* operating according to Equation 2 would need to store in order to fully represent a set of objects. They found that decision complexity in the image space varied over two orders of magnitude across different types of objects and was highly dependent on what other objects were involved in the task. This result matches qualitatively to humans’ varying difficulty in achieving viewpoint-invariant recognition across object sets.

In a practical object recognition system, the mapping between the image space and the decision space is highly nonlinear. Hence, decision complexity in the image space does not necessarily dictate decision complexity in the decision space. Intuition and practical experiences, however, suggest that the two are coupled. How intertwined the object manifolds are in the image space limits the minimum decision complexity one can practically attain in a decision space, after the noise in the input and visual processing has been accounted for. The wide range of image-space decision complexities observed by Tjan and Legge therefore implies that a general-purpose object recognition system cannot rely on a single form of representation for a wide range of tasks and objects.

Adaptive Selection of Representations

The idea that the human visual system may use multiple forms of representations is uncontroversial, but details are lacking. Specifically, what are the representations used by the human visual system, and how does the system decide which representation to use for a particular task? We make two observations (Tjan, 2001). First, to obtain a representation with a high degree of invariance, a series

of processing stages are needed for feature extraction and the construction of higher-order features. This hierarchical processing architecture is common to most invariant- or model-based theories and is consistent with neurological findings about the visual system (Rolls and Deco, 2002). The intermediate output of each processing stage is in fact a representation at a particular level of abstraction. Thus, in the process of forming a representation with high degree of invariance, we obtain as “byproducts” a series of intermediate representations, each making a different trade-off between discriminability and invariance, each suitable for a recognition task of a different decision complexity. Second, if object decisions are made by the optimal strategy of maximizing posterior probabilities (i.e., Equation 2), or are approximately so, the magnitude of posterior probability $\Pr(c|x)$ of the best choice can serve as a confidence measure regarding the decision (because posterior probabilities sum to one). The confidence of a decision can be low if (1) the representation is not detailed enough to make the discrimination, or (2) the representation is too detailed to attain a sufficient level of invariance for recognizing a novel input. Such a confidence measure can therefore be used to determine on-the-fly which level of representation is most appropriate for a given task.

Tjan (2001) proposed a simple architecture for object recognition that adaptively selects the most appropriate level of representation for a given task and objects. The architecture consists of a single visual processing pathway, loosely modular. Decision sites with local memory are attached to the processing modules along this pathway. Object-identity decision is made independently and in parallel at every decision site. The response latency of a site decreases with the maximum posterior probability evaluated at the site, but increases with the site's effective memory size. The first-arriving response from any site is taken to be the system's response. Simulation results showed that, although the intermediate representations were generic, the effective representation revealed behaviorally by the system appeared to be specific to the object category and task. The system showed an efficient trade-off between speed and accuracy, indicating that the object decisions were made at the appropriate level of abstraction.

Discussion

A great number of object recognition theories have been proposed in the past 30 years. While not all of them are biologically feasible, each provides important insight into the range of possible computations and representations for recognizing objects. Each theory makes a particular trade-off between discriminability and invariance. A system designed to discriminate similar objects is generally more sensitive to imaging conditions and requires a larger memory footprint to represent the objects than a system designed to discriminate highly dissimilar objects. Our analysis of decision complexity shows that it is unlikely for a single form of representation to be suitable for all kinds of object recognition tasks a human or other visual animals encounter each day. A key ingredient in a

comprehensive brain theory for object recognition is therefore a computational framework that allows on-demand selection or adaptation of representations based on the task. To this end, we proposed a simple confidence-driven horse-racing scheme (a sort of first past the post, temporal winner-take-all scheme) for self-selecting the most appropriate level of abstraction, given a finite set of available representations along a visual processing pathway. Simulation results suggested that the framework is consistent with known behavioral and neurological data. We believe this framework is sufficiently simple and concrete for it to be biologically viable.

Road Map: Vision

Related Reading: Feature Analysis; Object Recognition, Neurophysiology; Object Structure, Visual Processing; Pattern Recognition; Visual Scene Perception

References

- Basri, R., and Jacobs, D., 2001, Lambertian reflectance and linear subspaces, in *Proceedings of the IEEE 8th International Conference on Computer Vision*, Los Alamitos, CA: IEEE Computer Society Press, pp. 383–390.
- Biederman, I., 1987, Recognition-by-components: A theory of human image understanding, *Psychol. Rev.*, 94:115–147.
- Duda, R. O., and Hart, P. E., 1973, *Pattern Classification and Scene Analysis*, New York: Wiley.
- Huttenlocher, D. P., and Ullman, S., 1990, Recognizing solid objects by alignment with an image, *Int. J. Comput. Vision*, 5:195–212.
- Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., Von der Malsburg, C., Würtz, R. P., Konen, W., 1993, Distortion invariant object recognition in the dynamic link architecture, *IEEE Trans. Comput.*, 42:300–311.
- Lee, K. C., Ho, J., and Kriegman, D., 2001, Nine points of light: Acquiring subspaces for face recognition under variable lighting, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA: IEEE Computer Society Press, pp. 519–526.
- Lowe, D. G., 1987, Three-dimensional object recognition from single two-dimensional images, *Artif. Intell.*, 31:355–395.
- Marr, D., 1982, *Vision*, New York: Freeman.
- Mundy, J. L., and Zisserman, A., 1992, *Geometric Invariance in Computer Vision*, Cambridge, MA: MIT Press.
- Murase, H., and Nayar, S. K., 1995, Visual learning and recognition of 3-D objects from appearance, *Int. J. Comput. Vision*, 14:5–24.
- Poggio, T., and Edelman, S., 1990, A network that learns to recognize three-dimensional objects, *Nature*, 343:263–266.
- Rolls, E. T., and Deco, G., 2002, *Computational Neuroscience of Vision*, New York: Oxford University Press.
- Tjan, B. S., 2001, Adaptive object representation with hierarchically-distributed memory sites, in *Advances in Neural Information Processing Systems 13*, San Mateo, CA: Morgan Kaufmann, pp. 66–72.
- Tjan, B. S., and Legge, G. E., 1998, The viewpoint complexity of an object recognition task, *Vision Res.*, 38:2335–2350.
- Trucco, E., and Verri, A., 1998, *Introductory Techniques for 3-D Computer Vision*, Upper Saddle River, NJ: Prentice Hall.
- Ullman, S., 1997, *High-level Vision*, Cambridge, MA: MIT Press.
- Ullman, S., and Basri, R., 1991, Recognition by linear combinations of models, *IEEE Trans. Pattern Anal. Machine Intell.*, 13:992–1005.

Object Recognition, Neurophysiology

Guy Wallis and Heinrich H. Bülthoff

Introduction

As viewing distance, viewing angle, or lighting conditions change, so too does the image of an object that we see. Despite the seem-

ingly endless variety of images that objects can project, the human visual system is able to rapidly and reliably identify those objects. How humans achieve this feat of recognition has long been a source of debate. Researchers have still not agreed on even the most fun-

damental questions of how objects are represented in cortex. This article provides a brief overview of some theoretical approaches in the context of mainly neurophysiological evidence. It also considers the related question of objects within a physical context, that is, the analysis of visual scenes. Scene analysis is relevant to the question of object recognition because scenes are initially recognized at a holistic, object-like level, providing a context or “gist” that itself influences the speed and accuracy of recognition of the constituent objects (Rensink, 2000). A precise characterization of gist remains elusive, but it may well include information such as global color patterns, spatial frequency content, correlational structure, or anything that is useful for categorizing or recognizing the scene.

To provide an anatomical framework for this chapter it is instructive to review the major functional divisions of visual cortex. Visual processing begins at the back of neocortex, in the occipital lobe. From there, information flows down into the temporal lobe, forming the ventral stream, and up into the parietal lobe, forming the dorsal stream (Figure 1). On the basis of neuropsychological and single-cell recording data, theorists have proposed a functional division between these streams. The dorsal stream is considered to process deciding “where” an object is and the ventral stream “what” an object is (Ungerleider and Haxby, 1994). In this article we mainly focus on the “what” stream, since it is seen as the center of object recognition, but an integrated model of scene perception will almost certainly require a broader approach encompassing all four lobes.

The Ventral Stream

The path from primary visual cortex to the inferior temporal lobe (IT) passes through as many as ten neural areas before reaching the last wholly visual areas (Figure 1). Early recordings of the temporal lobe indicated neurons selective for faces, and from later recordings workers were able to verify that these cells could not be excited by simple visual stimuli or as part of an emotional response to seeing a particular face (Rolls, 1992; Logothetis and Sheinberg, 1996).

One striking feature of the response properties of neurons in the IT is that the farther down the ventral stream one looks, the more specialized and selective the neurons become. Of special interest

to the field of object recognition was the discovery that along with increasing selectivity, many neurons become tolerant to shifts in stimulus position; changes in viewing angle, size/depth, or illumination; or the spatial frequencies present in the image (Rolls, 1992).

A great deal of this work originally had to do with neurons selective for faces, but although face cells account for as much as 20% of neurons in some regions of the IT and STS, they account for only about 5% of all cells present in IT cortex. In the early 1990s, Tanaka and his colleagues (Tanaka et al., 1991) showed that many of the remaining neurons are selective for complex combinations of features, including a basic shape with bounded light and shaded or colored bounded regions, and that these neurons also demonstrate useful invariance properties. This work has dispelled the idea of a special stream designed specifically for face recognition.

Recent work has focused on the issue of how the cellular response properties of temporal lobe neurons change over time. Several studies have shown that repeated exposure to a particular object class results in changes in the number of neurons selective for that stimulus (e.g., Rolls, 1992; Miyashita, 1993; Logothetis and Sheinberg, 1996). In humans, we should not be surprised to learn that a car enthusiast has neurons tuned to the appearance of a yellow VW Beetle, or that a lepidopterist has neurons tuned to an Orange Tip butterfly.

The Dorsal Stream

Abstracting an object’s form from its precise location, size, or orientation is clearly important for tasks such as recognition and categorization. However, there are plenty of situations in which an object’s location and orientation are important, not least when we want to interact with that object by picking it up or manipulating it. The processing of location and orientation appears to be the major concern of neurons in the parietal lobe. These neurons form part of the dorsal stream. In humans, damage to the parietal lobe severely affects the localization of objects within a scene, leading to disorders such as visual neglect, and it appears that the dorsal stream is intrinsically linked to the control of visual attention and eye movements (Ungerleider and Haxby, 1994).

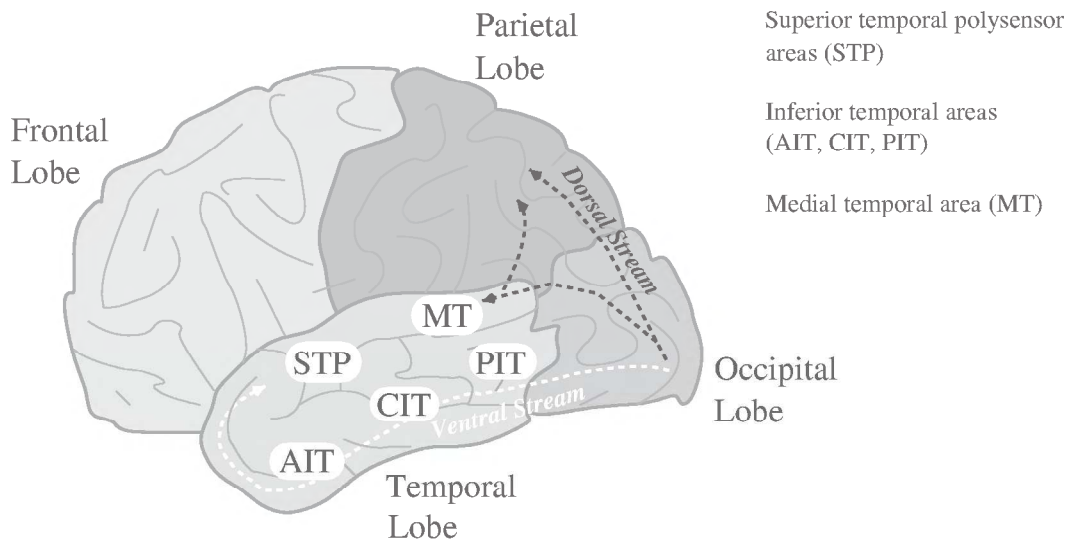


Figure 1. Principle divisions of neocortex, including the main areas of the temporal lobe. Dark arrows indicate information flow along the dorsal stream. Light arrow indicates flow along the ventral stream.

Of course, many tasks require the interaction of the two types of information, both where and what an object is. Our lepidopterist would like to be able to net a Tortoiseshell fluttering among Red Admirals. This raises an as yet unanswered question, namely, how these types of information interact and where various representations are held. It turns out that there are plenty of routes that information could take between the temporal and parietal lobes, including a direct route, via the occipital lobe or the frontal lobe. It has been shown, for example, that regions AIT and CIT of IT (see Figure 1) connect to the frontal lobe, and CIT and PIT connect to the parietal lobe (Webster, Bachevalier, and Ungerleider, 1994). One aim of any modeling work must be to investigate the possible significance of these connections.

A Processing Hierarchy

One of the striking features of the ventral stream is its hierarchical structure. Neurons in the regions of the temporal lobe furthest from the retina can be thought of as sitting on top of a processing pyramid (Figure 2). Receptive field size grows steadily larger the farther up this pyramid one looks, and the response times of neurons also rise systematically (Rolls, 1992).

One possible explanation for the presence of such a hierarchy is that the visual system is gradually building representations of ever-increasing complexity to produce neurons that respond to combinations of inputs, themselves forming the effective stimuli for later neurons. By responding to local combinations of neurons co-active in the previous layer, arbitrary spatial arrangements of the same features should fail to activate the same neuron. This should then reduce the chance of finding the trigger features supporting recognition in random arrangements of the features, an issue often referred to as the “feature-binding problem.” Some of the most selective and view-invariant responses belong to cells in the superior temporal areas. These neurons appear to pool the outputs of view-selective AIT cells. One explanation of how the STPa neurons know which AIT neurons to group together is discussed in the next section.

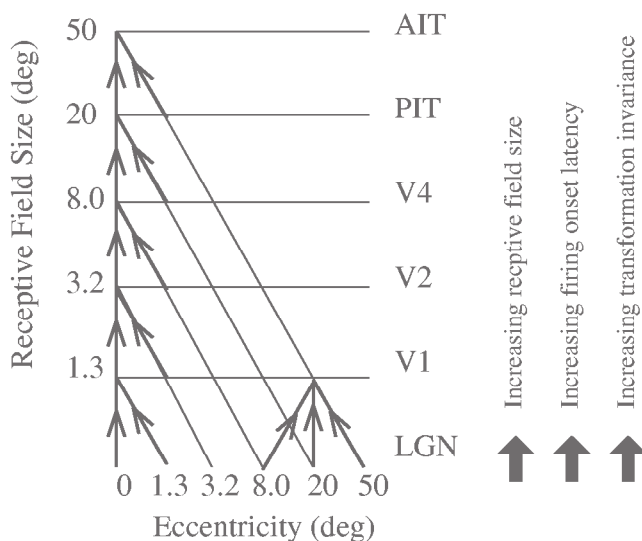


Figure 2. Schematic of convergence in the ventral processing stream. The steady growth in receptive field size suggests that neurons in one layer of the hierarchy receive input from a select group of neurons in the preceding layer. The time taken for the effects of seeing a new visual stimulus increases systematically through the hierarchy, supporting the notion of a strictly layered structure.

Neuroanatomists tell us that there are at least as many connections running back as there are running forward in the ventral stream, and this is important when one comes to devise models. The precise use of these connections remains unclear. Some theorists have argued that they are used in recall, and it is true that the act of remembering visual events causes activity to spread into primary visual areas. They may also control visual attention. Certainly attending to specific regions of the visual environment has been shown to facilitate the processing of signals in topographically matched regions of visual cortex, which may well be due to selectively raising the activity (or lowering the activation thresholds) of neurons along the processing hierarchy. One important role that the connections almost certainly do play is in relaying “top-down” influences on recognition, due to expectations or selective attention, perhaps prompted by the gist of a scene. Such influences include contextual priors, which in this case are functions that govern the likelihood of seeing a particular object in a particular context. Our lepidopterist will implicitly change these priors with habitat, improving the chances of correctly distinguishing a Caper White in the rocky bush of South Australia from a Cabbage White on the meadows of southern England.

Some have argued that the backward-projecting connections, apart from their role in relaying attentional mechanisms, play an integral role in normal visual processing. Some have gone so far as to suggest that each neural region forms a recurrent attractor network, each connected through the cortex, up to and including the temporal lobe. Although such models may be needed to deal with confusing or low-quality images, there is good evidence that timing constraints prohibit such a model from acting during the rapid recognition of everyday, familiar objects (Thorpe, Fize, and Marlot, 1996).

Encoding Objects in the Temporal Lobe

Despite the apparent selectivity of temporal lobe neurons, it is important to realize that they are not “grandmother cells” selective for a single entity (such as your grandmother) in the manner proposed in early theories of object representation. On the contrary, many of the cells reported in the literature responded to several examples of objects within their particular object category. Evidence is emerging that object encoding is achieved via small ensembles of firing cells that efficiently and robustly code for individual objects.

Under a distributed scheme, many hundreds or thousands of neurons, each selective for its specific feature, would act together to represent an object. Although many of these features represent only small regions of an object, others appear to represent an object’s outline, or some other global but general property. In addition, the neural representation of these features is more sophisticated than a simple template, since they may exhibit invariance to scale and size, something typical of temporal lobe neurons.

Implementing representation in a distributed code brings with it several advantages. First, the representations are robust to cell damage: since hundreds or thousands of neurons react to the presence of a single object, the death of one neuron within the ensemble will not adversely affect recognition accuracy or speed. Second, a distributed representation provides immediate recognition generalization to novel stimuli: a new object can be represented distinctly from all other stimuli by using a unique combination of the many well-established feature-selective neurons already present. In so doing, each neuron brings knowledge of how its feature changes in appearance with changes in viewpoint. The numerous beneficial, emergent properties of a distributed representation have long been realized by neural network theorists (see POPULATION CODES).

In addition to the general encoding and topological organization of IT cortex, work has also been carried out to establish what func-

tional organization might be present. Some researchers have made moves to describe the functional organization of IT. Cells were tested and their key stimulating features characterized, revealing a columnar structure in which groups of neurons appear to respond to similar though subtly different collections of features. Neighboring columns seem to have less in common. This work in part replicates the findings of other researchers who have described localized "clusters" or "patches" of face cells (see Rolls, 1992), and the findings have been taken by some as evidence for local excitatory and more diverse inhibitory connections within the processing layers, akin to those used in competitive networks (see Wallis and Bülthoff, 1999; Riesenhuber and Poggio, 2000).

Models of Object Representation and Recognition

A huge number of systems for object recognition have been proposed over the years. Some were inspired by the desire to build intelligent machines, others by the desire to describe human recognition processes. This section summarizes some of the popular models and their relevance, or otherwise, to human object recognition. Extensive reference lists can be found in review articles on the topic by Wallis and Bülthoff (1999) and Riesenhuber and Poggio (2000).

One family of models, which owes its heritage to artificial intelligence research in the 1970s, sees the need to extract cues to three-dimensional (3D) structure. Using texture gradients, linear perspective, structure from motion, and so on, models in this family seek to transform the retinal image into a full-fledged internal 3D model capable of rotation, scaling, and translation and therefore matching to a store of known objects. Various means for achieving this reconstruction have been proposed, although perhaps the most preeminent is the geon theory of Biedermann and its associated network model, called JIM (see OBJECT STRUCTURE, VISUAL PROCESSING). Unfortunately, although there are plenty of neurons sensitive to cues such as terminated edges or complex forms of motion, neurophysiologists have yet to find evidence for large quantities of the types of neural analyzers that this type of models would predict, and even less evidence for the set of 36 3D volumetric building blocks that Biederman's theory claims are combined to represent all objects. What is more, there is only limited evidence for the neural synchronization mechanism that it uses to bind elements of activated geons, and there is no evidence of neurons purely selective for the spatial relationships of parts, such as "left of," "above," and so on. Nonetheless, some form of structural representation must surely exist, particularly in defining object categories for distinguishing a quadruped from a biped, or a telephone from an elephant. The JIM model is one of the very few models focused on human object recognition that provides a principled means of extracting and representing structure.

As an alternative to this type of bottom-up object reconstruction, a number of approaches to object recognition have considered the possibility of matching the incoming image to a large collection of 2D images or whole 3D objects. This process takes a number of different forms. In some models the image of the object is normalized for size and location and then simply matched pixel by pixel to a stored set of images. Of course, simple 3D transformations such as depth rotation lead to nontrivial changes in the 2D projected image. To compensate for this, some models have employed local distortions of the incoming image in the matching process. Others have presupposed an ability to extract 3D anchor points in the image that allow stored 3D representations to be rotated and scaled in 3D before the matching process begins. In practice, most of these models work well on predefined sets of objects and small changes in appearance, but they are prone to errors if the incoming image changes considerably. Models that employ local distortion or rotation algorithms are more robust, but this robust-

ness comes at a cost. The models are slow and become slower the more objects are stored in the internal library. The simplest form of 2D template matching is at least fast, and if the process proceeds in parallel, it can scale extremely well as the number of objects increases. However, where all of these models fall down is in explaining our ability to categorize and generalize recognition of new objects to changes in viewing direction.

A possible solution to this final problem is based on a further alternative model for how objects are represented and recognized. This approach once again suggests that objects are stored as images or multiple views (Bülthoff and Edelman, 1992). However, rather than being stored as a single template, each view is represented as a collection of small picture elements, each tolerant to small view changes (Wallis and Bülthoff, 1999; Riesenhuber and Poggio, 2000). Such a system immediately reaps the benefits of a distributed encoding system in terms of robustness and transformation generalization for novel objects; it also accords with the types of neural response properties known to exist in the ventral stream.

In practice, many systems base recognition on a combination of pictorial features. Some have simply attempted to look across the entire image for telltale features, irrespective of relative position, as evidence for the presence of one object rather than another. Of course, models that throw away spatial information in this way run into the problem of "recognizing" random rearrangements of the features triggering recognition. This is not the case for real neurons responsive to faces, which often reduce their response to faces in which the features appear jumbled up (Rolls, 1992; Logothetis and Sheinberg, 1996). Nor is it true for cells responsive to more abstract features (Tanaka et al., 1991); indeed, this is an example of the feature-binding problem. As described earlier in this article, one solution to this problem is to combine features gradually over a series of stages, achieving translation invariance step by step. This has inspired many theorists to take this approach in object recognition. One of the first to construct a truly hierarchical model was Fukushima (1980). His neocognitron is an elegant example of how piecewise combinations of features can lead to comprehensive translation and scale invariance while at the same time retaining object specificity, thereby avoiding one form of the feature-binding problem. Fukushima's ideas accord well both with elements of the known neurophysiology of the ventral stream and a view-based scheme of object representation, and has inspired a whole series of models (see Wallis and Rolls, 1997; Riesenhuber and Poggio, 2000). The Riesenhuber and Poggio paper also describes their development of Fukushima's model and how it predicts the use of a nonlinear weighting mechanism on the inputs to neurons of each layer. Wallis and Rolls describe their own model, which is once again hierarchical and convergent but simpler in structure. Despite its simplicity, it has been shown to be able to learn invariant representations of objects without recourse to nonlocal learning mechanisms, supervised learning, specialist neural populations, or specific, prescribed connectivity. An important omission from such models is any explicit representation of object structure. As mentioned earlier, structure may well be important for higher levels of categorization, for distinguishing broad categories such as insects from mammals. Image-based approaches, on the other hand, are probably of more importance for within-category discrimination, such as discriminating a Peacock butterfly from a Meadow Brown. For more on this and related issues, see OBJECT RECOGNITION.

Another aspect that the hierarchical feedforward models lack is an account of the effects of top-down information due to expectation or selective attention. As such they really only deal with recognition within the high-acuity center of the visual field and would require some other mechanism for locating and fixating objects. One hierarchical model that does consider this has been described by Olshausen, Anderson, and van Essen (1993). It selects targets by controlling the breadth and number of pathways present in the

model's hierarchy. Recognition is achieved using a classical object-matching algorithm, which suffers from the disadvantages noted earlier, but the model does provide insight into a possible mechanism for object selection, and with it an additional solution to the problem of translation and scale invariance.

Temporal Order

Although it is possible to conceive of the ventral stream building features to represent individual views of objects, the question still remains as to how neurons learn to treat their preferred feature as the same, irrespective of size or location. Indeed, ultimately, one would like to understand how neurons learn to recognize objects as they undergo nontrivial transformations, perhaps due to changes in viewing direction or lighting.

One solution to this problem is to assume that each neuron receives some external information as to the identity of a particular stimulus. Of course, this simply begs the question of where this information originates in the first place. To describe a potential solution, it is worth reflecting on what clues our environment gives us about how to associate the stream of images that we see in everyday life. Recently, several theorists have argued that our natural environment provides a temporal cue to object identity. This cue emerges from the simple fact that we often study objects for extended periods. This then provides us with a simple heuristic for deciding how to associate novel images of objects with stored object representations. Since objects are often seen over extended periods, any unrecognized view coming straight after a recognized one is most probably of the same object. This heuristic will work as long as accidental associations from one object to another are random and associations from one view of an object to another are experienced regularly. There is every reason to suppose that this is actually what will happen under normal viewing conditions, and that by approaching an object, watching it move, or rotating it in our hand we will receive a consistent associative signal capable of bringing all of the views of the object together.

It was Miyashita (1993) who discovered that many neurons within IT cortex had developed selectivity for small sets of fractal images that he had been using in a short-term memory task. Although this task did not explicitly require the overall test sequence to be remembered, Miyashita noted that these neurons consistently responded well to single images that neighbored one another in the test sequence. For example, one neuron might respond preferentially to images 5, 6, and 7, whereas another neuron responded to images 37, 38, and 39. The fact that the images were generated randomly meant that there was no particular reason—on the grounds of spatial similarity—why these images should have become associated together by a single neuron. Instead, the results indicate the importance of temporal order in controlling the learning of neural selectivity. Recent studies of human recognition learning have found evidence for such a mechanism as well (Wallis and Bülthoff, 2001). Taken together, the two sources of evidence provide important preliminary support for the temporal association hypothesis (Wallis and Bülthoff, 1999). Several network models have made successful use of the temporal cue to view association, and it forms the core of learning in the model described by Wallis and Rolls (1997).

Discussion

This article has reviewed much of the current thinking on object recognition. In particular, it has proposed the presence of a distributed, view-based representation in which objects are recognized on the basis of multiple, 2D feature-selective neurons. Specialist cells appear to play a role in associating such feature combinations into

certain nontrivial image transformations, coding for a certain percentage of all stimuli in a largely view-invariant manner. The article has also pointed to evidence that a convergent hierarchy is used to build invariant representations over several stages, and that at each stage lateral competitive processes are at work between the neurons.

We have argued that temporal association could act as a cue for associating views of objects. If such a mechanism exists, it could only work in the ventral stream, since it would *not* be appropriate in the dorsal visual system, where motion and location are processed (Ungerleider and Haxby, 1994). Indeed, the importance of using temporal association in invariant object recognition, and the importance of not making such associations in the part of the visual system involved in processing motion and location, might be a fundamental reason for keeping these two processing streams apart.

We have touched on the analysis of visual scenes both within and beyond the ventral stream. Although much has been said about the roles of the parietal and temporal lobes, relatively little has been said about the frontal lobe. We do know that it acts as a temporary or working memory store, and that neurons within the frontal lobe are responsive to combinations of both where and what an object is. It may well turn out that the frontal lobe acts as a running store of objects currently being represented within a scene (Rensink, 2000). A challenge for models in the future will be to integrate the frontal lobe into the overall picture of scene analysis.

Road Map: Vision

Related Reading: Cortical Hebbian Modules; Fast Visual Processing; Object Recognition; Object Structure, Visual Processing; Visual Scene Perception, Neurophysiology

References

- Bülthoff, H., and Edelman, S., 1992, Psychophysical support for a two-dimensional view interpolation theory of object recognition, *Proc. Natl. Acad. Sci. USA*, 92:60–64. ♦
- Fukushima, K., 1980, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biol. Cybern.*, 36:193–202.
- Logothetis, N. K., and Sheinberg, D. L., 1996, Visual object recognition, *Annu. Rev. Neurosci.*, 19:577–621. ♦
- Miyashita, Y., 1993, Inferior temporal cortex: Where visual perception meets memory, *Annu. Rev. Neurosci.*, 16:245–263. ♦
- Olhausen, B. A., Anderson, C. H., and van Essen, D. C., 1993, A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information, *J. Neurosci.*, 13:4700–4719.
- Rensink, R. A., 2000, The dynamic representation of scenes, *Vis. Cognit.*, 7:17–42. ♦
- Riesenhuber, M., and Poggio, T., 2000, Models of object recognition, *Nature Neurosci.*, 3:1199–1204.
- Rolls, E. T., 1992, Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical areas, *Philos. Trans. R. Soc. Lond. B*, 335:11–21.
- Tanaka, K., Saito, H., Fukada, Y., and Moriya, M., 1991, Coding visual images of objects in the inferotemporal cortex of the macaque monkey, *J. Neurophysiol.*, 66:170–189.
- Thorpe, S., Fize, D., and Marlot, C., 1996, Speed of processing in the human visual system, *Nature*, 381:520–522.
- Ungerleider, L. G., and Haxby, J. V., 1994, “What” and “where” in the human brain, *Curr. Opin. Neurobiol.*, 4:157–165. ♦
- Wallis, G., and Bülthoff, H. H., 1999, Learning to recognize objects, *Trends Cognit. Sci.*, 3:22–31. ♦
- Wallis, G., and Bülthoff, H. H., 2001, Effects of temporal association on recognition memory, *Proc. Natl. Acad. Sci.*, 98:4800–4804.
- Wallis, G., and Rolls, E. T., 1997, A model of invariant object recognition in the visual system, *Progr. Neurobiol.*, 51:167–194.
- Webster, M. J., Bachevalier, J., and Ungerleider, L. G., 1994, Connections of inferior temporal areas TEO and TE with parietal and frontal-cortex in macaque monkeys, *Cerebr. Cortex*, 4:470–483.

Object Structure, Visual Processing

Shimon Edelman and Nathan Intrator

A Functional Characterization of Structure Processing

A computational-level analysis of the processes dealing with object and scene structure requires that we first identify the common functional characteristics of structure-related behavioral tasks. In other problems in high-level vision, effective functional characterization typically led to advances in the computational understanding, and to better modeling, of the relevant aspects of the human visual system. In the study of visual motion, for example, realization of the central role of the correspondence problem constituted just such an advance. Likewise, object recognition tasks, such as identification or categorization, have at their core a common operation, namely, the matching of the stimulus against a stored memory trace.

For the structure-processing tasks, a good candidate for the signature common characteristic is the *restriction of the spatial scope* of at least some of the operations involved to some fraction of the visual extent of the object or scene under consideration. In other words, a task should qualify for the label “structural” only if it calls for a separate treatment of some *fragment(s)* of the stimulus (and not merely of the whole). Here are a few examples of behavioral tasks that qualify as structural according to this criterion:

- *Given two objects, or an object and a class prototype, identify their corresponding regions.* The correspondence here may be based on local shape similarity (find the eyes of a face in a Cubist painting), or on similar role played by the regions in the global structure (find the eyes in a smiley icon).
- *Given an object and an action, identify a region in the object toward which the action can be directed.* Similarities between objects vis à vis this task are defined functionally (as in the parallel that can be drawn between the handle of a pan and a door handle: both afford grasping).
- *Given an object, describe its structure.* This explicitly structural task arises in the context of trying to make sense of an unfamiliar object (as in perceiving a hot-air balloon, upon seeing it for the first time, as a pear-like shape over a box-like one).

The characterization of structure processing in terms of scope-restricted spatial analysis mechanisms has two immediate implications. Consider, on the one hand, the *appearance-based* computational approaches to recognition and categorization, according to which objects are represented by collections of entire, spatially unanalyzed views. Because of the holistic nature of the representations they rely on, these approaches are seen to be incapable, in principle, of supporting structure processing (Hummel, 2000). On the other hand, the “classical” *structural decomposition* approaches (Biederman, 1987) have the opposite tendency: the recursive symbolic structure they impose on objects seems too rigid and too elaborate, compared to the basic principle of spatial analysis proposed above, which requires merely that the spatial scope of each of its operators be limited to a fragment of the visual scene.

Object Form Processing in Computer Vision

Until recently, the attainment of classical structural descriptions has been widely considered to be the ultimate goal of object form processing in computer vision. The specific notion that the structural descriptions are to be expressed in terms of volumetric parts, popularized by Marr (1982), was subsequently adopted by Biederman (1987), who developed it into a (psychological) theory of recog-

nition by components (RBC). In Biederman’s formulation, the representation is explicitly *compositional*: it consists of symbols that stand for generic parts (called “geons”); the symbols are drawn from a small repertoire and are bound together by categorical symbolically coded relations (such as “above” or “to the left of”). RBC’s compositional nature is explicit in a sense stressed by Fodor and McLaughlin (1990): a classical structural description of an entire object necessarily contains *tokenings* of its (stipulated) constituent parts, in the same sense that a sentence considered as a concatenation of some words necessarily contains each and every of its words in their original, unchanged format (see COMPOSITIONALITY IN NEURAL SYSTEMS).

By virtue of their compositionality, the classical structural descriptions meet the two main challenges in the processing of structure: productivity and systematicity. A visual system is productive if it is open-ended, that is, if it can deal effectively with a potentially infinite set of objects. A visual representation is systematic if a well-defined change in the spatial configuration of the object, such as swapping top and bottom parts, causes a principled change in the representation, such as the interchange of the representations of top and bottom parts. Compositionality, however, has its cost. The requirement that object parts be “crisp” and relations syntactically compositional is a principle that may be appealing (by analogy with an intuitive view of the language faculty that it embodies), but is difficult to adhere to in practice. Indeed, in computer vision, a panel of experts deemed the structural analysis of raw images (as opposed to the analysis of symbolically specified line drawings of Biederman’s examples) to be unpromising: “the principal problems with this approach seem to be the difficulty in extracting sufficiently good line drawings, and the idealized nature of the geon representation” (Dickinson et al., 1997, p. 284).

Both of these problems can be effectively neutralized by giving up the classical compositional representation of shape by a fixed alphabet of crisp “all-or-none” explicitly tokened primitives (such as geons) in favor of a fuzzy, superpositional coarse-coding by an open-ended set of image fragments. This alternative approach has met with considerable success in computer vision. For example, the system described by Nelson and Selinger (1998) starts by detecting contour segments, then determines whether their relative arrangement approximates that of a model object. Because none of the individual segment shapes or locations is critical to the successful description of the entire shape, this method does not suffer from the brittleness associated with the classical structural description models of recognition. Moreover, the tolerance for moderate variation in the segment shape and location data allows it to categorize novel members of familiar object classes (Nelson and Selinger, 1998).

In a similar fashion, the method of Burl, Weber, and Perona (1998) combines “local photometry” (shape primitives that are approximate templates for small snippets of images) with “global geometry” (the probabilistic quantification of spatial relations between pairs or triplets of primitives). In general, such methods use snippets of images taken from objects to be recognized to represent these objects; recognition is declared if at least some of the fragments are reliably detected, and if the spatial relations among these fragments conform to the stored description of the target. In all of these methods, the interplay of loosely defined local shape (“what”) and approximate location (“where”) information leads to robust algorithms supporting both recognition and categorization. These same methods may also lead to the development of an effective alternative to the classical structural description approach to object

form, provided that they can be extended to support hierarchical treatment of shape details across spatial scales.

Mechanisms Implicated in Structure Processing in Primate Vision

In theoretical neuroscience, ideas advanced to explain structure processing by the primate visual system can be roughly divided into two groups, following the distinction made in the preceding discussion between the classical crisp part-based compositional methods and the fuzzy fragment-based *what + where* approach. The two kinds of theories invoke distinct neural mechanisms to explain the manner in which object constituents (whether explicitly tokened crisp parts or fuzzy superimposed fragments) are (1) represented individually and (2) bound together to form the whole.

Consider the classical theories built around the syntactic compositionality idea (Biederman, 1987). First, these theories require that “crisply” defined geon-like parts and categorical relations be explicitly represented on the neural level. Although no evidence seems to exist for the neural embodiment of geons as such, there are reports that cells in the inferotemporal (IT) cortex exhibit a higher sensitivity to “nonaccidental” visual features than to “metric” properties of the stimuli; nonaccidental features such as curvature sign or parallelism of contours are used to define geons, because of their diagnosticity and invariance to viewpoint changes. Second, the classical theories hold that symbols representing the parts are bound into the proper structure dynamically, by the synchronous firing of the neurons that code each symbol. Thus, a mechanism capable of supporting dynamic binding must be available; it is possible that this function is fulfilled by the synchronous or phase-locked firing of cortical neurons (see SYNCHRONIZATION, BINDING, AND EXPECTANCY), although the status of this phenomenon in primates has been disputed.

The alternative theory, proposed by Edelman and Intrator (2000), calls for an open-ended set of fuzzy fragments instead of geons. The role of fragment detectors may be fulfilled by those neurons in the IT cortex that respond selectively to some particular views of an object or to a specific shape irrespective of view (Logothetis and Sheinberg, 1996). This very kind of shape-selective response may also constitute the neural basis of *binding by retinotopy*, an idea based on the observation (Edelman, 1994) that the visual field itself can serve as the frame encoding the relative positions of object fragments, simply because each such fragment is already localized within that frame when it is detected. The binding by retinotopy is possible if the receptive field of each cell is confined to some relatively limited portion of the entire visual field (as per the definition of the signature characteristic of structural processing proposed in the Introduction). Neurons with such response properties have been found in the IT cortex (Op de Beeck and Vogels, 2000) and in the prefrontal cortex, where they were called *what + where* cells (Rao, Rainer, and Miller, 1997).

Neuromorphic Models of Visual Structure Processing

We now proceed to outline two implemented models of structure representation in primate vision. The first of these, JIM.3 (Hummel, 2001), exemplifies the classical compositional approach, and the second, Chorus of Fragments, or CoF (Edelman and Intrator, 2000), the alternative one just discussed.

The JIM.3 model is structured as an eight-layer network (Figure 1). The first three layers extract local features: contours, vertices and axes of symmetry, and surface properties. Surfaces are represented in terms of five categorical properties: (1) elliptical or not; (2) possessing parallel, expanding, convex, or concave axes of symmetry; (3) possessing a curved or a straight major axis; (4) truncated or pointed; and (5) planar or curved in 3D. Units coding these

local features group themselves into representations of geons by synchrony of firing. These representations are then routed by the units of layer 4 to two distinct destinations in layer 5. The first of these is a population of units coding for geons and spatial relations that are independent or “disembodied” in the sense that each of them may have originated from any location within the image. Within this population, the emergence of a representation of the object’s structure requires dynamic binding, which the model stipulates to be carried out under attentional guidance and to take a relatively long time (a few hundred milliseconds).

The second destination of the outgoing connections of layer 4 is a population of geon units arranged in the form of a retinotopic map. Here, the relations between the geons are coded implicitly, by virtue of each representation unit residing in the proper location within the map, which reflects the location of the corresponding geon in the image. In contrast to the attention-controlled stream, this one can operate much faster, and is postulated to be able to form a structural representation in a few tens of milliseconds. This speed and automaticity have a price: because of the fixed spatial structure imposed by the retinotopic map, the representation this stream supports is more sensitive to object transformations such as rotation in depth and reflection (Hummel, 2001).

The other implemented model we outline here, the Chorus of Fragments (CoF) model, exemplifies the coarse-coded fragment-based approach to the representation of structure (Edelman and Intrator, 2000, 2001). It simulates cells with *what + where* receptive fields to represent object fragments and uses attentional *gain fields*, such as those found in area V4 (Connor et al., 1997), to decouple the representation of object structure from its location in the visual field (the gain field of a neuron refers to those locations where the presence of a secondary stimulus modulates the cell’s response to the primary stimulus shown within the classical receptive field; in this case, the modulation is exerted by shifting the focus of attention).

Unlike JIM.3, the CoF system operates directly on gray-level images, preprocessed by a front end that is a rough simulation of the primary visual cortex. The system illustrated in Figure 2 contains two *what + where* units, one (labeled “above center”) responsible for the top fragment of the object (as extracted by an appropriately configured Gaussian gain field), and the other (labeled “below center”) responsible for the bottom fragment. The units are trained jointly for three-way discrimination, for translation tolerance, and for autoassociation. Figure 3 shows the performance of a CoF system charged with learning to reuse fragments of the members of the training set (three bipartite objects composed of numeral shapes) in interpreting novel composite objects. The gain field mechanism allowed it to respond largely systematically to the learned fragments shown in novel locations, both absolute and relative.

The CoF model offers a unified framework for understanding the functional significance of *what + where* receptive fields and of attentional gain modulation. It extends the previous use of gain fields in the modeling of translation invariance, and highlights a parallel between *what + where* cells and probabilistic fragment-based approaches to structure representation in computer vision, such as that of Burl et al. (1998). The representational framework it embodies is both productive and effectively systematic. It is capable, as a matter of principle, of recognizing such objects that are related through a rearrangement of “middle-scale” fragments, without the need for dynamic binding, and without being taught those fragments individually. When coupled with statistical inference methods such as the Minimum Description Length principle, this model may be capable of unsupervised learning of useful fragments, an issue that is currently under investigation (Edelman and Intrator, 2001). Further testing is also needed to determine whether or not the CoF model can be scaled up to learn larger collections

Figure 1. The architecture of the JIM.3 model (Hummel, 2001). The model had been trained on a single view (actually, a line drawing) of each of 20 objects—hammer, scissors, and so on—as well as on some “nonsense” objects. It was then tested on translated, scaled, reflected, and rotated (in the image plane) versions of the same images. The model exhibited a pattern of results consistent with a range of psychophysical data obtained from human subjects (Hummel, 2001). Specifically, the categorization performance was invariant with respect to translation and scaling, and was reduced by rotation. Moreover, because of the dual nature of the binding process in JIM.3—dynamic and static/retinotopic—the model behaved differently when given attended and unattended objects: reflected images primed each other in the former case, but not in the latter case. (Figure courtesy of J. E. Hummel.)

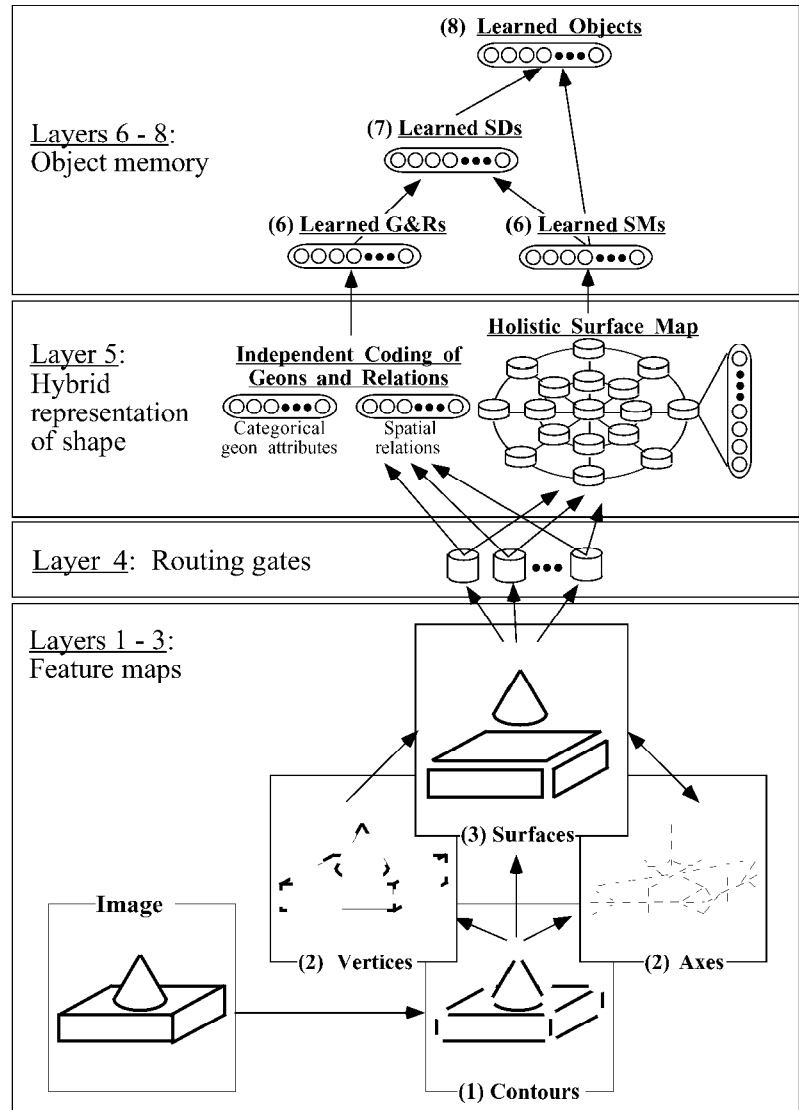
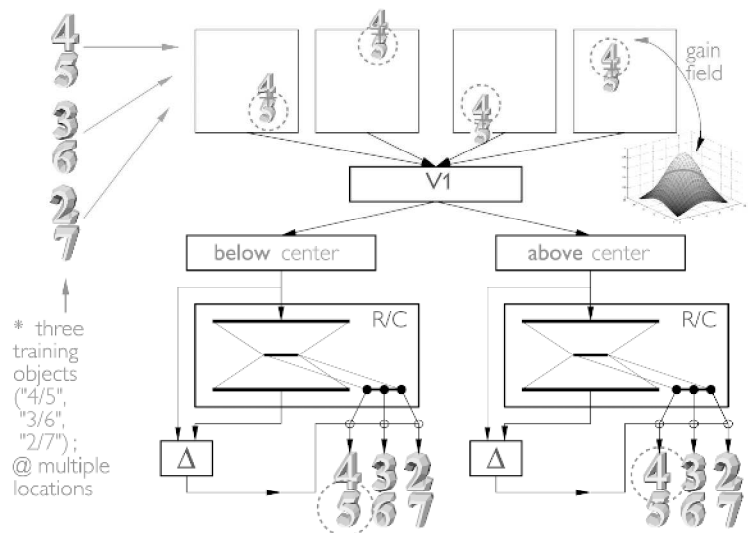


Figure 2. The CoF model, trained on three composite objects (numerals 4 over 5, 3 over 6, and 2 over 7). The model consists of two *what* + *where* units, responsible for the top and the bottom fragments of the stimulus, respectively. Gain fields (boxes labeled “below center” and “above center”) steer each input fragment to the appropriate unit. The learning mechanism (R/C, for Reconstruction and Classification) can be implemented either as a multilayer perceptron or as a radial basis function network. The reconstruction error (Δ) modulates the classification outputs and helps the system learn binding (a co-activation pattern over units of the preceding stage will have a small reconstruction error only if both its *what* and *where* aspects are correct).



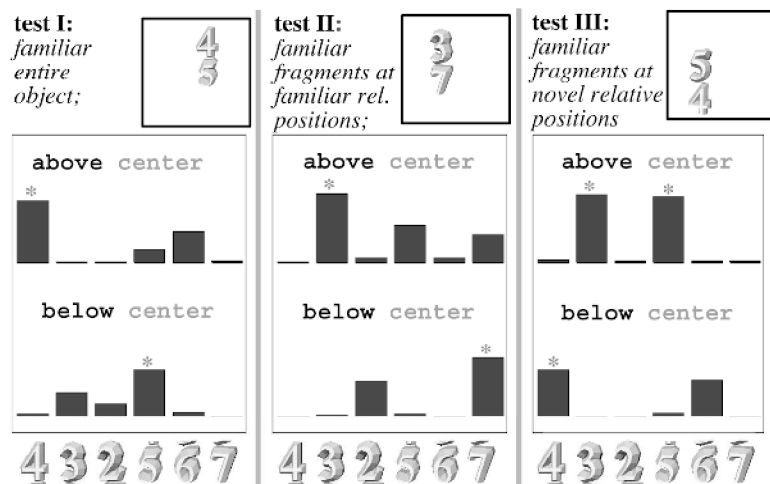


Figure 3. The response of the CoF model to a familiar composite object at a novel location (test I) and to novel compositions of fragments of familiar objects (tests II and III). In the test scenario, each unit ("above" and "below") must be fed each of the two input fragments ("above" and "below"); hence the 12 bars in the plots of the model's output.

of objects, and to represent finer structure, under realistic transformations such as rotation in depth.

Conclusions

For decades, the prevalent "classical" theory of visual structure processing has been rooted in the perceived computational need for the structure of an object to be "made explicit" to enable its recognition (Marr, 1982), and by the apparent uniqueness of the compositional solution to the problems of productivity and systematicity (Fodor and McLaughlin, 1990).

The first of these two issues is made moot by recent advances in computer vision, which indicate that neither recognition nor categorization requires a prior derivation of a classical structural description. Moreover, making structure explicit may not be a good idea, either from a philosophical viewpoint or from a practical one. On the philosophical level, it embodies a gratuitous ontological commitment to the existence of object parts, which are presumed to be waiting for detection by the visual system; on the practical level, reliable detection of such parts has proved to be an elusive goal. The second issue, focusing on productivity and systematicity of structure processing, is also being transformed at present by claims that a system can be productive and systematic without relying on representations that are compositional in the classical sense (Edelman and Intrator, 2000).

The alternative stance on these issues, discussed in the preceding sections, holds that structure can be represented by a coarse code based on image fragments that are bound together by retinotopy. This notion is supported by the success of computer vision methods (such as "local photometry, global geometry"), by data from neurophysiological studies in primates (such as the discovery of *what* + *where* cells), and by psychological findings and metatheoretical considerations not mentioned in this article (Edelman and Intrator, 2000). In the field of neuromorphic modeling, these developments have brought about a curious convergence between an approach initially grounded in classical structural description theory (Hummel, 2001) and that derived from a holistic view of object representation (Edelman and Intrator, 2001). In this rapidly changing field, the theoretical and factual aspects of structure processing (but, we believe, not the metatheoretical ones) are likely to require reconsideration on a regular basis.

Road Map: Vision

Background: Feature Analysis

Related Reading: Object Recognition; Object Recognition, Neurophysiology; Synchronization, Binding and Expectancy

References

- Biederman, I., 1987, Recognition by components: A theory of human image understanding, *Psychol. Rev.*, 94:115–147.
- Burl, M. C., Weber, M., and Perona, P., 1998, A probabilistic approach to object recognition using local photometry and global geometry, in *Proceedings of the 4th European Conference on Computing and Vision* (H. Burkhardt and B. Neumann, Eds.), LNCS series, vol. 1406–1407, Berlin: Springer-Verlag, pp. 628–641.
- Connor, C. E., Preddie, D. C., Gallant, J. L., and Van Essen, D. C., 1997, Spatial attention effects in macaque area V4, *J. Neurosci.*, 17:3201–3214.
- Dickinson, S., Bergevin, R., Biederman, I., Eklundh, J., Munck-Fairwood, R., Jain, A., and Pentland, A., 1997, Panel report: The potential of geons for generic 3-d object recognition, *Image Vision Comput.*, 15:277–292.
- Edelman, S., 1994, Biological constraints and the representation of structure in vision and language, *Psychology*, 5(57). FTP host: ftp.princeton.edu; FTP directory: /pub/harnad/Psycology/1994.volume.5/; file name: psyc.94.5.57.language-network.3.edelman.
- Edelman, S., and Intrator, N., 2000, (Coarse coding of shape fragments) + (retinotopy) ~ representation of structure, *Spat. Vision*, 13:255–264.
- Edelman, S., and Intrator, N., 2001, A productive, systematic framework for the representation of visual structure, in *Advances in Neural Information Processing Systems 13* (T. K. Leen, T. G. Dietterich, and V. Tresp, Eds.), Cambridge, MA: MIT Press, pp. 10–16.
- Fodor, J., and McLaughlin, B., 1990, Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work, *Cognition*, 35:183–204.
- Hummel, J. E., 2000, Where view-based theories of human object recognition break down: The role of structure in human shape perception, in *Cognitive Dynamics: Conceptual Change in Humans and Machines* (E. Dietrich and A. Markman, Eds.), Hillsdale, NJ: Erlbaum, chap. 7.
- Hummel, J. E., 2001, Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition, *Vis. Cognit.*, 8:489–517.
- Logothetis, N. K., and Sheinberg, D. L., 1996, Visual object recognition, *Annu. Rev. Neurosci.*, 19:577–621. ♦
- Marr, D., 1982, *Vision*, San Francisco, CA: Freeman. ♦
- Nelson, R. C., and Selinger, A., 1998, Large-scale tests of a keyed, appearance-based 3-D object recognition system, *Vision Res.*, 38:2469–2488.
- Op de Beeck, H., and Vogels, R., 2000, Spatial sensitivity of macaque inferior temporal neurons, *J. Comp. Neurol.*, 426:505–518.
- Rao, S. C., Rainer, G., and Miller, E. K., 1997, Integration of what and where in the primate prefrontal cortex, *Science*, 276:821–824.

Ocular Dominance and Orientation Columns

Kenneth D. Miller

Introduction

The classic example of activity-dependent neural development is the formation of ocular dominance columns in the cat or monkey primary visual cortex (reviewed in Miller and Stryker, 1990; Crair et al., 2001; Katz and Crowley, 2002). The primary visual cortex (V1) receives signals from the lateral geniculate nucleus of the thalamus (LGN), which in turn receives input from the retinas of the two eyes (Figure 1).

To describe ocular dominance columns, several terms must be defined. First, the *receptive field* of a cortical cell refers to the area on the retina in which appropriate light stimulation evokes a response in the cell, and also to the pattern of light stimulation that evokes such a response. Second, a *column* is defined as follows. V1 extends many millimeters in each of two “horizontal” dimensions. Receptive field positions vary continuously along these dimensions, forming a *retinotopic* map, a continuous map of the visual world. In the third, “vertical” dimension, the cortex is about 2 mm in depth, and consists of six layers. Receptive field positions do not significantly vary through this depth. Such organization, in which cortical properties are invariant through the vertical depth of cortex but vary horizontally, is called *columnar* organization and is a basic feature of cerebral cortex. Finally, *ocular dominance*, or eye preference, describes the degree to which a cortical cell’s responses are better driven by stimulation of one eye or the other. Like retinotopy, ocular dominance has a columnar organization: alternating stripes or patches of cortex are dominated throughout the cortical depth by a single eye, and are known as *ocular dominance columns*.

The anatomical basis for ODCs is the segregated pattern of termination of the LGN inputs to V1 (Figures 1 and 2A). Inputs serving a single eye terminate in alternating stripes or patches of cortex. This segregation arises early in development. Although the exact time at which ocular dominance columns emerge is currently controversial (compare Crair et al., 2001, with Katz and Crowley, 2002, noting that 2 weeks of age in cat corresponds developmentally to about 5 weeks of age in ferret), it is clear that they begin

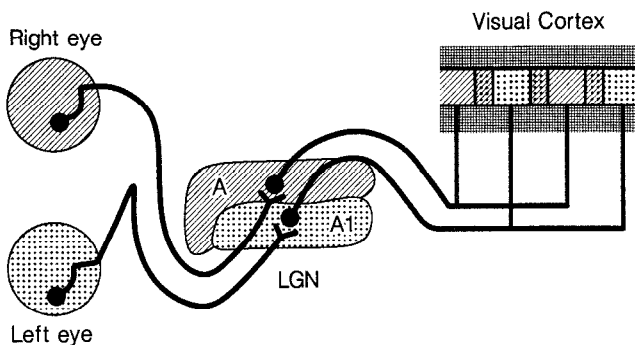


Figure 1. Schematic of the mature visual system. Retinal ganglion cells from the two eyes project to separate layers of the lateral geniculate nucleus (LGN). Neurons from these two layers project to separate patches or stripes within visual cortex (V1). Binocular regions (receiving input from both eyes) are depicted at the borders between the eye-specific patches. (From Miller, K. D., Keller, J. B., and Stryker, M. P., Ocular column dominance development: Analysis and simulation, *Science*, 245:605–615. © 1989 by the AAAS. Reprinted with permission.)

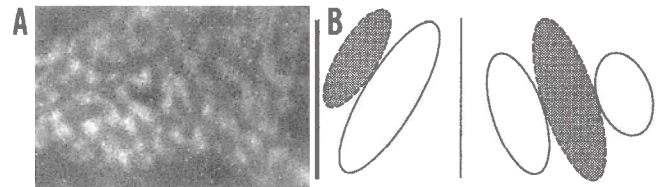


Figure 2. A, Ocular dominance columns from cat V1. A horizontal cut through the LGN-recipient layer of V1 is shown. Terminals serving a single eye are labeled white. Dark regions at edges are out of plane containing LGN terminals. Region shown is 5.3×7.9 mm. (Photograph courtesy of Dr. Y. Hata.) B, Two examples of simple cell-receptive fields (RFs). Regions of the visual field from which a simple cell receives ON-center (white) or OFF-center (dark) input are shown. Note: Ocular dominance columns (A) represent an alternation, across cortex, in the type of input (left or right eye) received by different cortical cells, while a simple cell RF (B) represents an alternation across visual space in the type of input (ON- or OFF-center) received by a single cortical cell.

to form at least a week before the beginning of the critical period. The *critical period* is the time during which the structure of the ocular dominance columns is susceptible to alteration by altered patterns of visual experience. For example, closure of one eye during the critical period will cause that eye’s patches to shrink and the open eye’s patches to expand.

A longstanding hypothesis is that the formation of ocular dominance columns results from an activity-instructed competition between the geniculate terminals serving the two eyes. Under this hypothesis, the signal indicating that different terminals represent the same eye is the correlation in their neural activities (reviewed in Weliky, 2000). These correlations exist because of both spontaneous activity, which is locally correlated within each retina, and visually induced activity, which correlates the activities of retinotopically nearby neurons within each eye and, to a lesser extent, between the eyes. An attractive feature of this hypothesis is that it suggests that the same mechanisms underlie both the formation of ocular dominance columns and their subsequent susceptibility to alteration by visual experience. However, one must then explain why ocular dominance columns begin forming *before* the beginning of the critical period, that is, at a time when alterations in visual experience do not influence ocular dominance column development. A possible explanation (Crair et al., 2001) is that before the critical period, the loop between cortex and LGN may dominate over the retinal input in determining LGN activities (see Weliky, 2000), so that alterations in retinal activity would have minimal effect on activity-instructed development of connections from LGN to cortex. Alternatively, it has been proposed that the initial formation of ocular dominance columns may be guided by molecular cues in cortex that label the regions appropriate for each eye’s innervation (no such molecules have yet been found), and that activity only comes to play an instructive role in ocular dominance column development at the onset of the critical period (Katz and Crowley, 2002). In this article, we shall consider models of ocular dominance column formation under the first scenario, in which the initial formation of ocular dominance columns is instructed by activity.

At least during the critical period, the process guiding the segregation of ocular dominance columns is *competitive*—the two eyes compete for cortical territory based on their patterns of activity

(reviewed in Miller and Stryker, 1990). If one eye is caused to have less activity than the other during the critical period, the more active eye takes over most of the cortical territory; but the eye with reduced activity suffers no loss of projection strength in retinotopic regions in which it lacks competition from the other eye. Thus, the fate of one eye's projection is not determined solely by its own activity, but rather by its activity *relative* to that of the opposite eye.

Orientation columns are another striking feature of visual cortical organization. Most V1 cells are orientation selective, responding selectively to light/dark edges over a narrow range of orientations. The preferred orientation of cortical cells varies regularly and periodically across the horizontal dimension of cortex, and is invariant in the vertical dimension. The initial development of orientation selectivity often begins before eye opening and is unaffected by whether the eyes are open or closed, but depends on normal spontaneous patterns of neural activity, while the later maturation of orientation selectivity depends on vision (reviewed in Miller, Erwin, and Kayser, 1999). The dependence of orientation selectivity on vision begins at about the same time that the critical period for ocular dominance plasticity begins, suggesting that a single set of changes renders both systems vulnerable to visual experience.

The inputs from LGN to V1 serving each eye are of two types: ON-center and OFF-center. Both kinds of cells have circularly symmetric receptive fields that are relatively insensitive to stimulus orientation, and respond to contrast rather than uniform luminance. ON-center cells respond to light against a dark background, or to light onset; OFF-center cells respond to dark against a light background, or to light offset. In the cat, the orientation-selective V1 cells that receive the bulk of LGN input are *simple cells*: cells with receptive fields consisting of alternating oriented subregions that receive exclusively ON-center or exclusively OFF-center input (Figure 2B). One theory for the development of orientation selectivity is that, like ocular dominance, it develops through a competition between two input populations: in this case, a competition between the ON-center and the OFF-center inputs (Miller, 1994).

Correlation-Based Models

To understand ocular dominance and orientation column formation, two processes must be understood:

1. The development of *receptive field structure*: Under what conditions do receptive fields become monocular (drivable only by a single eye) or orientation selective?
2. The development of *periodic cortical maps* of receptive field properties: What leads ocular dominance or preferred orientation to vary periodically across the horizontal dimensions of cortex, and what determines the periodic length scales of these maps?

Typically, the problem is simplified by consideration of a two-dimensional model cortex, ignoring the third dimension, in which properties such as ocular dominance and orientation are invariant.

One approach to addressing these problems is to begin with a hypothesized mechanism of synaptic plasticity and to study the outcome of cortical development under such a mechanism. Most commonly, theorists have considered a Hebbian synapse (see HEBBIAN SYNAPTIC PLASTICITY): a synapse whose strength is increased when pre- and postsynaptic firing are correlated, and possibly decreased when they are anticorrelated. Other mechanisms can lead to similar dynamics, in which synaptic plasticity depends on the correlations among the activities of competing inputs. We refer to models based on such mechanisms as correlation-based models.

Von der Malsburg's Model of V1 Development

Von der Malsburg (1973; von der Malsburg and Willshaw, 1976) first formulated a correlation-based model for the development of visual cortical receptive fields and maps. His model had two basic elements. First, synapses of LGN inputs onto cortical neurons were modified by a Hebbian rule that is *competitive*, so that some synapses were strengthened only at the expense of others. He enforced the competition by holding constant the total strength of synapses converging on each cortical cell (conservation rule). Second, cortical cells tended to be activated in *clusters*, due to intrinsic cortical connectivity, e.g., short-range horizontal excitatory connections and longer-range horizontal inhibitory connections.

The conservation rule leads to competition among the inputs to a single target cell. Inputs that tend to be coactivated—that is, that have correlated activities—are mutually reinforcing, working together to activate the postsynaptic cells and thus to strengthen their own synapses. Different patterns that are mutually un- or anticorrelated compete, since strengthening of some synapses means weakening of others. Cortical cells eventually develop receptive fields responsive to a correlated pattern of inputs.

The clustered cortical activity patterns lead to competition between different groups of cortical cells. Each input pattern comes to be associated with a cortical cluster of activity. Overlapping cortical clusters contain many coactivated cortical cells, and thus become responsive to overlapping, correlated input patterns. Adjacent, nonoverlapping clusters contain many anticorrelated cortical cells, and thus become responsive to un- or anticorrelated input patterns. Thus, over distances on the scale of an activity cluster, cortical cells will have similar response properties, while on the scale of the distance between nonoverlapping clusters, cortical cells will prefer un- or anticorrelated input patterns. This combination of local continuity and larger-scale heterogeneity leads to continuous, periodic cortical maps of receptive field properties.

In computer simulations, this model was applied to the development of orientation columns (von der Malsburg, 1973) and ocular dominance columns (von der Malsburg and Willshaw, 1976). For orientation columns, inputs were activated in oriented patterns, each pattern consisting of a stripe of inputs through a common center position. Individual cortical cells then developed selective responses corresponding to one such oriented pattern, with nearby cortical cells preferring nearby orientations. For ocular dominance columns, inputs were activated in monocular patterns consisting of a localized set of inputs from a single eye. Individual cortical cells came to be driven exclusively by a single eye, and clusters of cortical cells came to be driven by the same eye. The final cortical pattern consisted of alternating stripes of cortical cells preferring a single eye, with the width of a stripe approximately set by the diameter of an intrinsic cluster of cortical activity.

In summary, a competitive Hebbian rule leads individual receptive fields to become selective for a correlated pattern of inputs. Combined with the idea that the cortex is activated in intrinsic clusters, this suggests an origin for cortical maps: coactivated cells in a cortical cluster tend to become selective for similar, coactivated patterns of inputs. These basic ideas are used in most subsequent models.

Mathematical Formulation

A typical correlation-based model is mathematically formulated as follows (von der Malsburg, 1973; Linsker, 1986; Miller and Stryker, 1990). Let x, y, \dots represent retinotopic positions in V1, and let α, β, \dots represent retinotopic positions in the LGN. Let $S^\mu(x, \alpha)$ be the synaptic strength of the connection from α to x of the LGN projection of type μ , where μ may signify left eye, right eye, ON-center, OFF-center, etc. Let $B(x, y)$ represent the synaptic

strength and sign of connection from the cortical cell at y to that at x . For simplicity, $B(x, y)$ is assumed to take different signs for a fixed y as x varies, but alternatively, separate excitatory-projecting and inhibitory-projecting cortical neurons may be used. Let $a(x)$ and $a^\mu(\alpha)$ represent the activity of a cortical or LGN cell, respectively.

The activity $a(x)$ of a cortical neuron is assumed to depend on a linear combination of its inputs:

$$a(x) = f_1 \left(\sum_{\mu, \alpha} S^\mu(x, \alpha) a^\mu(\alpha) + \sum_y B(x, y) a(y) \right) \quad (1)$$

Here, f_1 is some monotonic function such as a sigmoid or linear threshold.

A Hebbian rule for the change in feedforward synapses can be expressed

$$\Delta S^\mu(x, \alpha) = A^\mu(x, \alpha) f_2[a(x)] f_3[a^\mu(\alpha)] \quad (2)$$

Here, $A(x, \alpha)$ is an “arbor function,” expressing the number of synapses of each type from α to x ; a minimal form is $A(x, \alpha) = 1$ if there is a connection from α to x , $A(x, \alpha) = 0$ otherwise. A typical form for the functions f_2 and f_3 is $f(a) = (a - \langle a \rangle)$, where $\langle a \rangle$ indicates an average of a over input patterns. This yields a *covariance rule*: synaptic change depends on the covariance of postsynaptic and presynaptic activity.

Next, the Hebbian rule must be made *competitive*. This can be accomplished by conserving total synaptic strength over the post-synaptic cell (von der Malsburg, 1973), which in turn may be done either subtractively or multiplicatively (Miller and MacKay, 1994). The corresponding equations are

$$\frac{d}{dt} S^\mu(x, \alpha) = \Delta S^\mu(x, \alpha) - \varepsilon(x) A(x, \alpha) \quad (\text{Subtractive}) \quad (3)$$

$$\frac{d}{dt} S^\mu(x, \alpha) = \Delta S^\mu(x, \alpha) - \gamma(x) S^\mu(x, \alpha) \quad (\text{Multiplicative}) \quad (4)$$

where $\varepsilon(x) = [\sum_{\kappa, \alpha} \Delta S^\kappa(x, \alpha)] / [\sum_{\kappa, \alpha} A(x, \alpha)]$, and $\gamma(x) = [\sum_{\kappa, \alpha} \Delta S^\kappa(x, \alpha)] / [\sum_{\kappa, \alpha} S^\kappa(x, \alpha)]$. Either form of constraint ensures that $\sum_{\mu, \alpha} (d/dt) S^\mu(x, \alpha) = 0$. Alternative approaches have been developed that lead Hebbian rules to be competitive (Miller and MacKay, 1994; Song, Miller, and Abbott, 2000).

Finally, synaptic weights may be limited to a finite range, $s_{\min} A(x, \alpha) \leq S^\mu(x, \alpha) \leq s_{\max} A(x, \alpha)$. Typically, $s_{\min} = 0$ and s_{\max} is some positive constant.

Semilinear Models

In semilinear models, the f s in Equations 1 and 2 are chosen to be linear. Then, after substituting for $a(x)$ from Equation 1 and averaging over input patterns (assuming that all inputs have identical mean activity and that changes in synaptic weights are negligibly small over the averaging time), Equation 2 becomes

$$\Delta S^\mu(x, \alpha) = \lambda A(x, \alpha) \sum_{y, \beta, \kappa} I(x - y) [C^{\mu\kappa}(\alpha - \beta) - k_2] \times S^\kappa(y, \beta) + k_1 A(x, \alpha) \quad (5)$$

Here, $I(x - y)$ is an element of the intracortical interaction matrix $\mathbf{I} \equiv (\mathbf{1} - \mathbf{B})^{-1} = \mathbf{1} + \mathbf{B} + \mathbf{B}^2 + \dots$, where the matrix \mathbf{B} is defined in Equation 1. This summarizes intracortical synaptic influences, including contributions via 0, 1, 2, ... synapses. The sum over κ is a sum over input types. The covariance matrix $C^{\mu\kappa}(\alpha - \beta) = \langle (a^\mu(\alpha) - \bar{a})(a^\kappa(\beta) - \bar{a}) \rangle$ expresses the covariation of input activities. The factors λ , k_1 , and k_2 are constants. Translation invariance has been assumed in both cortex and LGN.

When there are two competing input populations, Equation 5 can be further simplified by transforming to sum and difference variables: $S^S \equiv S^1 + S^2$, $S^D \equiv S^1 - S^2$. Assuming equivalence of the two populations (so that $C^{11} = C^{22}$, $C^{12} = C^{21}$), Equation 5 becomes

$$\Delta S^S(x, \alpha) = \lambda A(x, \alpha) \sum_{y, \beta} I(x - y) [C^S(\alpha - \beta) - 2k_2] \times S^S(y, \beta) + 2k_1 A(x, \alpha) \quad (6)$$

$$\Delta S^D(x, \alpha) = \lambda A(x, \alpha) \sum_{y, \beta} I(x - y) C^D(\alpha - \beta) S^D(y, \beta) \quad (7)$$

Here, $C^S \equiv C^{11} + C^{12}$, $C^D \equiv C^{11} - C^{12}$. A similar transformation to one sum and three difference coordinates can be made in the case of four competing input populations, such as ON and OFF inputs from left and right eyes (Erwin and Miller, 1998).

How Semilinear Models Behave

Linear equations like Equations 6 and 7 can be understood by finding the eigenvectors or “modes” of the operators on the right side of the equation. The eigenvectors are the synaptic weight patterns that grow independently and exponentially, each at its own rate. The fastest-growing eigenvectors typically dominate development and determine basic features of the final pattern, although the final pattern ultimately is stabilized by nonlinearities such as the limits on the range of synaptic weights or the nonlinearity involved in multiplicative renormalization (Equation 4).

I will focus on the behavior of Equation 7 for S^D . S^D describes the difference in the strength of two competing input populations. Thus, it is the key variable describing the development of ocular dominance segregation, or development under an ON-center/OFF-center competition. It also is sensible to imagine that S^D is initially small, and that the dynamics do not intrinsically favor either of the competing input types; under these assumptions, the initial development of S^D will be described by linear equations, giving some justification for studying linear equations for the development of S^D .

Equation 7 can be simply solved in the case of full connectivity from the LGN to the cortex, when $A(x, \alpha) \equiv 1$ for all x and α . Then modes of $S^D(x, \alpha)$ of the form $e^{ikx} e^{il\alpha}$ grow exponentially and independently, with rate proportional to $\tilde{I}(k) \tilde{C}^D(l)$, where \tilde{I} and \tilde{C}^D denote the Fourier transforms of I and C^D , respectively. The wave number k determines the wavelength $2\pi/|k|$ of an oscillation of S^D across cortical cells, while the wave number l determines the wavelength $2\pi/|l|$ of an oscillation of S^D across geniculate cells. The fastest-growing modes, which will dominate early development, are determined by the k and l that maximize $\tilde{I}(k)$ and $\tilde{C}^D(l)$, respectively. The peak of a function's Fourier transform corresponds to the cosine wave that best matches the function, and thus represents the “principal oscillation” in the function.

To understand these modes (Figure 3), consider first the set of inputs received by a single cortical cell, that is, the shape of the mode for a fixed cortical position x . This can be regarded as the “receptive field” of the cortical cell. Each receptive field oscillates with wave number l . This oscillation, of $S^D \equiv S^1 - S^2$, is an oscillation between receptive field subregions dominated by S^1 inputs and subregions dominated by S^2 inputs. Thus, in ocular dominance competition, monocular cells (cells whose entire receptive fields are dominated by a single eye) are formed only by modes with $l = 0$ (no oscillation). Monocular cells thus dominate development if the peak of the Fourier transform of the C^D governing left/right competition is at $l = 0$, which occurs if this $C^D(\alpha)$ is a nonnegative, monotonically decreasing function of $|\alpha|$ such as a Gaussian. Now instead consider an ON/OFF competition: S^1 and

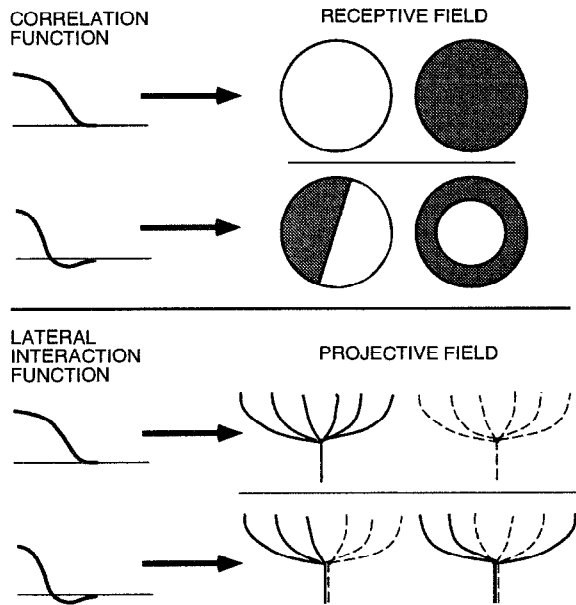


Figure 3. Schematic of the outcome of semilinear correlation-based development. *Top*, The correlation function (C^D) determines the structure of receptive fields (RFs). White RF subregions indicate positive values of S^D ; dark subregions indicate negative values. If C^D oscillates, there is a corresponding oscillation in the type of input received by individual cortical cells, as in simple cell RFs. Alternative RF structures could form, as shown, but oriented simple-cell-like outcomes predominate for reasonable parameters (Miller, 1994). When C^D does not oscillate, individual cortical cells receive only a single type of input, as in ocular dominance segregation. *Bottom*, The intracortical interactions (I) similarly determine the structure of projective fields. Here, solid lines indicate positive values of S^D , dotted lines indicate negative values.

S^2 represent ON- and OFF-center inputs from a single eye. Then the receptive fields of modes with non-zero l resemble simple cells: they receive predominantly ON-center and predominantly OFF-center inputs from successive, alternating subregions of the visual world. Thus, simple cells can form if the C^D governing ON/OFF competition has its peak at a non-zero l , which occurs if this $C^D(\alpha)$ oscillates from positive to negative with increasing $|\alpha|$.

Now consider the arborizations or “projective fields” projecting from a single geniculate point, that is, the shape of the mode for a fixed geniculate position α . These oscillate with wave number k . In ocular dominance competition, this means that left- and right-eye cells from α project to alternating patches of cortex. When monocular cells form ($l = 0$), these alternating patches of cortex are the ocular dominance columns: alternating patches of cortex receiving exclusively left-eye or exclusively right-eye input, respectively. Thus, the width of ocular dominance columns—the wavelength of alternation between right-eye-dominated and left-eye-dominated cortical cells—is determined by the peak of the Fourier transform of the intracortical interaction function I . In ON/OFF competition, with $l \neq 0$, the identity of the cortical cells receiving the ON-center or OFF-center part of the projection varies as α varies, so individual cortical cells receive both ON- and OFF-center input, but from distinct subregions of the receptive field.

In summary, there is an oscillation across receptive fields, with wave number l determined by the peak of \tilde{C}^D , and an oscillation across arbors, with wave number k determined by the peak of \tilde{I} (Figure 3). These two oscillations are “knit together” to determine the overall pattern of synaptic connectivity. The receptive field oscillation, which matches the receptive field to the correlations, is

the quantitative generalization of von der Malsburg’s finding that individual receptive fields become selective for a correlated pattern of inputs. Similarly, the arbor oscillation matches projective fields to the intracortical interactions, and thus to the patterns of cortical activity clusters. This quantitatively generalizes the relationship between activity clusters and maps. Note that the factor e^{ikx} can be regarded as inducing a phase shift, for varying x , in the structure of receptive fields. Thus, cortical cells that are nearby on the scale of the arbor oscillation have similar receptive fields, while cells $\frac{1}{2}$ wavelength apart have opposite receptive fields.

The competitive, renormalizing terms (Equations 3 and 4) do not substantially alter this picture, except that multiplicative renormalization can suppress ocular dominance development in some circumstances (Miller and MacKay, 1994). These results hold also for localized connectivity (finite arbors), and thus generally characterize the behavior of semilinear models (Miller and Stryker, 1990). The major difference in the case of localized connectivity is that, if k or l corresponds to a wavelength larger than the diameter of connectivity from or to a single cell, then it is equivalent to $k = 0$ or $l = 0$, respectively.

Understanding Ocular Dominance and Orientation Columns with Semilinear Models

This understanding of semilinear models leads to simple models for the development of ocular dominance columns (Miller and Stryker, 1990), orientation columns (Miller, 1994), and their codevelopment (Erwin and Miller, 1998), as follows.

Monocular cells develop through a competition of left- and right-eye inputs in a regime in which $\tilde{C}^D(l)$ is peaked at $l = 0$. The wavelength of ocular dominance column alternation is then determined by the peak of $\tilde{I}(k)$.

Orientation-selective simple cells develop through a competition of ON-center and OFF-center inputs in a regime in which $\tilde{C}^D(l)$ is peaked at $l \neq 0$. The mean wavelength of alternation of ON-center and OFF-center subregions in the simple cells’ receptive fields is determined by the peak of $\tilde{C}^D(l)$. This wavelength corresponds to a cell’s preferred spatial frequency under stimulation by sinusoidal luminance gratings. In individual modes, all cortical cells have the same preferred orientation, but their spatial phase varies periodically with cortical position. The mixing of such modes of all orientations, along with the saturating nonlinearities limiting the sizes of individual synaptic weights, leads to a periodic variation of preferred orientation across cortex.

For ocular dominance and orientation selectivity to codevelop, the two eyes must be sufficiently uncorrelated to allow ocular dominance segregation, but sufficiently correlated that preferred orientations match in the two eyes on binocular cells and that the map of preferred orientations is continuous across ocular dominance boundaries. The requirements for this to occur turn out to be simple generalizations of the above two requirements, plus an additional requirement that between-eye correlations be specific for center type (e.g., between-eye ON-ON and ON-OFF correlations should be distinct, in a manner that varies over a receptive field diameter).

This model of ocular dominance column formation is similar to that of von der Malsburg (von der Malsburg and Willshaw, 1976). The latter model assumed anticorrelation between the two eyes; this assumption was required because of the use of multiplicative renormalization (Equation 4). With subtractive renormalization (Equation 3), ocular dominance column formation can occur even with partial correlation of the two eyes (Miller and MacKay, 1994).

The model of orientation-selective cell development is quite different from that of von der Malsburg (1973). Von der Malsburg postulated that oriented input patterns lead to the development of orientation-selective cells. The ON/OFF model instead postulates that ON/OFF competition results in oriented receptive fields in the

absence of oriented input patterns; the circular symmetry of the input patterns is spontaneously broken. This symmetry-breaking potential of Hebbian development was first discovered by Linsker (1986). In all of these models, the continuity and periodic alternation of preferred orientation is due to the intracortical connectivity.

The models can be compared to experiment most simply by measuring activity correlations in the LGN, such as to determine whether the ON/OFF C^D has the predicted oscillation or whether the between-eye correlations have the predicted structure. Progress toward such tests is reviewed in Weliky (2000).

Related Semilinear Models

Linsker (1986) proposed a model that was highly influential in two respects. First, he pointed out the potential of Hebbian rules to spontaneously break symmetry, yielding orientation-selective cells given approximately circularly symmetric input patterns. Second, he demonstrated that Hebbian rules could lead to segregation *within* receptive fields, so that a cell would come receive purely excitatory or purely inhibitory input in alternating subregions of the receptive field. Two factors underlay the results. One factor was that oscillations in a correlation function can induce oscillations in a receptive field, as described earlier. The other factor was a constraint in the model fixing the percentage of positive or negative synapses received by a cell; this constraint forced an alternation of positive and negative subregions even when the correlation function did not oscillate.

Tanaka has independently formulated models of ocular dominance (Tanaka, 1991) and orientation columns (Miyashita and Tanaka, 1992) that are similar to those described above. The major difference is that in his regime, each cortical cell comes to receive only a single LGN input. (The reason is that he works in a regime in which, on each postsynaptic cell, (1) total synaptic weight is conserved, (2) the only stable outcome is if all or all-but-one synapses are saturated at either the lower or upper bounds on synaptic weights [this is also true for linear rules with subtractive weight normalization—see Miller and MacKay, 1994—although Tanaka's rule is slightly different], and (3) there is a lower bound on synaptic weights at zero, but no upper bound. The result is that the only stable configuration is one in which one synapse acquires all of the conserved weight, and all of the other synapses are forced to the lower bound at zero.) Tanaka defines cortical receptive fields as the convolution of the input arrangement with the intracortical interaction function. This definition means that a cortical cell's receptive field is due to its single input from the LGN, plus its input from all other cortical cells within reach of the intracortical interaction function. Thus, orientation selectivity in this model arises from the breaking of circular symmetry in the pattern of inputs to different cortical cells rather than to individual cortical cells.

The Problem of Map Structure

The models described to this point account well for basic features of primary visual cortex, including the structure of individual receptive fields and local continuity across cortex of receptive field properties. However, certain details of real orientation maps are not replicated by these models (reviewed in Erwin, Obermayer, and Schulten, 1995; Swindale, 1996). One reason may be the simplicity of the model of cortex: the real cortex has three dimensions rather than two, has cell-specific connectivity rather than connectivity that depends only on distance, and has plastic rather than fixed intracortical connections. Another reason is that the details of map structure inherently involve nonlinearities, by which the fastest-growing modes interact and compete, whereas the semilinear framework

primarily focuses on early pattern formation, in which the fastest-growing modes emerge and mix randomly without interacting.

Some simple models that focus on map development rather than on receptive field development and that use reduced, feature-based representations of the input yield maps that strikingly match the structures observed in monkey visual cortex (Erwin et al., 1995; Swindale, 1996). One such model uses the self-organizing feature map (SOFM) of Kohonen (see SELF-ORGANIZING FEATURE MAPS). In the SOFM, only a single cluster of cortical cells is activated in response to a given input pattern. This is an abstraction of the idea that the cortex responds in localized activity clusters. In addition, an abstract representation of the input is used. Correlation-based models are "high-dimensional" models: the vector of input weights received by a cell has hundreds or thousands of dimensions, one for each input cell. The SOFM model of visual cortex is instead a "low-dimensional" or "feature-based" model: the vector of input weights received by a cell has only five dimensions, representing features of the visual input (two dimensions represent retinotopic position and one each represent ocular dominance, orientation selectivity, and preferred orientation). Assumptions are made as to the relative "size" of, or variance of the input ensemble along, each dimension. There is no obvious biological interpretation for this comparison between dimensions. Under certain such assumptions, Hebbian learning in the feature space leads to maps of orientation and ocular dominance that are, in detail, remarkably like those seen in macaque monkeys (Erwin et al., 1995; Swindale, 1996). (SOFMs with high-dimensional input representations have also been studied, but it is not clear whether they are any better at capturing map structure than other high-dimensional models.)

The SOFM and other feature-based models, such as those based on the "elastic net" algorithm (Erwin et al., 1995; Swindale, 1996), lead to locally continuous mappings in which a constant distance across the cortex corresponds to a roughly constant distance in the reduced "input space." This means that, when one input feature is changing rapidly across cortex, the others are changing slowly. Thus, the models predict that orientation changes rapidly where ocular dominance changes slowly, and vice versa. This feature may be key to replicating the details of macaque maps. However, recent results suggest that this relationship may not hold between retinotopy and orientation in cat visual cortex (Das and Gilbert, 1997).

A possibly related approach to map organization supposes that the structure of maps is determined by evolutionary pressure to minimize the wire length of neuronal connections (Chklovskii and Koulakov, 2000; Koulakov and Chklovskii, 2001). Assuming a certain "connectivity function," specifying probabilistically which cells should connect to which, a feature map that allows such connectivity to be achieved with minimal wire length is computed. For certain forms of the connectivity function, orientation maps are obtained with pinwheels—point singularities around which preferred orientation rotates by 180 degrees—much as in real maps (Koulakov and Chklovskii, 2001). The model predicts that the ocular dominance map should go from a stripe-like structure to a patch-like structure when one eye contributes less than 40% of the total innervation, and just such a transition at about the appropriate point is seen in monkey visual cortex (Chklovskii and Koulakov, 2000).

Another approach to studying map development is based on very general assumptions as to symmetries of the map-formation process. It was shown that any of a large class of developmental models of orientation selectivity that obey a few basic symmetries and that produce a periodic map of preferred orientation with period λ should produce an initial density of pinwheels of at least π per λ^2 (Wolf and Geisel, 1998). Comparison with actual maps showed a range of densities from 2 to 4, suggesting that if real maps develop by a self-organizing process, those with densities less than π must undergo reorganization involving pinwheel annihilation during

early development. It will be exciting if such pinwheel annihilation should be observed, although it is possible that such annihilation might occur only before maps become visible with current methods.

Open Questions

Among the many open questions in the field are these: How can biologically interpretable developmental models replicate the details of cortical maps? How might more realistic models of Hebbian learning and of competition (e.g., Song et al., 2000) modify the basic insights into development obtained by studying simple semi-linear models? How will these insights, which apply to development of feedforward connections, be modified by considering codevelopment of feedforward and intracortical synaptic innervations? (See Song and Abbott, 2001, and Kayser and Miller, 2002.) How will they be modified by developing models that learn on temporal as well as spatial correlations? The simple models have yielded real insights, but the field now seems prepared to enter more complex terrain.

Road Maps: Neural Plasticity; Vision

Related Reading: Development of Retinotectal Maps; Hebbian Synaptic Plasticity; Pattern Formation, Biological; Visual Cortex: Anatomical Structure and Models of Function

References

- Chklovskii, D. B., and Koulakov, A. A., 2000, A wire length minimization approach to ocular dominance patterns in mammalian visual cortex, *Physica A*, 284:318–334.
- Crair, M., Horton, J., Antonini, A., and Stryker, M., 2001, Emergence of ocular dominance columns in cat visual cortex by 2 weeks of age, *J. Comp. Neurol.*, 430:235–249.
- Das, A., and Gilbert, C. D., 1997, Distortions of visuotopic map match orientation singularities in primary visual cortex, *Nature*, 387:594–598.
- Erwin, E., and Miller, K. D., 1998, Correlation-based development of ocularly-matched orientation maps and ocular dominance maps: Determination of required input activity structures, *J. Neurosci.*, 18:9870–9895.
- Erwin, E., Obermayer, K., and Schulten, K., 1995, Models of orientation and ocular dominance columns in the visual cortex: A critical comparison, *Neural Computat.*, 7:425–468.
- Katz, L. C., and Crowley, J. C., 2002, Development of cortical circuits: Lessons from ocular dominance columns, *Nature Rev. Neurosci.*, 3:34–42. ♦
- Kayser, A. S., and Miller, K. D., 2002, Opponent inhibition: A developmental model of layer 4 of the neocortical circuit, *Neuron*, 33:131–142.
- Koulakov, A., and Chklovskii, D., 2001, Orientation preference patterns in mammalian visual cortex: A wire length minimization approach, *Neuron*, 29:519–527.
- Linsker, R., 1986, From basic network principles to neural architecture (series), *Proc. Natl. Acad. Sci. USA*, 83:7508–7512, 8390–8394, 8779–8783.
- Miller, K. D., 1994, A model for the development of simple cell receptive fields and the ordered arrangement of orientation columns through activity-dependent competition between ON- and OFF-center inputs, *J. Neurosci.*, 14:409–441.
- Miller, K. D., Erwin, E., and Kayser, A., 1999, Is the development of orientation selectivity instructed by activity? *J. Neurobiol.*, 41:44–57. ♦
- Miller, K. D., and MacKay, D. J. C., 1994, The role of constraints in Hebbian learning, *Neural Comput.*, 6:100–126.
- Miller, K. D., and Stryker, M. P., 1990, The development of ocular dominance columns: Mechanisms and models, in *Connectionist Modeling and Brain Function: The Developing Interface* (S. J. Hanson and C. R. Olson, Eds.), Cambridge, MA: MIT Press/Bradford, pp. 255–350.
- Miyashita, M., and Tanaka, S., 1992, A mathematical model for the self-organization of orientation columns in visual cortex, *NeuroReport*, 3:69–72.
- Song, S., and Abbott, L., 2001, Cortical development and remapping through spike timing-dependent plasticity, *Neuron*, 32:339–350. ♦
- Song, S., Miller, K. D., and Abbott, L. F., 2000, Competitive Hebbian learning through spike-timing-dependent synaptic plasticity, *Nature Neurosci.*, 3:919–926.
- Swindale, N. V., 1996, The development of topography in the visual cortex: A review of models, *Network*, 7:161–247. ♦
- Tanaka, S., 1991, Theory of ocular dominance column formation: Mathematical basis and computer simulation, *Biol. Cybern.*, 64:263–272.
- von der Malsburg, C., 1973, Self-organization of orientation selective cells in the striate cortex, *Kybernetik*, 14:85–100.
- von der Malsburg, C., and Willshaw, D. J., 1976, A mechanism for producing continuous neural mappings: Ocularity dominance stripes and ordered retino-tectal projections, *Exp. Brain Res.*, Suppl. 1:463–469.
- Weliky, M., 2000, Correlated neuronal activity and visual cortical development, *Neuron*, 27:427–430.
- Wolf, F., and Geisel, T., 1998, Spontaneous pinwheel annihilation during visual development, *Nature*, 395:73–78.

Olfactory Bulb

Andrew P. Davison and Gordon M. Shepherd

Introduction

The olfactory bulb is the second stage in the olfactory pathway. It receives input from the sensory neurons in the olfactory epithelium and sends its outputs to the olfactory cortex, among other brain regions. The bulb is of special interest to neural modelers. It was one of the first regions of the brain for which compartmental models of neurons were constructed, which led to some of the first computational models of functional microcircuits. The aim of this article is to give an overview of (1) olfactory bulb cells and circuits, (2) current ideas about the computational functions of the bulb, and (3) modeling studies to investigate these functions. Together with the article on the OLFACTORY CORTEX (q.v.), this material provides an introduction to the nature of information processing in the olfactory system.

Cells and Circuits

Olfactory Sensory Input

The first stage in the vertebrate olfactory pathway is the detection of odor molecules by olfactory sensory neurons (OSNs). Odor molecules bind to olfactory receptor proteins (ORs), leading to the generation of receptor potentials, which are converted into action potentials. There are genes for several hundred to more than a thousand ORs in mammalian genomes. Each OSN expresses only one of these proteins. The axons of OSNs project to the olfactory bulb, where they form synapses with the dendrites of olfactory bulb neurons in spherical regions of neuropil called glomeruli. There are 1,000–2,000 glomeruli in a rodent olfactory bulb. The subset of OSNs expressing the same OR type all project their axons to the

same glomeruli, generally one on the medial side and one on the lateral side of the bulb. Furthermore, since it appears that each glomerulus receives axons from only a single type of receptor neuron, there is a one-to-one mapping between a glomerulus and an OR type.

An odor molecule typically binds to many ORs, with varying affinities, and therefore activates many glomeruli to different degrees. Since the olfactory bulb is a laminar structure, there is a mapping from 1,000-dimensional odor space (one dimension per OR) to two-dimensional neural space.

The Basic Circuit of the Olfactory Bulb

There are two distinct levels of synaptic interactions in the olfactory bulb, the glomerular layer and the external plexiform layer (EPL) (Figure 1). These layers can be regarded as levels of input processing and output control, respectively.

Within the glomeruli, OSN axons make excitatory synapses onto the distal dendritic tuft of mitral, tufted, and periglomerular (PG) cells. Mitral and tufted (M/T) cells make excitatory dendrodendritic synapses onto PG cells. PG cells make inhibitory dendrodendritic synapses onto M/T cells and probably onto other PG cells. PG cells also send axons to neighboring glomeruli, where they form inhibitory synapses onto M/T cell dendrites.

The secondary dendrites of M/T cells extend long distances laterally in the EPL, up to half of the circumference of the bulb for mitral cells. There is evidence that action potentials can propagate from the soma to the very tips of these dendrites. M/T secondary dendrites make reciprocal, dendrodendritic synapses with granule cells, which are anaxonal interneurons. These reciprocal synapses consist of an excitatory synapse from M/T cells onto granule cell spines paired with an inhibitory synapse from the spine onto the M/T cell dendrite.

The effects of the dendrodendritic synapses are strongly affected by the intrinsic properties of the cells, particularly the granule cells.

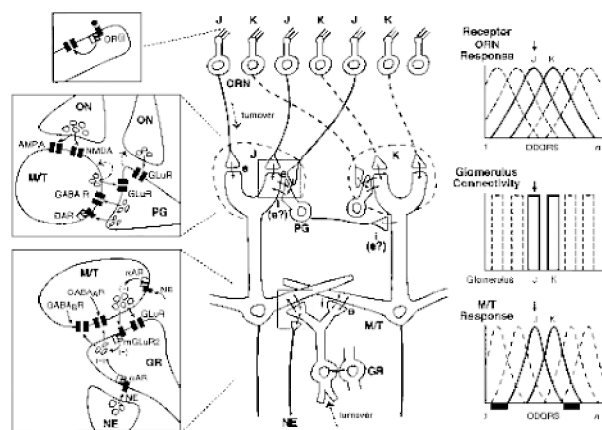


Figure 1. Basic circuit for the olfactory bulb. Abbreviations (left, molecular components): OR, olfactory receptor; ON, olfactory nerve; AMPA, 2-amino-5-phosphonvaleric acid; NMDA, *N*-methyl-D-aspartate; M/T, mitral/tufted cell; PG, periglomerular cell; GluR, ionotropic glutamate receptor; GABA R, γ -aminobutyric acid receptor; DAR, dopamine receptor; NE, norepinephrine; α AR, α -adrenoreceptor; mGluR2, metabotropic glutamate receptor; GR, granule cell. Middle (synaptic circuit): ORN, olfactory receptor neuron; J, K, ORN subsets; e, excitatory; i, inhibitory. Right (structure-function relations) top: overlapping response spectra of ORNs to a range of odors (1 – *n*); middle: connectivity of subsets to individual glomeruli; bottom: response spectra of M/T cells show less overlap because of lateral inhibition (black bars below abscissa).

First, the synapses are onto spines, so a small current can produce a large local membrane depolarization. Second, the granule cell dendrites contain an A-type potassium current, which prevents short-duration synaptic inputs from producing an action potential. Granule cell action potentials may not be required for recurrent inhibition of the M/T cells; glutamate released by the M/T cell depolarizes the granule cell spine sufficiently to produce GABA release, which then inhibits the M/T cell. In mitral cells, subthreshold membrane potential oscillations may interact with postsynaptic potentials to gate the times at which action potentials occur.

M/T cells exhibit autoexcitation due to diffusion of synaptically released glutamate to extrasynaptic ionotropic and metabotropic receptors on both primary and secondary dendrites. This can lead to very prolonged responses to excitatory input. M/T cells do not make chemical synapses with other M/T cells. However, several studies have recently demonstrated functional excitatory connections between mitral cells, probably due again to glutamate spill-over to extrasynaptic receptors. M/T cells give off collaterals in the granule cell layer.

Mitral cells send axons to several basolateral brain regions, including the piriform cortex (primary OLFACTORY CORTEX) and the amygdala. The projections of tufted cells are more limited. Centrifugal inputs from several brain areas include cholinergic, serotonergic, and noradrenergic pathways. These inputs mostly terminate on granule and PG cells to influence neuronal excitability and neurotransmitter release. They are critical in setting the behavioral state of the whole system.

Compartmental Models of Olfactory Bulb Neurons

Wilfrid Rall introduced the compartmental method for computational modeling of neurons in 1964 (see PERSPECTIVE ON NEURON MODEL COMPLEXITY). Motoneurons in the spinal cord and mitral and granule cells in the olfactory bulb (Rall and Shepherd, 1968) were the first neurons to be modeled using this approach. Both active and passive dendritic properties were explored in these models. These studies have been extended by compartmental models in which the distribution of active properties in the dendritic trees has been assessed by systematic parameter searches (Bhalla and Bower, 1993). A limitation of earlier studies is that the physiological data were obtained from single-point recordings in the soma, so distal parts of the neurons (e.g., the mitral cell primary dendrite tuft) are much less well constrained than the soma and proximal parts.

Recently, more tightly constrained models have been made possible by experimental data obtained from single- and dual-patch recordings from the soma and from distal sites on the primary dendrite. These studies have shown that full action potentials are generated throughout the primary dendrite, supported by an even distribution of fast Na and K conductances. The M/T cell primary dendrite supports the backpropagation of action potentials from the soma to the tuft. Under conditions of moderate to strong ON input and/or somatic inhibition, action potentials can be initiated in the tuft and propagate in the forward direction to the soma.

These properties have been closely simulated by a tightly constrained compartmental model of the mitral cell (Shen et al., 1999). The results provide one of the best models, in a vertebrate central neuron, of the shifting sites of action potential initiation dependent on the integration of excitatory and inhibitory synaptic inputs with active membrane properties. They support a wide range of studies giving evidence of the importance of active properties of distal dendrites in neuronal function. This gives added emphasis to the importance of the contributions of distal dendrites to the basic circuits for different brain regions, which need to be incorporated into neural networks in order to fully capture the mechanisms used by the real nervous system.

Compartmental models of granule cells have shown that high-input-resistance spines and low levels of activity favor reciprocal over lateral inhibition; conversely, lower-input-resistance spines and high levels of activity favor lateral inhibition (Antón, Granger, and Lynch, 1993), and that with entirely passive membrane in a morphologically detailed model, “the degree of spread of synaptic potentials can define functionally related subsets of spines within the dendritic tree . . . that can mediate discrete localized inhibition onto subsets of mitral or tufted cell secondary dendrites” (Woolf, Shepherd, and Greer, 1991).

The Computational Functions of the Olfactory Bulb

As the first stage of synaptic processing in the olfactory pathway, the olfactory bulb carries out several key functions.

Glomeruli Contribute to Odor Detection

A mechanism that has long been recognized for supporting odor detection is the tremendous convergence of OSNs onto glomeruli. In mammals, the convergence ratio is of the order 10,000:1. As noted earlier, it is believed from axon-labeling studies that all of the OSNs expressing a given OR converge onto one or two common target glomeruli. Thus, at the lowest concentrations, when only the highest-affinity receptor type is activated, all odor responses are concentrated on a single pair of glomeruli.

Spatial Maps Mediate Odor Recognition

A given odor activates a particular set of OR types. The set of activated ORs is mapped onto a two-dimensional pattern of activated glomeruli. These patterns have been visualized by a large number of imaging techniques (see Xu, Greer, and Shepherd, 2000, for a review). The responses to different odor stimuli have also been characterized. These studies have shown that odor molecules that are structurally similar activate glomeruli that are near one another, and the patterns of activation for different odorants generally overlap considerably. There is thus increasing support for the hypothesis that odor recognition is based on patterns characteristic for given odors.

Odor Maps Encode Odor Concentration

Odor mapping methods show that increasing odor concentration recruits increasing numbers of activated glomeruli, as the higher concentrations cause binding of additional ORs with lower affinities in OSNs projecting to other glomeruli (see Xu et al., 2000). Thus, odor concentration appears to be encoded at least in part by the number of activated glomeruli, and probably also by the intensity of their activation. Modeling studies have suggested that odor intensity could additionally be encoded by the number of M/T cells within a glomerulus that are firing (Antón, Lynch, and Granger, 1991). Experimentally, there is a gradient of decreasing excitability from tufted to mitral cells, so that increasing odor intensity will recruit additional neurons to fire. An argument against this hypothesis is the observation that M/T cells have different projection patterns and so are not a single population for the purpose of transmitting information to cortex.

Temporal Structure Encodes Odor Concentration

Time may also be used as a dimension for encoding stimulus intensity. First, the temporal firing pattern of M/T cells changes as odor concentration is increased. Second, time delays may be used to encode odor intensity. White et al. (1998) developed a network model of the olfactory bulb that receives input from an array of fiber-optic chemodetectors constituting an artificial nose. The network functions as a delay line neural network in which odor iden-

tity is encoded by spatial pattern and odor intensity by response delay. This network outperforms standard feedforward neural networks in discriminating odors when limited training sets are used. This work represents an important practical application of neural networks in the olfactory system for artificial chemosensing systems, with applications in industry and for detecting narcotics and hidden explosives.

Lateral Inhibition Mediates Odor Discrimination

A critical function of olfactory bulb circuits is to contribute to the neural basis for discrimination between different odors. As we have seen, the responses to different odors are carried in some 2,000 parallel glomerular pathways. Discrimination between odors requires circuits that can compare the patterns of responses across the pathways and extract the differences.

A key operation for mediating odor discrimination is reducing the overlap between the representations of different odors. Strongly activated mitral cells suppress the activity of more weakly activated mitral cells by lateral inhibition, mediated through the dendrodendritic synapses with granule and periglomerular cells. This sharpens the tuning curves of mitral cells connected to a given glomerulus. This is equivalent to reducing the overlap between the affinity spectra (the molecular receptive ranges) of the individual glomeruli. This was demonstrated experimentally by Yokoi, Mori, and Nakanishi (1995), who found that affinity spectra for a series of aldehydes were broadened by blocking lateral inhibition, presumably mediated through granule cells.

The distinction between the roles of granule and PG cells in mediating lateral inhibition has been investigated in an olfactory bulb model developed by Linster and Hasselmo (1997): PG cell activity affects the number of active mitral cells, while granule cell activity determines the response intensity of active mitral cells. Both granule and PG cell activities may be controlled by centrifugal inputs. The result of the two levels of lateral inhibition is to make the odor representation more sparse.

A complementary mechanism for reducing overlap is to add temporal structure to the responses (i.e., two odors may activate the same ensembles of cells, but at different times) by way of the lateral interactions. Adding time as an extra dimension gives more space in which to separate out odors. Experimentally, the response patterns of mitral cells have complex temporal structures. Laurent et al. (2001) have introduced a dynamical systems model of the olfactory bulb/antennal lobe in which the state of the system, represented by the instantaneous firing rates of the projection neurons, follows a heteroclinic orbit in phase space. Orbits starting from nearby starting points (representing similar odorants) diverge rapidly, thus allowing easier discrimination between them. Experimentally, they provide evidence in zebrafish that an initially “clustered” representation becomes declustered with time.

Temporal Correlations May Contribute to Odor Discrimination

It has been suggested that one function of the bulb is to introduce temporal correlations, such as spike synchronization, between the signals from different receptors responding to the same odor (Laurent et al., 2001). If the olfactory cortex has cells that are tuned as “coincidence detectors,” this could increase the salience of mitral cells that are synchronized relative to unsynchronized cells. Gamma-frequency, odor-induced oscillations in local field potential (LFP) recordings have been reported in the olfactory bulbs of vertebrates and in the antennal lobes of insects. These oscillations are thought to reflect the rhythmical, synchronous firing of populations of neurons. In insects and rabbits, the spiking of individual neurons is phase-locked to the LFP, and spiking in simultaneously

recorded pairs of neurons is often closely synchronized. Oscillations with different (sometimes harmonically related) frequencies may be elicited in different regions of the olfactory bulb. In insects, different subpopulations of neurons may be synchronized with the global oscillation at different periods during the response (see Laurent et al., 2001). In honeybees, oscillatory synchronization has been shown to have functional relevance: pharmacologically induced desynchronization impairs the ability to discriminate between similar odorants but has no effect on discrimination of dissimilar odorants (see Laurent et al., 2001).

Synaptic Modification Produces a Transformation of Odor Space

Not all synapses are equal in strength. Therefore, the firing of some cells will have a larger influence on bulb output than others. The strength of synapses may be changed by mechanisms of synaptic plasticity during learning experiences. For example, recordings of mitral cell activity in the olfactory bulb of sheep before and after giving birth showed that before parturition, the majority of cells responded preferentially to food odors, and after parturition, the majority responded preferentially to lamb odors (Kendrick, Lévy, and Keverne, 1992). Accompanying this change in preference were increases in neurotransmitter release in the bulb. It appears, therefore, that a function of the bulb is to produce a mapping between the input, physical/chemical odor space and the output, neural odor space, such that behaviorally important odors cover a larger region of output space, enhancing the ability of the system to discriminate between similar odors. Odors with less behavioral significance would be mapped to a reduced region of output space. Changes in synaptic strengths can alter this mapping. In the case of the sheep, synaptic changes in the bulb increased the representation of lamb odors in output space at the expense of the extent of output space representing food odors.

Realistic Network Models

Several models have been developed that attempt to reproduce the anatomy and physiology of real cells and synapses and the connectivity of the real olfactory bulb network without any a priori idea of how the network should behave.

White et al. (1992) have investigated how responses of olfactory bulb neurons to electrical or odor stimulation may be shaped by microcircuit interactions in the salamander olfactory bulb. The total number of cells in the model is 1,000-fold fewer than found in the real system, but the ratios between the populations are maintained. Mitral cells are modeled with three compartments (soma and two dendritic tufts) using standard cable equations. Granule and PG cells are represented by single compartments. Presynaptic spikes and/or subthreshold increases in membrane potential trigger postsynaptic conductance changes. In simulations, each of the major types of response seen in electrophysiological recordings of salamander cells is reproduced by varying the spatial pattern of activity applied to the receptor cells (Figure 2).

Davison, Feng, and Brown (2003) simulated a network of 100 mitral cells and 15,000 granule cells using simplified versions of the models of Bhalla and Bower (1993) and realistic synaptic conductances. This model network displayed properties of center-surround inhibition that are consistent with the experimental findings of Yokoi et al. (1995) (Figures 3A–D). The model also displayed synchronization between mitral cells in response to odor inputs, as seen in the rabbit olfactory bulb and similar to the synchronization seen in the insect antennal lobe (Figure 3E). Weakly activated mitral cells fire less frequently than, but always synchronously with, strongly activated cells. Nearby cells synchronize more readily than widely separated ones. These findings also dem-

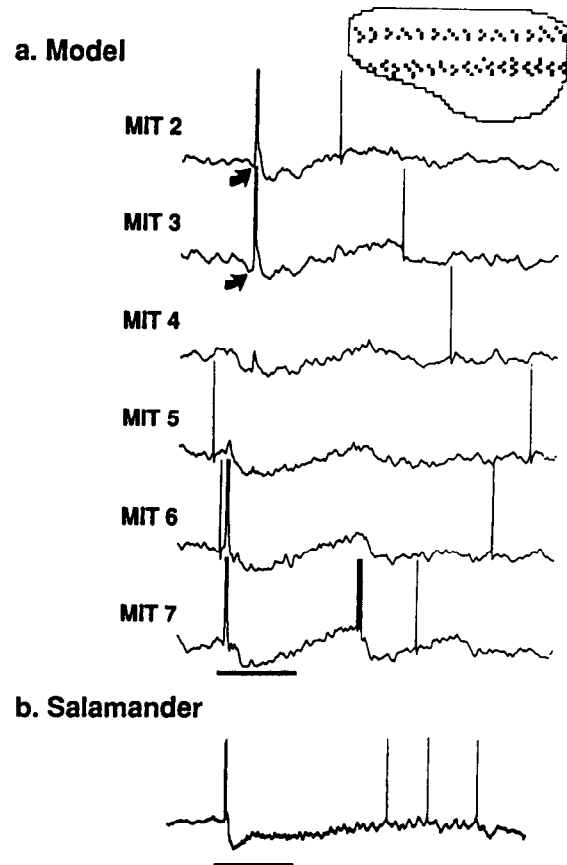


Figure 2. Odor stimulation of modeled mitral (MIT) cell responses (MIT 2–7) (A) compared with experimental recordings (B). Note the pattern of excited receptor cells in the receptor sheet in the inset at top. Arrows in A indicate a brief hyperpolarization that is also seen in experimental recordings. (From White, J., Hamilton, K. A., Neff, S. R., and Kauer, J. S., 1990, Emergent properties of odor information coding in a representational model of the salamander olfactory bulb, *J. Neurosci.*, 12:1772–1780. Reprinted with permission.)

onstrate the central role of the dendrodendritic synapses in generating both the spatial and the temporal properties of the network behavior.

Bazhenov et al. (2001) have developed a realistic model of the locust antennal lobe based on single-compartment cell models with ionic currents chosen to generate realistic firing profiles. The model reproduces the phenomena of transient oscillatory synchronization and slow temporal patterning seen in experimental recordings. The fast excitatory/inhibitory connections between local neurons (LNs) and projection neurons (PNs) were found to be necessary for network oscillations, while inhibitory connections between LNs were needed to make the synchronization transient. Adding slow inhibitory synapses from LNs to PNs was sufficient to produce the slow temporal patterns.

Discussion

The olfactory bulb has a simple structure that has made it attractive for analysis of microcircuit organization and models of neural circuits. For brain theorists, its organization offers examples of information processing without impulses and of output functions of dendrites, which has forced new concepts of the neuron as a complex computational unit. It provides unique opportunities for correlating

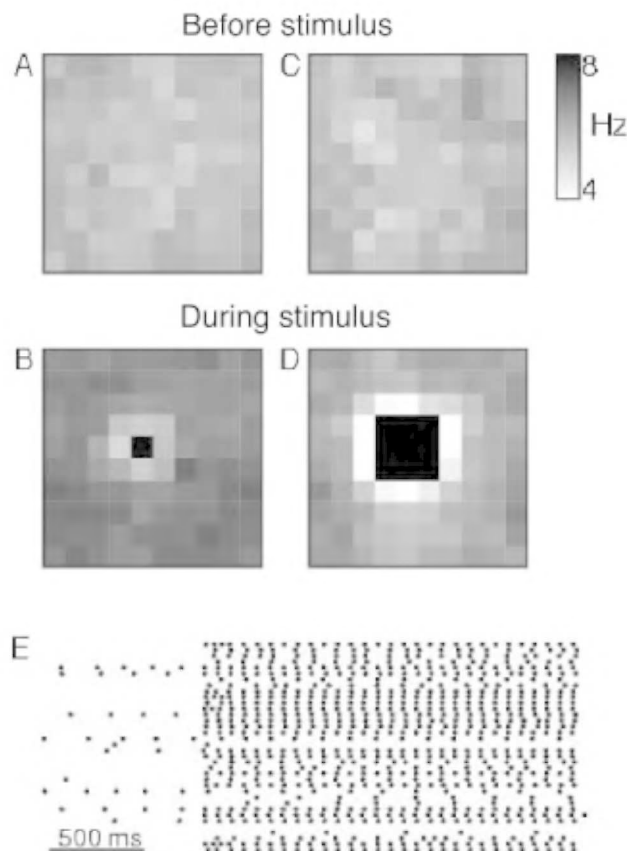


Figure 3. Lateral inhibition and stimulus-induced synchronization in an olfactory bulb model. (A–D), 10×10 array of mitral cells with uniform background activity. The gray-scale value of each square represents the mean firing rate, averaged over 2 s of simulation. Firing rates above 8 Hz are represented by black. Firing rates below 4 Hz are represented by white. A and C, The background firing rate, before stimulus onset. B, Stimulation of a single mitral cell. The firing rates of neighboring cells are depressed by about 1.5 Hz relative to the bulk of the mitral cell population. D, Stimulation of a 3×3 block of mitral cells. The firing rates of cells neighboring this block are depressed by about 3 Hz relative to the bulk of the population. E, Raster plot for a 6×6 array of mitral cells. Each dot represents an action potential. Each line is a different mitral cell. The cells are unsynchronized prior to stimulus onset. The stimulus increases the degree of synchronization in the network. (From Davison, A. P., Feng, J., and Brown, D., 2003, Dendrodendritic inhibition and simulated odor responses in a detailed olfactory bulb network model, *J. Neurophysiol.* (submitted).

membrane and cellular properties with network functions, thus pointing the way toward a deeper understanding of the neural basis of network functions. The main challenge for the future is to un-

derstand how the spatial and temporal aspects of the olfactory responses interact in olfactory information processing.

Road Maps: Mammalian Brain Regions; Other Sensory Systems

Related Reading: Dendritic Processing; Olfactory Cortex

References

- Antón, P. S., Granger, R., and Lynch, G., 1993, Simulated dendritic spines influence reciprocal synaptic strengths and lateral inhibition in the olfactory bulb, *Brain Res.*, 628:157–165.
- Antón, P. S., Lynch, G., and Granger, R., 1991, Computation of frequency-to-spatial transform by olfactory bulb glomeruli, *Biol. Cybern.*, 65:407–414.
- Bazhenov, M., Stopfer, M., Rabinovich, M., Abarbanel, H., Sejnowski, T., and Laurent, G., 2001, Model of cellular and network mechanisms for odor-evoked temporal patterning in the locust antennal lobe, *Neuron*, 30:569–581.
- Bhalla, U. S., and Bower, J. M., 1993, Exploring parameter space in detailed single cell models: Simulations of the mitral and granule cells of the olfactory bulb, *J. Neurophysiol.*, 69:1948–1965.
- Davison, A. P., Feng, J., and Brown, D., 2001, Spike synchronization in a biophysically-detailed model of the olfactory bulb, *Neurocomputing*, 38–40:515–521.
- Davison, A. P., Feng, J., and Brown, D., 2003, Dendrodendritic inhibition and simulated odor responses in a detailed olfactory bulb network model, *J. Neurophysiol.* (submitted).
- Kendrick, K. M., Lévy, F., and Keverne, E. B., 1992, Changes in the sensory processing of olfactory signals induced by birth in sheep, *Science*, 256:833–836.
- Laurent, G., Stopfer, M., Friedrich, R. W., Rabinovich, M. I., Volkovskii, A., and Abarbanel, H. D. I., 2001, Odor encoding as an active, dynamical process: Experiments, computation, and theory, *Annu. Rev. Neurosci.*, 24:263–297. ♦
- Linster, C., and Hasselmo, M., 1997, Modulation of inhibition in a model of olfactory bulb reduces overlap in the neural representation of olfactory stimuli, *Behav. Brain Res.*, 84:117–127.
- Rall, W., and Shepherd, G. M., 1968, Theoretical reconstruction of field potentials and dendrodendritic synaptic interactions in olfactory bulb, *J. Neurophysiol.*, 31:884–915.
- Shen, G. Y., Chen, W. R., Midtgard, J., Shepherd, G. M., and Hines, M. L., 1999, Computational analysis of action potential initiation in mitral cell soma and dendrites based on dual patch recordings, *J. Neurophysiol.*, 82:3006–3020.
- White, J., Dickinson, T. A., Walt, D. R., and Kauer, J. S., 1998, An olfactory neuronal network for vapor recognition in an artificial nose, *Biol. Cybern.*, 78:245–251.
- White, J., Hamilton, K. A., Neff, S. R., and Kauer, J. S., 1992, Emergent properties of odor information coding in a representational model of the salamander olfactory bulb, *J. Neurosci.*, 12:1772–1780.
- Woolf, T. B., Shepherd, G. M., and Greer, C. A., 1991, Local information processing in dendritic trees: Subsets of spines in granule cells of the mammalian olfactory bulb, *J. Neurosci.*, 11:1837–1854.
- Xu, F., Greer, C. A., and Shepherd, G. M., 2000, Odor maps in the olfactory bulb, *J. Comp. Neurol.*, 422:489–495. ♦
- Yokoi, M., Mori, K., and Nakanishi, S., 1995, Refinement of odor molecule tuning by dendrodendritic synaptic inhibition in the olfactory bulb, *Proc. Natl. Acad. Sci. USA*, 92:3371–3375.

Olfactory Cortex

Christiane Linster, Michael E. Hasselmo, Matthew A. Wilson, and Gordon M. Shepherd

Introduction

The olfactory cortex has traditionally played an important role in theoretical studies of cortical function. It is the earliest cortical

region to differentiate in the evolution of the vertebrate forebrain. It is the only region within the forebrain to receive direct sensory input. The olfactory input processed by the cortex dominates the behavior of most vertebrate species. Thus, the role of the olfactory

cortex is critical for the evolution of much of vertebrate behavior. Finally, the olfactory cortex has the simplest organization among the main types of cerebral cortex. These features have suggested that the olfactory cortex may serve as a model for understanding basic principles underlying cortical organization.

The olfactory pathway begins with the olfactory receptor neurons in the nose, which project their axons to the olfactory bulb. The function of the OLFACTORY BULB (q.v.) is to perform the initial stages of sensory processing of the olfactory signals before sending this information to the olfactory cortex. The olfactory cortex is defined as the region of the cerebral cortex that receives direct connections from the olfactory bulb (Figure 1). It is subdivided into several areas that share a basic organization but are distinct in terms of details of cell types, lamination, and sites of output to the rest of the brain. The main area involved in olfactory perception is the piriform (also called prepiriform) cortex (Figure 1), which projects to the mediodorsal thalamus, which in turn projects to the frontal neocortex. This is often regarded as the main olfactory cortex, and is the subject of this article.

Evolutionary Significance of the Olfactory Cortex

For brain theorists interested in principles of cortical organization, the early appearance of the olfactory cortex in phylogeny deserves attention. The cerebral cortex first appears in vertebrate evolution in fishes as a simple structure composed of three layers: a superficial layer containing incoming nerve fibers, dendrites of intrinsic and output neurons, and scattered cell bodies of interneurons; a layer of the large cell bodies of output neurons; and a deep layer of interspersed input and output fibers, and scattered cell bodies of interneurons. This is the classical three-layered cortex. The cortex on the ventrolateral surface that receives direct olfactory input from the olfactory bulb is termed *paleocortex*, which is the olfactory cortex as described above. On the medial surface is another part related to the septum, termed *archicortex*; this is the anlage of the hippocampus in higher vertebrates. On the dorsal surface is the so-called dorsal cortex, generally believed to be the anlage of neocortex.

During phylogeny, the paleocortex and archicortex develop in extent and complexity but retain their three-layered character. Neocortex, however, emerges in mammals as a five- to six-layered structure. It is controversial among evolutionary neurobiologists whether the dorsal cortex can in fact be considered an early representation of neocortex or whether it is more properly considered an anlage, i.e., a predecessor of true neocortex. In reptiles, such as turtles, this dorsal cortex has become sufficiently differentiated to serve as the visual cortex for visual input relayed from the thalamus.

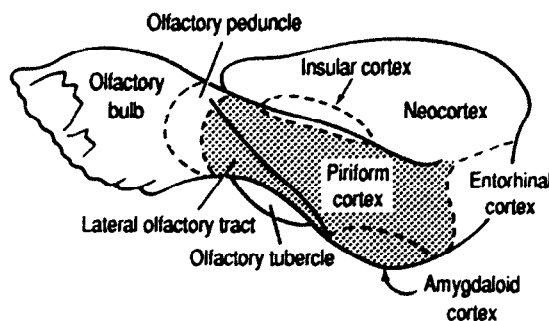


Figure 1. The relation of the olfactory cortex to the main components of the olfactory pathway. (From Haberly, L. B., 1990, Olfactory cortex, in *The Synaptic Organization of the Brain* (G. M. Shepherd, Ed.), New York: Oxford University Press. Reprinted with permission.)

Whether this can be regarded as a true “primary” visual cortex, homologous to primary visual cortex in the mammal, or whether it is only a primitive anlage of primary visual cortex is a matter for debate.

With the rise of modern studies of synaptic organization, it was hypothesized that comparisons between brain cortical regions in phylogeny should focus less on the numbers of layers and more on the particular types of circuits that are present and the functions that they mediate (Shepherd, 1998). We will therefore identify the main types of circuits that are present in the olfactory cortex. We will then describe compartmental and network models, and discuss the insights gained from these models into olfactory processing and their relevance for understanding general properties of cortical networks.

Basic Circuit for Olfactory Cortex

The concept of a basic or “canonical” circuit is of critical importance for computational neuroscience and brain theory. The basic circuit combines the results of anatomical, physiological, neurochemical, and computational studies into a consensus representation of the main circuits in a particular region (Shepherd, 1998). This objective is facilitated by the extent to which the region in question has distinct layers, clearly differentiated cell types, and readily characterizable inputs. Of all cortical regions, the olfactory cortex best satisfies these criteria.

Our current understanding of the olfactory cortex basic circuit has arisen from a series of anatomical, physiological, and pharmacological studies. The current consensus model is summarized in Figure 2. The main features of the basic circuit include the following.

The primary sensory input (through the lateral olfactory tract from the olfactory bulb) makes its synapses on the most distal parts of the apical dendrites of the pyramidal neurons. This continues the pattern present in the earlier stages of the olfactory pathway, in which primary sensory input is delivered to the most distal parts of the dendrites of the sensory neurons, and their axons in turn make synapses on the most distal dendrites of their targets, the mitral/tufted cells of the olfactory bulb. Thus, distal dendrites, rather than being sites for weak background modulation of neuronal activity, are the preferred sites for rapid transmission of specific sensory transmission from neuron to neuron in this pathway. This fact is critically important for brain theorists and neural modelers, because it means that the specific properties of distal dendrites must be included in network models in order to represent the mechanisms of processing. The properties of dendrites are discussed in PERSPECTIVE ON NEURON MODEL COMPLEXITY.

The distal inputs in olfactory cortex are made exclusively onto dendritic spines of the apical dendrites. These are very small branches, only a micron or so in length and 0.1–0.2 μm in diameter, terminating in a head (1–2 μm across) that receives an excitatory synapse. Spines are of considerable current interest as sites for activity-dependent mechanisms, such as long-term potentiation (LTP), that may underlie learning (reviewed in Shepherd, 1998). Both the afferent excitatory inputs and the recurrent excitatory inputs are made to spines, and both show properties of LTP.

The intrinsic cortical circuits for processing information consist of inhibitory and excitatory local circuits. The inhibitory circuits are of two types: those for feedforward inhibition and those for feedback (lateral) inhibition, as indicated in Figure 2. A given interneuron may be involved exclusively in one of these types, or it may be a node for convergence and integration of both types. The excitatory circuits provide not only for the excitation of the inhibitory interneurons in the feedback (lateral) pathway, but also for direct recurrent excitation of other pyramidal neurons. These intrinsic excitatory and inhibitory inputs are made to different regions

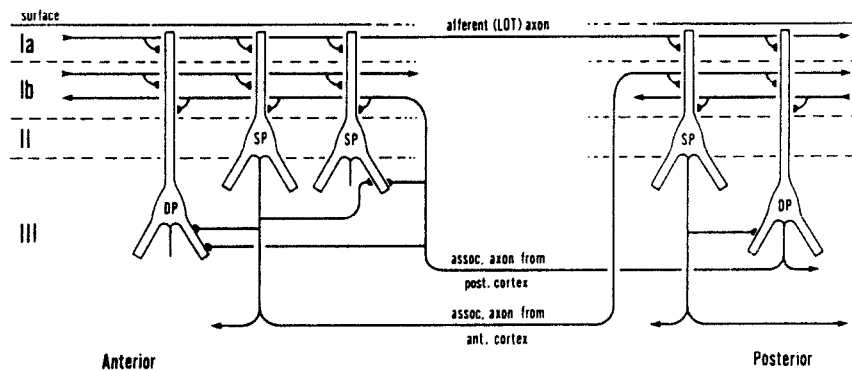
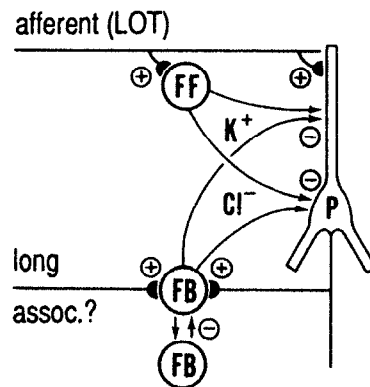


Figure 2. The basic circuit of the olfactory cortex. FF, feedforward; FB, feedback; P, pyramidal neurons, superficial (S) and deep (D). (From Haberly, L. B., 1990, Olfactory cortex, in *The Synaptic Organization of the Brain* (G. M. Shepherd, Ed.), New York: Oxford University Press. Reprinted with permission.)



and levels of the apical and basal dendritic trees of the pyramidal neurons. Thus, in the case of apical dendrites, these inputs can gate the transfer of the specific sensory responses in the distal dendrites to the soma.

The essential elements of the olfactory cortex basic circuit can thus be summarized as follows: (1) pyramidal output neurons, with apical and basal dendritic fields; (2) differentiation of pyramidal neurons into subtypes in sublayers; (3) reception of excitatory inputs by dendritic spines; (4) different modes of input driving: direct excitation, feedforward inhibition; (5) intrinsic recurrent axon collaterals for feedback and lateral inhibition; (6) intrinsic recurrent axon collaterals for feedback and lateral excitation; and (7) lamination of inputs to the dendritic trees of pyramidal neurons. Taken together, these constitute a unique set of circuit elements that not only is characteristic of olfactory cortex but also is shared with the other type of three-layered cortex, the dentate-hippocampal complex. Furthermore, these elements are also embedded in most regions of neocortex, where they are further elaborated into additional layers, additional subtypes of neurons and internal circuits, and additional types of inputs and outputs (see Shepherd, 1998).

Network Models of Olfactory Cortex

The first suggestion that olfactory cortex could serve as a simple model for learning and memory in cortical networks was made by Lewis Haberly. In a landmark paper (Haberly, 1985), he described the features of the olfactory cortex outlined above, and pointed out that this organization, distributed in a broad sheet, would subserve the functions of a cortex with content-addressable memory. The critical features were the widespread distribution of inputs by the input fibers and the presence of recurrent excitation, providing for a wealth of combinatorial possibilities for activation and reactivation of the cortical circuits. He further pointed out the possible

similarities between processing of the olfactory input by olfactory cortex and processing of complex visual stimuli in the face area of the neocortex.

These suggestions stimulated studies by several laboratories that have established the olfactory cortex as an attractive subject for network models of cortical functions. We will summarize briefly some of the main studies to date.

Ambros-Ingerson, Granger, and Lynch (1990) drew on the basic olfactory circuit to discuss the principles of a model of piriform (olfactory) cortex that would function as an associative memory network having the ability to identify conjunctions of odor components that constitute complex odors. The role of piriform cortex in olfactory memory was contrasted with the role of the hippocampus in maintaining or enabling the establishment of long-term olfactory associations. A reduced model of the basic olfactory circuit was described that incorporated properties of LTP and that implements a "combinatorial memory system." In this model, during learning, novel combinations of stimulus features result in unique representations. Complex stimuli composed of previously experienced odor components produce a response that is biased toward the existing representations within the cortex. It was proposed that piriform cortex could be regarded as a model for a general cortical memory representational system of this type.

In pursuing this model, Ambros-Ingerson et al. (1990) obtained results that led to a proposal that interactions between the olfactory bulb and olfactory cortex, with synaptic modification in the input pathway, could result in a form of hierarchical clustering that could serve to construct perceptual hierarchies used for storage and recognition of complex olfactory stimuli. This was a departure from earlier models, which had explored the role of intrinsic excitatory connections in associative memory functions. In this model, the olfactory cortex selectively inhibits previously active olfactory bulb neurons. The response to subsequent odor presentations leads

to responses that reflect the differences between stimuli. While experimental work reported a tendency for cortical response generalization following the presentation of a number of similar stimuli, supporting the type of clustering predicted by this model, this intriguing hypothesis awaits further experimental testing.

Building on the work of Haberly and his collaborators, Wilson and Bower (1992) used the piriform cortex to approach the investigation of cortical function on two fronts. First, compartmental models of pyramidal neurons based on anatomical and physiological data, as reviewed above, were used to simulate intracellular potentials, extracellular field potentials, and ensemble impulse activity as recorded experimentally in response to orthodromic volleys in the lateral olfactory tract. Second, this network model, constrained by physiological response properties, was used for simulations that attempted to demonstrate computational properties that would underlie its functions as an associative memory. The results of this study showed the ability to store and retrieve patterned impulse information in a network that displayed the physiological responses and temporal dynamics of the real cortex, using only a local Hebbian-type rule for modification of intrinsic excitatory synaptic interactions (Figure 3).

An interesting aspect of these simulations was the suggestion of a role for oscillatory phenomena, such as the prominent gamma range (30–100 Hz) extracellular oscillations. It was proposed that these oscillations coordinate computational processes that directly underlie the associative memory function of piriform cortex (Wilson and Bower, 1992). Oscillatory phenomena have become one of the chief subjects of interest among workers pursuing models of piriform cortex.

Based on this model, Liljenstrom (1991) drew on the work of Hopfield to develop further a model that used simplified sigmoidal output units. The local circuits for feedforward and feedback inhibition, together with the excitatory interactions between pyramidal neurons, were critically important for the input-output dynamics. The model demonstrated simultaneous slow theta and rapid gamma oscillations characteristic of olfactory cortex. It reproduced the experimental effects of acetylcholine, both on the modulation of these oscillations by selectively increasing excitability and suppressing intrinsic synaptic transmission and on associative memory functions (reviewed in Hasselmo and Linster, 1999).

Hasselmo developed both simplified network models and compartmental network simulations to investigate learning mechanisms (see Hasselmo and Linster, 1999). These models explored the effects of selective modification of both input and intrinsic excitatory connections. Associative memory performance was enhanced by the combination of suppression of intrinsic fiber transmission and increased excitability during learning, as seen with cholinergic modulation. The model could also exhibit effective associative memory properties under these conditions. This work represents a synthesis of abstraction and physiological detail in modeling of cortical function. It was proposed that this model could represent a basic unit for learning and memory in cortical circuits. This model has helped to guide subsequent experimental studies on the suppressive modulatory actions of cholinergic inputs and interpret their implications for the associative memory functions of the network (see Linster and Hasselmo, 2001, for review). Based on this modeling, several behavioral experiments on cholinergic modulation from Hasselmo's group have shown that cholinergic modulation is indeed involved in the processing and learning of overlapping olfactory stimuli (Linster and Hasselmo, 2001). This group has also proposed a possible role for the neuromodulator noradrenaline (NA) in signal-to-noise enhancement in piriform cortex; in that study, data on noradrenergic suppression of synaptic transmission in piriform cortex slices were combined with a model of odor processing in piriform cortex.

This illustrates a noteworthy aspect of olfactory cortical models: their close application to guiding and interpreting experimental

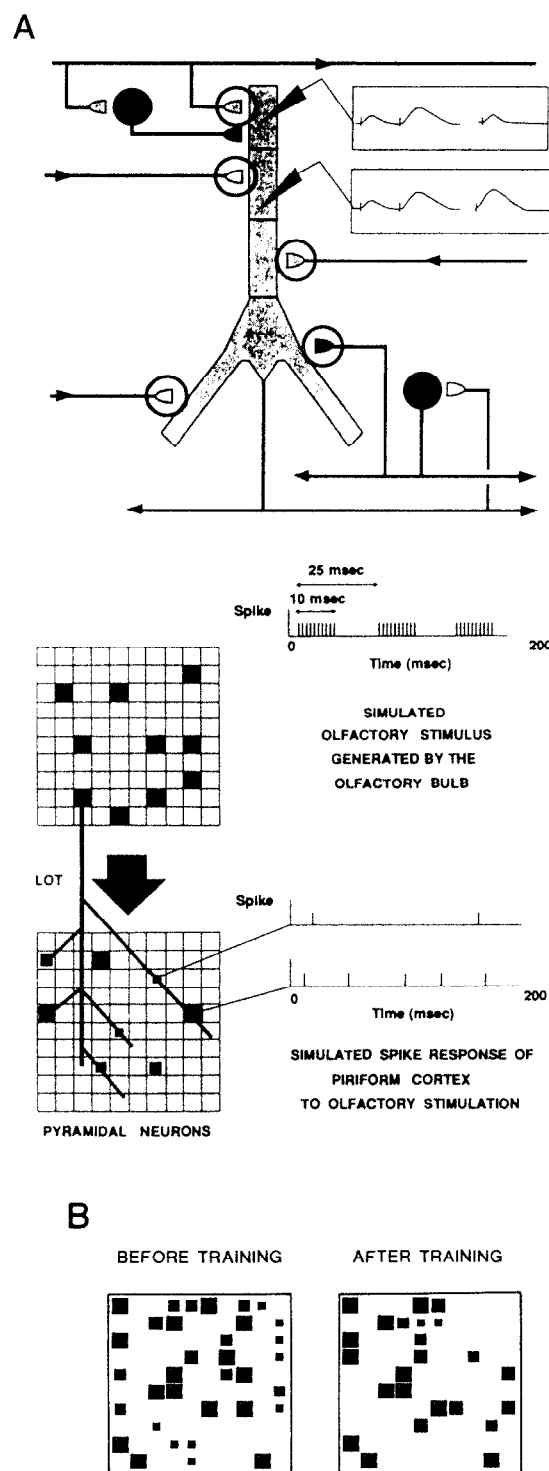


Figure 3. Autoassociation memory functions of a network model based on the basic circuit of Figure 2. *A*, Top, pyramidal neuron model with excitatory (open) and inhibitory (filled) synaptic inputs. Circles indicate synapses where Hebbian modification rules applied during learning. Middle, bulbar input patterns to the cortex (spike activity indicated by size of square). *B*, Responses of pyramidal neurons before and after learning produced a stable activity pattern. (Modified from Hasselmo et al., 1990.)

studies of olfactory cortex. Cattarelli and her colleagues (Litaudon, Datiche, and Cattarelli, 1997) have also used this approach to identify different functional regions within the olfactory cortex.

Nonlinear and chaotic properties of olfactory cortex were studied by Freeman (1987), who showed that a simple cortical model could display the chaotic properties of EEG rhythms; he proposed that a feedback gain parameter Q sets the level of arousal of the cortex. It is this factor that appears to be sensitive to the modulatory actions of acetylcholine, as described above. Simulated actions of acetylcholine on this parameter in the simplified olfactory cortical model produce point attractor, limit cycle attractor, and strange chaotic or nonchaotic attractor behavior (Liljenstrom, 1991). Recently, several theoretical papers have shown that the neural dynamics observed in olfactory cortex can be modeled to encode odor stimuli in spatiotemporal activity patterns (corresponding to limit cycles) (Li and Hertz, 2000) that resemble those described experimentally (Freeman, 1987) in the olfactory bulb.

An important development in studies of olfactory cortical networks is the realization that their operations are closely interrelated with those of the olfactory bulb, so that they function as an integrated system. An analogy with thalamocortical systems may be made in this regard. This has been recognized in both experimental and computational approaches. Fukai (1996) analyzed the importance of mutual feedback interactions between the olfactory bulb and cortex in a model that represented both structures as chained oscillators. The models incorporated the backprojection from cortical units to the bulbar oscillators in particular ways. Both structures exhibited rapid and robust synchronous oscillations in the presence of odorant stimuli, while they showed either nonoscillatory states or propagating waves in the absence of stimuli, depending on the values of model parameters. Feedback interactions between the two structures were shown to enhance the establishment of large-scale synchrony. The results suggest, in agreement with experimental data, that the modulation of neural activity through centrifugal inputs may play an important role at the early stage of cortical information processing.

A second example of combined olfactory bulb–olfactory cortex networks is the model of Li and Hertz (2000). They postulated that olfactory bulb to olfactory cortex transmission encodes recognition of an odor, while feedback from cortex to bulb generates odor segmentation by producing adaptation that is odor specific. As discussed in OLFATORY BULB (q.v.), this allows the system to recognize a subsequent novel odor. Independently, Wilson (2000) has provided experimental evidence in single-unit recordings from olfactory cortex that cell responses do habituate to a given odor, that this habituation is associated with a decrease in amplitude of excitatory postsynaptic potentials (EPSPs) from the olfactory bulb, and that this habituation is odor specific. Other examples of analysis of the distributed system that includes olfactory bulb and olfactory cortex include Chabaud et al. (1999).

Running through much of current studies, both experimental and theoretical, of odor processing is the question of the relative importance of spatial and temporal patterns in encoding the information contained in different odorous compounds. The presence of spatial patterns, forming virtual internal “odor images” within the olfactory bulb, has been demonstrated by many methods. However, these patterns are not seen in the olfactory cortex. Temporal patterns are seen in both the responses of single cells and summed EEG potentials, and a strong case has been made on both experimental and computational grounds for their role in encoding odors (see Freeman, 1987). It is important to recognize that both space and time are involved in encoding odors, at all stages of odor processing, as in the processing of other senses; construction of networks is not complete without incorporation of both aspects.

In view of the similarities in organization of the olfactory bulb of vertebrates and the antennal lobe of insects (Hildebrand and

Shepherd, 1997), it is of interest to inquire into possible similarities between olfactory cortex and the next stage in the insect olfactory pathway, the mushroom bodies. Several properties that may be involved have been identified experimentally; these properties include long-lasting inhibitory potentials, synaptic plasticity, and wide interconnectivity. It has been proposed that these properties, incorporated into a network model, are well suited for decoding temporal patterns specific for given odors in the dendrites of Kenyon cells in the mushroom bodies.

The models of olfactory cortical function discussed here have been at the neuronal or circuit/network level. The behaviors at these levels depend in turn on properties at the membrane and synaptic level, and on the differential properties of cell bodies and dendrites. An initial step toward assessing these has been made by constructing compartmental models of apical dendrites and their spines and analyzing their responses to excitatory and inhibitory inputs. Basic logic operations of AND, OR, and NOT-AND were found to arise from spine interactions when active membrane properties were placed in the spines or the dendritic branch (Shepherd, Woolf, and Carnevale, 1989). These studies support the idea, discussed earlier, that distal dendrites of pyramidal neurons receive and process rapid, precise input information, rather than mediating only slow and weak background modulation. They emphasize the importance of including apical and dendritic properties in network models to represent the full complexity of cortical circuits and cortical functions.

Summary

The organization of the olfactory cortex can be summarized by a basic circuit composed of a unique set of elements, which may be embedded and elaborated in more complicated cortical regions. Models of olfactory cortex emphasize the importance of cortical dynamics, including the interactions of intrinsic excitatory and inhibitory circuits and the role of oscillatory potentials, in generating the computations performed by the cortex. This replaces earlier interpretations of simple modulation or synchronization of activity within the cortex. In this way, the olfactory cortex, and neural networks based on it, may serve as a useful approach to the study of computations defined by cortical dynamics. It is also recognized that the olfactory cortex functions in close interaction with the olfactory bulb, forming a bulbar-cortical system, in processing odor inputs.

Acknowledgments. C. L. is supported by a fellowship from the Alfred P. Sloan foundation and M. H. by grants NIH60013, NIH61492, and NIH60450. M. W. has been supported by grants from the NSF and ONR. G. M. S. is supported by RP1 DC 00086-34; PO1 DC 04732-02 under the Human Brain Project; and the Department of Defense under a Multiple University Research Initiative.

Road Maps: Mammalian Brain Regions; Other Sensory Systems

Related Reading: Evolution of the Ancestral Vertebrate Brain; Hippocampal Rhythm Generation; Olfactory Bulb; Perspective on Neuron Model Complexity

References

- Ambros-Ingerson, J., Granger, R., and Lynch, G., 1990, Simulation of paleocortex performs hierarchical clustering, *Science*, 247:1344–1348.
- Chabaud, P., Ravel, N., Wilson, D. A., and Gervais, R., 1999, Functional coupling in rat central olfactory pathways: A coherence analysis, *Neurosci. Lett.*, 276:17–20.
- Freeman, W. J., 1987, Simulation of chaotic EEG patterns with a dynamic model of the olfactory system, *Biol. Cybern.*, 56:139–150.

- Fukui, T., 1996, Bulbocortical interplay in olfactory information processing via synchronous oscillations, *Biol. Cybern.*, 74:309–317.
- Haberly, L. B., 1985, Neuronal circuitry in olfactory cortex: Anatomy and functional implications, *Chem. Senses*, 10:219–238. ♦
- Hasselmo, M. E., and Linster, C., 1999, Modeling the piriform cortex, *Cereb. Cortex*, 13:525–560. ♦
- Hildebrand, J. G., and Shepherd, G. M., 1997, Mechanisms of olfactory discrimination: Converging evidence for common principles across phyla, *Annu. Rev. Neurosci.*, 20:595–631.
- Li, Z., and Hertz, J., 2000, Odour recognition and segmentation by a model of olfactory bulb and cortex, *Netw. Comput. Neural Syst.*, 11:83–102.
- Liljenstrom, H., 1991, Modeling the dynamics of olfactory cortex using simplified network units and realistic architecture, *Int. J. Neural Syst.*, 2:1–15.

- Linster, C., and Hasselmo, M. E., 2001, Neuromodulation and the functional dynamics of piriform cortex, *Chem. Senses*, 26:585–594, Review. ♦
- Litaudon, P., Datiche, F., and Cattarelli, M., 1997, Optical recording of the rat piriform cortex activity, *Prog. Neurobiol.*, 52:485–510.
- Shepherd, G. M., Ed., 1998, *The Synaptic Organization of the Brain*, 4th ed., New York: Oxford University Press. ♦
- Shepherd, G. M., Woolf, T. B., and Carnevale, N. T., 1989, Comparisons between active properties of distal dendritic branches and spines: Implications for neuronal computations, *J. Cogn. Neurosci.*, 1:273–286.
- Wilson, D. A., 2000, Odor specificity of habituation in the rat anterior piriform cortex, *J. Neurophysiol.*, 83:139–145.
- Wilson, M., and Bower, J. M., 1992, Cortical oscillations and temporal interactions in a computer simulation of piriform cortex, *J. Neurophysiol.*, 67:981–995.

Optimal Sensory Encoding

Li Zhaoping

Introduction

What is optimal depends on computational tasks. Many recent works define optimality in information-theoretic terms, such as information transmission rates. This can be particularly relevant in the early stages of vision, which are mainly concerned with transmitting information indiscriminately. In this article we focus on the better-known visual system to discuss optimal sensory encoding, although encoding in other sensory systems can be addressed by similar avenues.

Consider a simplified visual input model with, say, $1,000 \times 1,000$ pixels arranged in a regular grid at one byte per pixel and 20 images per second. This model provides many megabytes per second of raw data. Given the information bottleneck in the long optic nerve from retina to thalamus and the limited firing rates (thus limited data capacity) of cortical neurons (see SENSORY CODING AND INFORMATION TRANSMISSION), early vision can greatly benefit from a data encoding that reduces the rate of data transmission without significant information loss. Since nearby image pixels tend to convey similar signals (e.g., luminance values) and thus carry redundant information, significant savings can be achieved by avoiding transmitting the information redundantly. If, within a particular time window, each original pixel codes one byte of information, 80% of which is redundant information shared with neighboring pixels, then one million pixels code only 200 kilobytes of nonredundant information. One way to avoid redundancy is to transform the original signal $\mathbf{S} = \{S_1, S_2, \dots, S_N\}$ in the N neurons (e.g., photoreceptors) to signals $\mathbf{O} = \{O_1, O_2, \dots, O_M\}$ in another M (more/fewer) neurons (e.g., the retinal ganglion cells or cortical neurons), such that signals in O_i and O_j for all i, j are not significantly redundant. Consequently, 200 kilobytes of information in \mathbf{S} could be coded by only 0.2 bytes in each neuron O_i if $M = N$, which needs a much reduced firing rate. Lossless encoding means that, if needed, \mathbf{S} can be reconstructed from \mathbf{O} . Such observations have led to the *Infomax* proposal, namely, that early vision constructs an optimal coding of input to allow maximum information transmission from retina to cortex under limited channel capacity of the optic nerve or neural activities (Attneave, 1954; Barlow, 1961; Linsker, 1990; Atick, 1992). This principle has provided many insights into the properties of the receptive fields in early vision.

Optimal Encoding Illustrated by Stereo Vision

Consider the redundancy and encoding of stereo signals (Li and Atick, 1994a). Let S_L and S_R be the signals to the left and right eyes

(Figure 1). They may be the average luminance in the images or the Fourier components (of a particular frequency) of the images. Assume that they have zero mean (for simplicity) and equal variance (or signal power) $\langle S_L^2 \rangle = \langle S_R^2 \rangle$ ($\langle \dots \rangle$ denotes average over the input ensemble). The redundancy is seen in the correlation matrix:

$$R^S = \begin{pmatrix} \langle S_L^2 \rangle & \langle S_L S_R \rangle \\ \langle S_R S_L \rangle & \langle S_R^2 \rangle \end{pmatrix} = \langle S_L^2 \rangle \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$$

where $0 \leq r \leq 1$ is the correlation coefficient between S_L and S_R . The value of r is high, $r \rightarrow 1$, for mean luminance signals $S_{L,R}$, but low, $r \rightarrow 0$, if $S_{L,R}$ are a high spatial frequency Fourier component of the respective images. A simplifying assumption is that \mathbf{S} are Gaussian signals, which are defined to have a probability distribution $P(\mathbf{S}) \propto \exp(-\sum_{ij} S_i S_j (R^S)^{-1}_{ij} / 2)$. An encoding

$$O_+ = S_+ \equiv (S_L + S_R)/\sqrt{2}, \quad O_- = S_- \equiv (S_L - S_R)/\sqrt{2}$$

gives zero correlation $\langle O_+ O_- \rangle$ in \mathbf{O} , leaving output probability $P(\mathbf{O}) = \prod_i P(O_i)$ factorized, as is easily verified. The transform $\mathbf{S} \rightarrow \mathbf{O}$ is linear, which approximates the cell response properties in the retina and, to a less degree, in primary visual cortex. The cell coding O_+ is a binocular cell, owing to the binocular summation of inputs, while the cell coding O_- is monocular or ocularly opponent. Note that S_{\pm} are the eigenvectors of the correlation matrix R^S , or the principal components of the signals, and their signal power $\langle S_{\pm}^2 \rangle = (1 \pm r) \langle S_L^2 \rangle$ is the corresponding

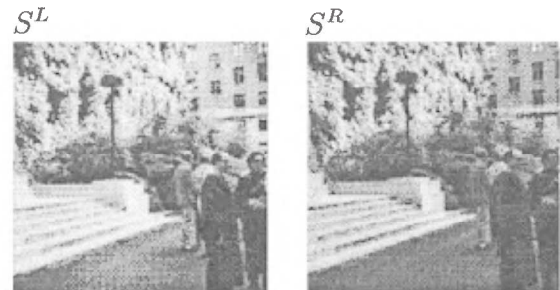


Figure 1. A stereo pair input to the two eyes.

eigenvalues. In reality, input noise \mathbf{N} is added on \mathbf{S} , and the coding transform introduces additional noise \mathbf{N}_o ; hence, $O_{\pm} = [(S_L + N_L) \pm (S_R + N_R)]/\sqrt{2} + N_{o,\pm}$, giving effective output noise $N_{\pm} = (N_L \pm N_R)/\sqrt{2} + N_{o,\pm}$. For simplicity, the noise terms are assumed to be independent of each other and of the signals. Let $\langle N^2 \rangle \equiv \langle N_L^2 \rangle = \langle N_R^2 \rangle$, and $\langle N_o^2 \rangle \equiv \langle N_{o,+}^2 \rangle = \langle N_{o,-}^2 \rangle$. Input $S_{L,R} + N_{L,R}$ has

$$I_{L,R} = \frac{1}{2} \log_2 \frac{\langle S_{L,R}^2 \rangle + \langle N^2 \rangle}{\langle N^2 \rangle}$$

bits of (mutual) information about $S_{L,R}$, since, for Gaussian signals and noise, the information amount is $(1/2)\log_2$ (signal-to-noise), whereas O_{\pm} has

$$I_{\pm} = \frac{1}{2} \log_2 \frac{\langle O_{\pm}^2 \rangle}{\langle N_{\pm}^2 \rangle} = \frac{1}{2} \log_2 \frac{\langle S_{\pm}^2 \rangle + \langle N^2 \rangle + \langle N_o^2 \rangle}{\langle N^2 \rangle + \langle N_o^2 \rangle}$$

bits of information about $S_{L,R}$ or S_{\pm} . Note that the redundancy between S_L and S_R causes higher or lower signal powers $\langle O_{+}^2 \rangle$ or $\langle O_{-}^2 \rangle$ in O_{+} or O_{-} , respectively, leading to a higher or lower information rate I_{+} or I_{-} . As an initial choice, define cost as the total signal power, although there can be many other cost considerations (discussed later). Since $I_{\pm} = (1/2)\log_2(\langle O_{\pm}^2 \rangle) + \text{constant} = (1/2)\log_2(\text{cost}) + \text{constant}$, we note that the gain in information per unit cost ($\Delta I/\Delta \text{cost}$) is smaller in the O_{+} than in the O_{-} channel. This motivates a reduction (increment) of costs in the O_{+} (O_{-}) channels by introducing the gains V_{\pm} , such that $O_{\pm} = V_{\pm}[(S_L + N_L) \pm (S_R + N_R)]/\sqrt{2} + N_{o,\pm}$, at the expense (benefit) of the information transmitted:

$$I_{\pm} = \frac{1}{2} \log_2 \frac{V_{\pm}^2(\langle S_{\pm}^2 \rangle + \langle N^2 \rangle) + \langle N_o^2 \rangle}{V_{\pm}^2 \langle N^2 \rangle + \langle N_o^2 \rangle} \quad (1)$$

Hence, the optimal encoding, balancing the cost and information extraction, is to find the gains V_{\pm} to minimize

$$\begin{aligned} E(V_{\pm}) &\equiv \sum_a (\langle O_a^2 \rangle) - \lambda \sum_a (I_a) \\ &= \text{cost} - \lambda \cdot \text{Information} \end{aligned} \quad (2)$$

where λ is the Lagrange multiplier whose value determines the balance. The optimal gains can be obtained by $\partial E/\partial V_{\pm} = 0$ to give

$$\begin{aligned} V_{\pm}^2 &\propto \text{Max} \left\{ \left[\frac{1}{2} \frac{\langle S_{\pm}^2 \rangle}{\langle S_{\pm}^2 \rangle + \langle N^2 \rangle} \right. \right. \\ &\quad \times \left. \left(1 + \sqrt{1 + \frac{4\lambda}{\log 2} \frac{\langle N^2 \rangle}{\langle N_o^2 \rangle} \frac{\langle S_{\pm}^2 \rangle}{\langle S_{\pm}^2 \rangle + \langle N^2 \rangle}} \right) - 1 \right], 0 \right\} \end{aligned} \quad (3)$$

In the zero noise limit, when $(\langle S_{\pm}^2 \rangle/\langle N^2 \rangle) \gg 1$, $V_{\pm}^2 \propto \langle S_{\pm}^2 \rangle^{-1}$. As expected, this suppresses the stronger ocular summation signal S_{+} and amplifies the weaker ocular contrast signal S_{-} , in order to save the cost, since the cost increases linearly with V_{\pm}^2 , but the extracted information increases only logarithmically with V_{\pm}^2 . Hence, for instance, when the coding noise \mathbf{N}_o is negligible (i.e., $(\langle N_o^2 \rangle/V_{\pm}^2 \langle N^2 \rangle) \ll 1$), output \mathbf{O} and the original input $\mathbf{S} + \mathbf{N}$ contain about the same amount of information about the true signal \mathbf{S} , but \mathbf{O} consumes much less power with $V_{+} \ll V_{-} < 1$, when $r \sim 1$. This gain $V_{\pm} \propto \langle S_{\pm}^2 \rangle^{-1/2}$ also equalizes output power $\langle O_{+}^2 \rangle \approx \langle O_{-}^2 \rangle$, since $\langle O_{\pm}^2 \rangle = V_{\pm}^2 \langle S_{\pm}^2 \rangle + \text{noise power}$, making the output correlation matrix R^o (with elements $R_{ab}^o = \langle O_a O_b \rangle$) proportional to an identity matrix (since $\langle O_{+} O_{-} \rangle = 0$). Such a transform $\mathbf{S} \rightarrow \mathbf{O}$, which leaves output channels decorrelated and equally powered, is called *whitening*. Any rotation $\mathbf{O} \rightarrow \mathbf{UO}$ via a rotation or unitary transform $\mathbf{U}(\mathbf{U}\mathbf{U}^T = \mathbf{I})$, by angle θ in the two-dimensional space

\mathbf{O} , multiplexes the channels O_{+} and O_{-} to give two alternative channels

$$\begin{pmatrix} O_1 \\ O_2 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} O_{+} \\ O_{-} \end{pmatrix} = \begin{pmatrix} \cos(\theta)O_{+} + \sin(\theta)O_{-} \\ -\sin(\theta)O_{+} + \cos(\theta)O_{-} \end{pmatrix}$$

which are also decorrelated ($\langle O_1 O_2 \rangle = 0$). Furthermore, note from Equations 2 and 1 that cost = $\text{Tr}(R^o)$ and Information = $(1/2) \log (\det R^o)/(\det R^N)$, where R^N is the correlation matrix of the noises in the output channel and $\text{Tr}(\cdot)$ and $\det(\cdot)$ denote the trace and determinant of a matrix. Since both the trace and the determinant are invariant to unitary transforms (rotations), the optimized objective function $E = (\text{cost} - \lambda \text{Information})$ is invariant to this rotation $O_{\pm} \rightarrow O_{1,2}$. Hence, both encoding schemes $S_{L,R} \rightarrow O_{\pm}$ and $S_{L,R} \rightarrow O_{1,2}$, with the former a special case of the latter, are equally optimal in making the output decorrelated (nonredundant), in extracting information about $S_{L,R}$, and in saving the coding cost $\sum_a \langle (O_a)^2 \rangle$. Since

$$\begin{pmatrix} O_1 \\ O_2 \end{pmatrix} = \begin{pmatrix} S_L(\cos(\theta)V_{+} + \sin(\theta)V_{-}) + S_R(\cos(\theta)V_{+} - \sin(\theta)V_{-}) \\ S_L(-\sin(\theta)V_{+} + \cos(\theta)V_{-}) + S_R(-\sin(\theta)V_{+} - \cos(\theta)V_{-}) \end{pmatrix}$$

in general O_1 and O_2 prefer different eyes. In particular, $\theta = -45^\circ$ gives $O_{1,2} \propto S_L(V_{+} \mp V_{-}) + S_R(V_{+} \pm V_{-})$. The visual cortex indeed has neurons of a whole spectrum of ocularities.

Variations of Optimal Encodings

It is now apparent that infomax coding as defined in Equation 2 is related to whitening, decorrelation, principal component analysis, and *factorial codes*, defined as when probabilities of signals factorize $P(\mathbf{O}) = \prod_a P(O_a)$. Among the many other relatives of optimal codings are *minimum entropy* or *minimum description length*, since minimizing $\langle O_1^2 \rangle + \langle O_2^2 \rangle$ reduces the total output entropy $H(O_1) + H(O_2)$ ($H(\cdot)$ stands for entropy) for Gaussian signals O_a ; *independent component analysis*, since principal components are independent components for Gaussian signals; *redundancy reduction*, since the well-known inequality $\sum_a H(O_a) > H(\mathbf{O})$ means that minimizing $\sum_a H(O_a)$ reduces the redundancy, intuitively defined as $\sum_a H(O_a)/H(\mathbf{O}) - 1 \geq 0$ (equal to zero when there is no redundancy), between output channels; *sparse coding*, since it is defined as lowering the coding bits $H(O_a)$ for all channels a ; *maximum entropy code*, since $H(\mathbf{O})$ is maximized given $\sum_a H(O_a)$ when redundancy is removed; *predictive codes*, since the code effectively predicts or explains away S_R from S_L to achieve minimum $\sum_a \langle O_a^2 \rangle$ for given $I(\mathbf{O}; \mathbf{S})$ (information in \mathbf{O} about \mathbf{S}); and *minimum predictability codes* or *least mutual information* between output channels, since $\sum_a H(O_a) = H(\mathbf{O})$ means zero mutual information between output channels O_a and O_b . All of these variations of “optimal encoding” often mean approximately or exactly the same (Nadal and Parga, 1997), depending on their precise definitions and the statistics of the signals concerned, and should not be thought of as independent coding principles.

Optimal Visual Encoding in Space, Time, Color, and Scale

In general, for simple linear encoding of approximately Gaussian signals \mathbf{S} , a recipe for optimal coding is visualized in Figure 2. Given input signal \mathbf{S} with noise \mathbf{N} , the encoding transform \mathbf{K} and additional coding noise \mathbf{N}_o gives output signal $\mathbf{O} = \mathbf{K}(\mathbf{S} + \mathbf{N}) + \mathbf{N}_o$. The optimal transform \mathbf{K} is dictated by the input statistics characterized by the correlation matrix R^S . The first step is principal component analysis, transforming $\{S_a\}$ via a matrix \mathbf{K}_o to the principal components $\{S_k\}$, i.e., $S = \mathbf{K}_o S$. The powers of the components S are the eigenvalues of R^S . Next, the optimal gain V_k to S_k is determined by S_k 's signal-to-noise ratio via Equation 3. A particular optimal coding transform is $\mathbf{K} = \mathbf{V}\mathbf{K}_o$, where \mathbf{V} is a diag-

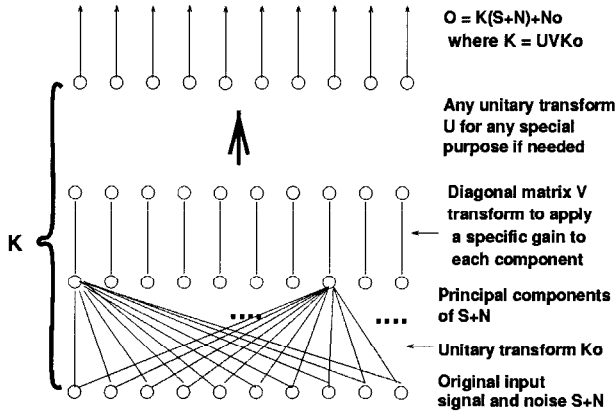


Figure 2. A schematic of the steps to obtain infomax (linear) code for Gaussian signals.

onal matrix with diagonal elements equal to the optimal gains V_k or $V(k)$. The resulting \mathbf{O} have decorrelated components and retain the maximum information about \mathbf{S} given output cost $\sum_a \langle O_a^2 \rangle$. Furthermore, any transform in the class $\mathbf{K} = \mathbf{UVK}_0$, where \mathbf{U} is any unitary transformation (rotation, $\mathbf{UU}^T = 1$), is equally optimal, since it leaves the outputs \mathbf{O} with the same information extraction and cost, and, in the zero noise limit, the same decorrelation. The conceptual steps above correspond mathematically to finding the (degenerate) solution \mathbf{K} of $\partial E / \partial \mathbf{K} = 0$, where $E(\mathbf{K}) = \text{cost} - \lambda$ Information.

In spatial coding (Atick, 1992), the signal at visual location x is S_x . Since the signal correlation is translation invariant, i.e., $\langle S_x S_{x'} \rangle$ is a function of only $x - x'$, the principal components are Fourier modes, and K_0 is the Fourier transform $K_0^{kx} \sim e^{-ikx}$ such that $S_x \rightarrow S_k \sim \sum_x K_0^{kx} S_x \sim \sum_x e^{-ikx} S_x$. Field (1987) measured the power spectrum as $\langle S_k^2 \rangle \sim 1/k^2$ with Fourier frequency k . Assuming white noise power $\langle N^2 \rangle$, the high signal-to-noise ratio S^2/N^2 in the low- k region leads to the gain V_k or $V(k) \propto k$ that increases with k . However, for high k , where S^2/N^2 is low, $V(k)$ quickly decays with increasing k to zero, according to Equation 3, in order not to amplify noise. This gives a bandpass $V(k)$ as a function of k (Figure 3). If \mathbf{U} is the inverse Fourier transform $U^{x'k} \sim e^{ikx'}$, then the whole transform $\mathbf{K} = \mathbf{UVK}_0$ transforms signal S_x to activities $O_{x'}$ of a neuron with a receptive field at location x' as a bandpass filter, i.e., $O_{x'} \sim \sum_k V(k) \sum_x e^{ik(x'-x)} S_x + \text{noise}$. This is roughly what retinal output (ganglion) cells do, achieving a center-surround transform on the input image and emphasizing the intermediate-frequency band where the signal-to-noise ratio is of order 1. Function $V(k)$ is the well-known contrast sensitivity function. When the visual environment dims down, reducing the overall signal-to-noise ratio $\langle S_k^2 \rangle / \langle N^2 \rangle$ in all frequencies, say from $(\langle S_k^2 \rangle / \langle N^2 \rangle) \sim 100/k^2$ to $(\langle S_k^2 \rangle / \langle N^2 \rangle) \sim 1/k^2$, the bandpass region should shift toward lower frequencies, effectively making $V(k)$ a low pass. This explains the dark adaptation of the retinal ganglion cells' receptive fields, from a center-surround contrast-enhancing (bandpass) filter to a Gaussian-like smoothing (low-pass) filter, to integrate signals and smooth out noise.

Encoding in time is analogous to encoding in space. Image statistics in time (Dong and Atick, 1995) determine the temporal frequency sensitivities $V(\omega)$ (of frequency ω) of the optimal temporal filter. Given a sustained input $S(t)$ over time t , the output $O(t)$ may be more sustained or transient depending on whether the filter is more low pass or band pass. By an appropriate choice of the rotation transform \mathbf{U} (Dong and Atick, 1995; Li, 1996), the temporal filter can be made causal, i.e., the output \mathbf{O} depends only on input \mathbf{S} of the past but not the future.

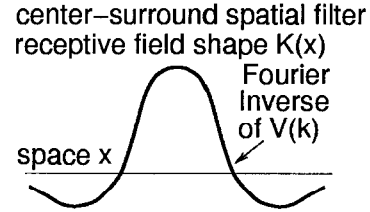
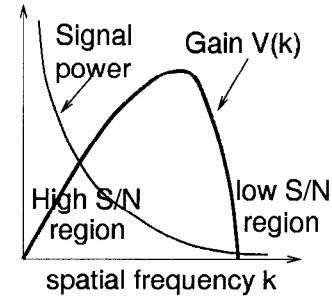


Figure 3. The contrast gain $V(k)$ as a function of spatial frequency k , determined from the signal-to-noise ratio (S/N) of the inputs ($S + N$) at that frequency. The corresponding spatial filter $K(x)$ is the Fourier inverse of $V(k)$, adopted by the retinal ganglion cells on the photoreceptor inputs.

Visual color encoding (Atick, 1992) is analogous to stereo encoding. The inputs are three-dimensional (3D), S_r , S_g , and S_b , for red, green, and blue signals. The principal components include a strong luminance channel, a weighted summation of the cone inputs, and two weaker chrominance channels, one roughly red-green opponency and another yellow-blue opponency. Optimal encoding then involves appropriate gains to these channels and additional multiplexing of them as needed. Physiologically, color and space codings are coupled, resulting, for instance, in the red-center, green-surround receptive fields (Figure 4) of the retinal ganglion cells. This can be understood in a simplified two-cone system, red and green. The high signal-to-noise luminance channel ($S_r + S_g$) needs a center-surround or bandpass spatial filter, while the low signal-to-noise chromatic channel ($S_r - S_g$) needs a smoothing or low-pass filter. The multiplexing of these two channels, a rotational operation \mathbf{U} in the 2D color space, leads to addition or subtraction

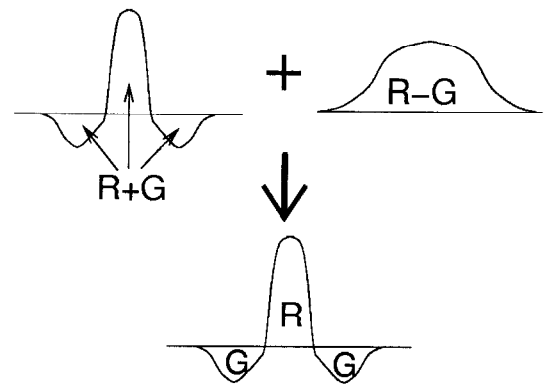


Figure 4. Multiplexing the center-surround achromatic ($R + G$) filter with the chromatic ($R - G$) Gaussian-like filter gives a red-center, green-surround double (in space and in color) opponency receptive field observed in retina.

of these two filters. The results are the red-center, green-surround or green-center, red-surround receptive fields. In the retina and/or primary visual cortex, codings in space, time, color, and stereo are all coupled together (Atick, 1992; Li and Atick, 1994a,b; Li, 1996).

Multiscale Encoding in the Primary Visual Cortex

Primary visual cortex receives retinal outputs via the lateral geniculate nucleus. Its receptive fields are orientation selective in the shape of small bars or edges. Different receptive fields have different orientations and different sizes (or tuned to different spatial frequency bands), in a multiscale fashion such that receptive fields of different sizes are roughly scaled versions of each other, also called wavelet coding. These receptive fields can be seen as components of another optimal code by a particular choice of the rotation (unitary) matrix \mathbf{U} in the coding transform $\mathbf{K} = \mathbf{U}\mathbf{V}\mathbf{K}_0$. Retinal receptive fields are given when $\mathbf{U} = \mathbf{K}_0^{-1}$, and are theoretically the same for all retinal ganglion cells except for a spatial translation. Another optimal code, apparently not adopted anywhere in our visual system, is when $\mathbf{U} = \mathbf{I}$, an identity matrix. The receptive fields would be infinitely large, and each would be unique and a particular principal component (Fourier component) with a particular gain. The \mathbf{U} transform for the multiscale coding is when \mathbf{U} is somewhere in between the two extremes $\mathbf{U} = \mathbf{K}_0^{-1}$ and $\mathbf{U} = \mathbf{I}$. To construct a cortical receptive field, \mathbf{U} multiplexes the principal components (Fourier waves) within a finite frequency range $\mathbf{k} \in (\mathbf{k}_1, \mathbf{k}_2)$ such that the resulting receptive field is responsive only to a restricted range of orientations and spatial frequencies \mathbf{k} . The code can be viewed as an intermediate between the Fourier wave code, where each receptive field is infinitely large and responds to only one frequency and orientation, and the retinal code, where each receptive field is small and responsive to all frequencies k and all orientations. Different cortical units cover different ranges of frequencies to give a complete sampling (Li and Atick, 1994b).

It has been argued (reviewed by Simoncelli and Olshausen, 2001) that the multiscale code, which should be as good as the retinal code if the visual inputs assume Gaussian statistics, is actually better in light of the actual non-Gaussian nature of the signals. Oriented receptive fields have been argued to capture the non-trivial third-order statistics, in particular the third-order correlation $\langle S_a S_b S_c \rangle$ between signals from three image pixels a , b , and c , that are not accounted for by Gaussian statistics. Previous works (Simoncelli and Olshausen, 2001) argued that the cortical orientation-selective receptive fields match the orientation features in inputs, and that the neurons are inactive unless those matches happen. The code is thus argued to be a sparser code, since the activities of different cells are supposedly less correlated (see SPARSE CODING IN THE PRIMATE CORTEX). Why doesn't retina adopt this code? One reason could be that the cortical representation is in addition overcomplete, i.e., the number M of cortical units (output units O_a) is orders of magnitude larger than the number N of the retinal units (input units S_a). The overcompleteness has been argued to improve sparseness, though at the expense of the neural proliferation, because cells tuned to different image features cannot be active together. However, it should be noted that if cortical activities \mathbf{O} depend linearly on visual input \mathbf{S} , the \mathbf{O} units are necessarily (mathematically) dependent on, or correlated with, each other in an overcomplete representation where $M > N$ (Li, 1996). Cortical response \mathbf{O} depends on visual input \mathbf{S} nonlinearly, by rectification, thresholding, saturation, and normalization, etc. (Simoncelli and Olshausen, 2001). The observed nonlinearity is unlikely to be sufficient to achieve decorrelation. However, the nonlinearity and the overcomplete representation are more likely to serve nontrivial cognitive computations (Li, 2002) beyond the traditional coding considerations.

Discussion

It is clear that maximizing information transmission alone is not enough to specify optimal codes. One may prefer one code or another when considering other costs and benefits (see, e.g., Levy and Baxter, 1996). The retinal code has the advantage of small and identical receptive field shapes that involve shorter neural wiring and easier specifications. It also has stronger correlation between output signals than the Fourier wave codes outside the zero noise limit (both codes should have zero second-order correlation in zero noise limit), making it easier for error correction purposes. Its translation invariance also allows an object translated laterally to induce the same pattern of neural activities except for a change in the responding neurons. When this invariance is extended to objects moving in depth (when images of objects change sizes), the cortical multiscale code is preferred. In this case many different receptive fields are scaled and/or translated versions of each other, leading to translation invariance within a scale and scale invariance between scales (Simoncelli et al., 1992).

More significant are optimality measures not based on information measures. For example, to give a best estimation \hat{S} of input S from $O = K(S + N) + N_o$, the optimal coding transform K to minimize the estimation error $\langle (S - \hat{S})^2 \rangle$ given output power $\langle O^2 \rangle$ certainly does not satisfy infomax. Another example is afforded by the two classes of the retina ganglion cells. Whereas the infomax principle explains well the receptive fields of the more numerous class of retinal ganglion cells, the P cells in monkeys or X cells in cats, another class of ganglion cells, M cells in monkeys or Y cells in cats, have receptive fields that are relatively larger, color unselective, and tuned to higher temporal frequencies. These M cells do not extract the maximum information possible (infomax) about input S , but can serve to extract the information as fast as possible (Li, 1992), i.e., the temporal outputs $(O(t = -\infty), \dots, O(t - 1), O(t))$ should contain some information about $S(t' \leq t)$ with a shortest possible delay $t - t'$. This observation should have significant implications for how P and M pathways should interact at later stages of processing.

Information theory provides excellent means to *quantify the amount* of information to design optimal coding for *information transmission*. Cognitive functions often require a selection of the *quality or modality* of information, which is beyond information theory. Information theory is more likely to find application in the early stages of sensory processing, before information is selected or discriminated for any specific cognitive task, when general purpose information transmission is the main concern. This explains the success of information theory in the retina and partly in the primary visual cortex, to the extent that there is quantitative agreement with experimental results and to the extent that information theory has predictive power for new data (Dong and Atick, 1995; Chen and Li, 1998). Optimal sensory coding in later stages of sensory pathways is expected to depend on cognitive tasks beyond simple information transmission, and should require applications of alternative theories in future research.

Road Map: Neural Coding

Related Reading: Feature Analysis; Information Theory and Visual Plasticity; Sensory Coding and Information Transmission; Unsupervised Learning with Global Objective Functions

References

- Atick, J. J., 1992, Could information theory provide an ecological theory of sensory processing? *Network: Computat. Neural Syst.*, 3:213–251. ♦
- Attneave, F., 1954, Informational aspects of visual perception, *Psychol. Rev.*, 61:183–193.

- Barlow, H. B., 1961, Possible principles underlying the transformations of sensory messages, in *Sensory Communication* (W. A. Rosenblith, Ed.), Cambridge, MA: MIT Press, pp. 217–234. ♦
- Chen, D., and Li, Z., 1998, A psychophysical experiment to test the efficient stereo coding theory, in *Theoretical Aspects of Neural Computation* (K. M. Wong, I. King, and D. Y. Yeung, Eds.), New York: Springer-Verlag.
- Dong, D. W., and Atick, J. J., 1995a, Temporal decorrelation: A theory of lagged and non-lagged responses in the lateral geniculate nucleus, *Network: Computat. Neural Syst.*, 6:159–178.
- Dong, D. W., and Atick, J. J., 1995b, Statistics of natural time-varying images, *Network: Computat. Neural Syst.*, 6:345–358.
- Field, D. J., 1987, Relations between the statistics of natural images and the response properties of cortical cells, *J. Opt. Soc. Am. A*, 4:2379–2394.
- Levy, W. B., and Baxter, R. A., 1996, Energy efficient neural codes, *Neural Computat.*, 8:531–543.
- Li, Z., 1992, Different retinal ganglion cells have different functional goals, *Int. J. Neural Syst.*, 3:237–248.
- Li, Z., 1996, A theory of the visual motion coding in the primary visual cortex, *Neural Computat.*, 8:705–730.
- Li, Z., and Atick, J. J., 1994a, Efficient stereo coding in the multiscale representation version *Network: Computat. Neural Syst.*, 5:157–174. ♦
- Li, Z., and Atick, J. J., 1994b, Towards a theory of striate cortex, *Neural Computat.*, 6:127–146. ♦
- Li, Z., 2002, A saliency map in primary visual cortex, *Trends Cog. Sci.*, 6:9–16. ♦
- Linsker, R., 1990, Perceptual neural organization: Some approaches based on network models and information theory, *Annu. Rev. Neurosci.*, 13:257–281.
- Nadal, J.-P., and Parga, N., 1997, Redundancy reduction and independent component analysis: Conditions on cumulants and adaptive approaches, *Neural Comput.*, 9:1421–1456.
- Simoncelli, E. P., Freeman, W. T., Adelson, E. H., and Heeger, D. J., 1992, Shiftable multiscale transforms, *IEEE Trans. Inf. Theory*, 38:587–607.
- Simoncelli, E., and Olshausen, B., 2001, Natural image statistics and neural representation, *Annu. Rev. Neurosci.*, 24:1193–1216. ♦

Optimality Theory in Linguistics

Kie Zuraw

Introduction

Prince and Smolensky (1993) introduced Optimality Theory (OT) as a framework for linguistic analysis. Kager (1999) gives an entry-level introduction to OT. McCarthy (2001) surveys advanced topics, and the Rutgers Optimality Archive (<http://ruccs.rutgers.edu/roa.html>) contains hundreds of OT papers. Within phonology, OT has largely supplanted rule-based frameworks. OT has also been applied to syntax and semantics, although not as widely; Legendre, Grimshaw, and Vikner (2001) provide an overview of current work in OT syntax.

Rule-based frameworks account for linguistic patterns through the sequential application of transformations to lexical entries. Variation between two pronunciations of the English plural suffix—[s] in *cats* but [z] in *dogs*—is explained by a rule that devoices the suffix after voiceless consonants (like [t]). The input *cat* + /z/, assembled from entries in the speaker's mental dictionary, is transformed by rule into the output *cat*[s]. In OT, the output is instead chosen through competition with other candidates: a constraint requiring adjacent consonants to match in voicing favors *cat*[s] over *cat*[z].

Generation of utterances in OT involves two functions, *Gen* and *Eval*. *Gen* takes an input and returns a (possibly infinite) set of output candidates. Some candidates might be identical to the input, others modified somewhat, others unrecognizable. *Eval* chooses the candidate that best satisfies a set of ranked constraints; this optimal candidate becomes the output.

The constraints of *Eval* are of two types. *Markedness* constraints enforce well-formedness of the output itself, prohibiting structures that are difficult to produce or comprehend, such as consonant clusters or phrases without overt heads. *Faithfulness* constraints enforce similarity between input and output, for example, requiring all input consonants to appear in the output, or all morphosyntactic features in the input to be overtly realized in the output. Markedness and faithfulness constraints can conflict, so the constraints' ranking, which differs from language to language, determines the outcome. One language might eliminate consonant clusters by deleting consonants, despite the resulting faithfulness violations; another might retain all input consonants, violating the markedness constraint.

In standard OT, constraints are strictly ranked and violable. *Strict ranking* means that a candidate violating a high-ranked constraint cannot redeem itself by satisfying lower-ranked constraints (constraints are not numerically weighted, and lower-ranked constraints cannot gang up on a higher-ranked constraint). *Violability* means that the optimal candidate need not satisfy all constraints. *Eval* can be viewed as choosing the subset of candidates that best satisfy the top-ranked constraint, then, of this subset, selecting the sub-subset that best satisfies the second-ranked constraint, and so on. Another way of describing *Eval* is that a candidate *i* is optimal if and only if, for any constraint that prefers another candidate *j* to *i*, there is a higher-ranked constraint that prefers *i* to *j*.

The *tableau* (a standard expositional device in OT) in Figure 1 illustrates output selection for the input /ilp/ in a hypothetical mini-language. Each of the four output candidates is flawed: *c*, the most faithful, has a consonant cluster, violating the markedness constraint *CC, as indicated by the asterisk at the intersection of *CC's column and *c*'s row. Candidate *b* has deleted a segment, and *a* has inserted a segment; these candidates violate the faithfulness constraints DON'TDELETE and DON'TINSERT, respectively (phonologists' MAX and DEP). Candidate *d* has inserted a segment without breaking up the consonant cluster, violating both DON'TINSERT and *CC.

*CC is the highest-ranked constraint (ranking is indicated by left-to-right ordering of the constraints' columns; we can also write

	/ilp/	*CC	DON'T DELETE	DON'T INSERT
<i>a</i>	[ilip]			*
<i>b</i>	[il]		*!	
<i>c</i>	[ilp]	*!		
<i>d</i>	[ilpi]	*!		*

Figure 1. Optimality Theory tableau.

*CC >> DON'TDELETE >> DON'TINSERT). *Eval* first eliminates *c* and *d* from the competition (an exclamation point represents elimination) because they alone violate *CC. The shading in the cells to the right represents the irrelevance of *c*'s and *d*'s performance on any lower-ranked constraints. *Eval* next eliminates *b*, because it violates DON'TDELETE. The remaining candidate, *a*, is optimal, as indicated by the pointing finger. In this language, an input string /ilp/ is pronounced [ilip]; in another language the constraint ranking, and thus the output, might be different. There are rankings that would choose *a* or *b* as the optimal candidate. Candidate *d*, however, is *harmonically bounded* by *a*, and by *c*: its violations are a proper superset of both *a*'s and *c*'s. Therefore, *d* cannot be the optimal candidate under any ranking of just these three constraints, although it could be optimal with a larger constraint set.

Wilson (2001) proposes an alternative formulation of *Eval* in which markedness constraints are “targeted”: they compare only candidates that are maximally perceptually similar and impose only pairwise preferences on candidates. For each constraint, starting with the highest ranked, *Eval* adds any new pairwise preferences that do not contradict those imposed by higher-ranked constraints, and constructs the transitive closure.

OT in Linguistic Theory

This section reviews why OT has been so widely adopted, and its advantages and disadvantages (see McCarthy, 2001).

OT was developed as a response to a “conceptual crisis at the center of phonological thought” (Prince and Smolensky, 1993, p. 1) concerning the role of output constraints. In a 1970 *Linguistic Inquiry* article, Charles Kisseberth identified a “conspiracy” in Yawelmani: rules of vowel insertion and deletion conspire to place every consonant adjacent to a vowel. Kisseberth proposed introducing constraints (such as *CCC, forbidding three-consonant clusters) to block or trigger rules, which could then be simplified and made more similar across languages. Output constraints were increasingly exploited in the literature, but many aspects of their use were unclear. How should a constraint be designated to block or trigger a rule? What if output constraints conflicted? How could nonabsolute preferences be expressed? For example, Yawelmani allows the sequences CiCC and CCiC, but underlying CCC is repaired to CiCC. Therefore, in addition to the constraint *CCC and the rule of *i*-insertion, there must be a constraint preferring CiCC over CCiC. But this second constraint is violable, because CCiC sequences do occur. OT addressed these problems by eliminating rules entirely in favor of constraints, and specifying how constraints interact.

One advantage of OT over rule-based theories is that it predicts the emergence of the unmarked (TETU): a markedness constraint that is frequently violated in a language may still affect outputs. The constraint favoring CiCC over CCiC in Yawelmani, for example, is not surface-true (CCiC sequences do occur, because high-ranking faithfulness constraints preserve them), but when *CCC forces a vowel to be inserted, CiCC is preferred over CCiC. A major contribution of OT has been to focus attention on TETU, of which many new cases have been found.

Another advantage of OT is its straightforward account of what McCarthy calls “homogeneity of target/heterogeneity of process.” A rule specifies the structure to which it applies (the target) and the operation to be performed on that structure (process). It has long been observed, however, that rules applying different processes to the same target tend to occur, both across languages and within the same language. A rule-based theory has no explanation for why a structure should be a recurring target. In OT, however, the explanation is straightforward: there is a markedness constraint against the target, but whether and how the target is repaired depends on interaction with other constraints. In Figure 1, for ex-

ample, permuting the constraint ranking yields three minilanguages: one that allows CC clusters, one that eliminates them by vowel insertion, and one that eliminates them by consonant deletion. The set of predicted languages that results from permuting the ranking of a group of constraints is its *factorial typology*. A proposed set of interacting constraints is considered viable only if its factorial typology matches the typology of observed languages—that is, it predicts all existent and no nonexistent patterns.

In some cases, OT's prediction of heterogeneity of process may be overly exuberant. For example, all else being equal, languages that resolve intervocalic CC clusters by deletion delete the first consonant, not the second. Wilson's targeted constraints close this gap and others in the factorial typology: with targeted constraints, deleting the second consonant cannot be optimal under any constraint ranking.

OT is at a disadvantage in dealing with opacity. In a rule-based framework, opacity occurs when a later rule either eliminates the structure that caused an earlier rule to apply (obscuring why the earlier rule applied) or creates a structure that would have caused an earlier rule to apply (obscuring why the earlier rule failed to apply). Standard OT, however, is unable to capture most opacity. Several additional proposals have therefore been made, including harmonic serialism, turbid output representations, output-output faithfulness, sympathy, targeted constraints, and constraint conjunction (see McCarthy, 2001, chap. 3, for a survey). The computability consequences of these proposals, in learning and/or generation, remain to be established.

Computability of OT

Generation

In rule-based frameworks, generation—mapping input to output, the speaker's task—is straightforward. Each rule identifies target structures in a representation, makes the required change, and passes the result to the next rule. In OT, generation presents a computational challenge, because the candidate set may be infinite (in phonology, it is always infinite, because insertions are unlimited). In that case, *Eval* cannot proceed in the obvious way, by first going through all candidates and totaling violations of the highest-ranked constraint, because that first step would never end.

Eisner (1997), building on earlier work by Mark Ellison, proposes a simple way of dealing with the infinite candidate set. At every point in his generation algorithm, the candidate set is represented as a finite-state automaton (FSA), rather than as a list. This is possible if the candidates and constraints are expressed in Eisner's Primitive Optimality Theory (OTP) formalism.

The winning candidate in OTP can be defined recursively. *Repns* is an FSA that accepts all syntactically well-formed OTP representations of input-output mappings. *Input* is an FSA that accepts mappings from the given input to any output. Intersecting *Repns* and *Input* produces an FSA, S_0 , that accepts well-formed mappings from the given input. S_0 is the initial candidate set.

Further, define an FSA C_i for each of the n constraints in the hierarchy, where C_i corresponds to the highest-ranked constraint. Each C_i accepts any mapping, but the arcs that a mapping traverses when it violates CONSTRAINT $_i$ are weighted. C_1 is intersected with S_0 to produce an FSA that accepts S_0 , but with the arcs corresponding to violations of CONSTRAINT1 weighted. Dijkstra's Best Paths algorithm, which finds the least-weighted path(s) through an FSA, is then applied to $C_1 \cap S_0$ to yield an FSA (S_1) that accepts the representations in S_0 that minimally violate CONSTRAINT1—i.e., the set of candidates left after CONSTRAINT1 has applied. Repeating this procedure for all n constraints, the winning candidate (or set of candidates, if there are not enough constraints to select a unique winner) is S_n .

Comprehension

Comprehension—the listener’s task—has been little addressed for standard OT, although Eisner (2000) proposes an algorithm for comprehension under “directional constraint evaluation.” A comprehension algorithm would yield, for a given output form, the (possibly infinite) set of inputs that would map to that output under the given grammar. The problem is not trivial: the input may contain a markedness violation not found in the output just in case the constraint ranking is such that the violation would have been repaired by a higher-ranking faithfulness constraint, and the result of the repair would be the observed output.

Learning

Learning—the child’s task—includes (at least) two subtasks: building a lexicon and determining the constraint ranking of the target language. If the constraint set is not universal, the learner must also determine what the constraints of her language are; see Boersma (1998) for a model of learning articulatory and acoustic constraints, and Albright and Hayes (1999) for an algorithm that learns morphophonological constraints.

Little work exists on the learnability of the lexicon. Prince and Smolensky (1993) propose “lexicon optimization”: where possible, learners construct lexical representations that are identical to the surface representations they hear. When the learner encounters alternations, such as the different pronunciations of the English plural suffix, she must construct a single lexical representation. Curtin (2001) presents evidence that children’s early lexical representations are phonetically detailed and do not strip out redundancies; this suggests that lexical consolidation of different pronunciations of the same morpheme occurs relatively late in learning, perhaps after most of the constraint ranking is established.

The problem of establishing a constraint ranking has been addressed more thoroughly. Tesar and Smolensky’s (2000) Constraint Demotion Algorithm and its variants rank a set of constraints given a set of outputs. The algorithm compares an observed output (presumed to come from a mature speaker) to any candidate erroneously rated as optimal under the learner’s current constraint ranking. In order to make the observed output optimal, for every constraint *B* that prefers the spurious output, some higher-ranked constraint *A* must prefer the observed output. If this is not already the case, the learner demotes *B* below *A*. The learner must know the input form in order to evaluate faithfulness constraints; in a more realistic model, some interleaving of input-learning and ranking-learning would be necessary. Variants of the algorithm accommodate the common proposal that the learner ranks markedness above faithfulness unless she encounters evidence to the contrary.

The Constraint Demotion Algorithm finds a ranking consistent with the learning data, if one exists. The algorithm has not been successfully generalized to learn variable grammars (discussed below), however, and is not robust to occasional errors in the learning data.

Probabilistic and Variable OT

Intraspeaker variation is common in language: a speaker may produce an utterance differently on different occasions. For example, American English speakers optionally produce [nt] as a nasalized flap (so that “winter” sounds similar to “winner”). The desire to capture variability in OT has led to proposals of variable constraint ranking.

Anttila (1997) proposes that a “stratified” constraint ranking is equivalent to all the linear rankings that are consistent with it, and the predicted frequency of a variant is the proportion of linear rank-

ings that generate it. Suppose a language has the stratified ranking $A \gg \{B, C, D\}$ (i.e., *B*, *C*, and *D* are freely ranked, but below *A*), and a candidate *a* is optimal only under $A \gg B \gg C \gg D$ and $A \gg B \gg D \gg C$. The stratified ranking collapses six linear rankings, two of which produce *a*, so *a* should be observed 33% of the time. In a corpus study of Finnish genitive plurals, Anttila found a good match between predicted and observed frequencies of variants. No learning algorithm has been proposed, however, for grammars with free rankings.

Boersma (1998) proposes stochastic constraint ranking—ranking that is neither absolutely fixed nor absolutely free, but probabilistic. Each constraint in an individual’s grammar has a ranking value in arbitrary units. For every utterance, the speaker generates effective values for each constraint by randomly perturbing each ranking value slightly. Each constraint is thus associated with a probability density function centered on its ranking value. Figure 2 illustrates a minigrammar in which constraint C_1 is nearly always top ranked, and C_4 is nearly always bottom ranked, but C_2 and C_3 are variably ranked, with a preference for the ranking $C_2 \gg C_3$.

Stochastic constraint ranking captures fine-grained frequencies. In an Anttilian grammar with three variably ranked constraints, a variant can occur only 0%, 33%, 67%, or 100% of the time, depending on which rankings produce that variant. In a Boersmian grammar with the same three constraints, the variant can occur at any frequency, depending on the ranking values of the constraints. Boersma and Hayes (2001) suggest some cases of very infrequent variants that would be difficult to capture in an Anttilian model, although firm data remain to be gathered.

An advantage of Boersma’s model is its learnability. Boersma’s Gradual Learning Algorithm can learn stochastic grammars from variable learning data (if the learning data are not variable, the ranking values learned are so far apart that the ranking is effectively fixed). In each learning trial of the algorithm, the learner compares its production to an adult target form. If there is a mismatch, the learner increments the ranking values of all constraints that prefer the learner’s incorrect form, and decrements the ranking values of all constraints that prefer the adult form. The algorithm is robust to errors in the learning data; if an erroneous learning datum nudges a constraint in the wrong direction, subsequent data push it back. The Gradual Learning Algorithm can also model the course of acquisition. Curtin (2001) shows how, for the acquisition of prosody, the Gradual Learning Algorithm successfully models variability in children’s productions, stage-like progression, and the order in which markedness constraints are demoted.

The Gradual Learning Algorithm can learn rates of variation because conflicting variants exert opposite influences on ranking values. The more frequent variant occurs in more learning trials, so the relevant constraints’ ranking values are separated to the degree that the variants differ in frequency. The algorithm is also able to learn rates of lexical variation (situations in which each word’s pronunciation is stable, but certain words display a phonological phenomenon and others do not), as shown in Zuraw (2000). In Zuraw’s model, the resulting grammar has high-ranking faithfulness constraints that ensure the correct pronunciation of existing words, with lexical variation encoded in low-ranked constraints that come into play in the production and comprehension of new words.

Discussion

OT was partly inspired by neural networks. The ideas of optimization, parallel evaluation, competition, and soft, conflicting constraints are familiar. Prince and Smolensky (1997) discuss the implementation of OT in a neural network. Constraints are implemented as connection weights, and the network implements a Lya-

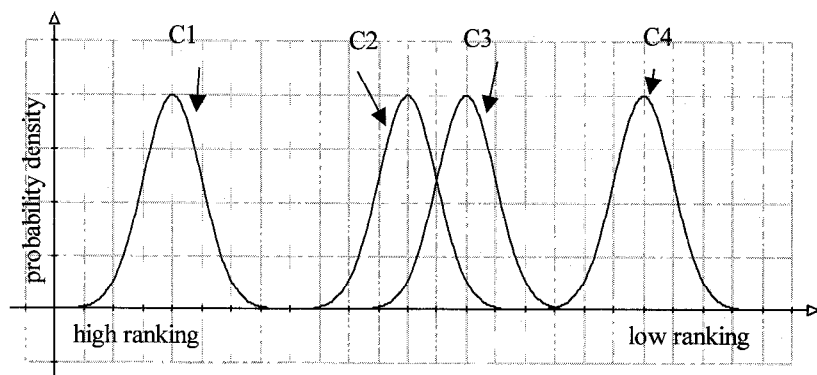


Figure 2. Stochastic constraint ranking.

punov function that maximizes “harmony” ($\sum_{ij} a_i w_{ij} a_j$; the sum, for all pairs i, j of neurons, of the product of the neurons’ activations and their connection weight). Hierarchically structured representations (e.g., consonants and vowels grouped into syllables) can be represented as matrices of neurons, where each matrix is the tensor product of a vector for a linguistic unit and a vector for its position in the hierarchy. Implementing strict domination (rather than the usual numerical weighting) of constraints remains unsolved, however, so translation between OT grammars and neural networks is not in general possible.

Road Map: Linguistics and Speech Processing

Related Reading: Speech Production

References

- Albright, A., and Hayes, B., 1999, An automated learner for phonology and morphology, UCLA Working Paper in Linguistics, University of California, Los Angeles.
- Anttila, A., 1997, Deriving variation from grammar: A study of Finnish genitives, in *Variation, Change, and Phonological Theory* (F. Hinskens, R. van Hout, and L. Wetzels, Eds.), Amsterdam: John Benjamins, pp. 35–68.
- Boersma, P., 1998, *Functional Phonology*, The Hague: Holland Academic Graphics.
- Boersma, P., and Hayes, B., 2001, Empirical tests of the Gradual Learning Algorithm, *Ling. Inquiry*, 32:45–86. ♦
- Curtin, S., 2001, Enriched lexical representations and constraint organization in a developing system, Ph.D. diss., University of Southern California.
- Eisner, J., 1997, Efficient generation in primitive Optimality Theory, in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Association for Computational Linguistics*, San Francisco: Morgan Kaufmann, pp. 313–320.
- Eisner, J., 2000, Directional constraint evaluation in Optimality Theory, in *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, San Francisco: Morgan Kaufmann, pp. 257–263.
- Kager, R., 1999, *Optimality Theory*, Cambridge, Engl.: Cambridge University Press. ♦
- Legendre, G., Grimshaw, J., and Vikner, S., Eds., 2001, *Optimality-Theoretic Syntax*, Cambridge, MA: MIT Press. ♦
- McCarthy, J., 2001, *A Thematic Guide to Optimality Theory*, Cambridge, Engl.: Cambridge University Press.
- Prince, A., and Smolensky, P., 1993, *Optimality Theory: Constraint Interaction in Generative Grammar*, New Brunswick, NJ: Rutgers Center for Cognitive Science Technical Report TR-2. (See Kager, 1999, for textbook treatment.)
- Prince, A., and Smolensky, P., 1997, Optimality: From neural networks to universal grammar, *Science*, 275:1604–1610. ♦
- Tesar, B., and Smolensky, P., 2000, *Learnability in Optimality Theory*, Cambridge, MA: MIT Press.
- Wilson, C., 2001, Consonant cluster neutralisation and targeted constraints, *Phonology*, 18:147–197.
- Zuraw, K., 2000, Patterned exceptions in phonology, Ph.D. diss., University of California, Los Angeles.

Optimization, Neural

Carsten Peterson and Bo Söderberg

Introduction

Many combinatorial optimization problems require a more or less exhaustive search to achieve exact solutions, with the computational effort growing exponentially or worse with system size. Hence, for large problems, the quest for an exact solution has to be abandoned. Instead, various kinds of heuristic methods have been developed that yield reasonably good approximate solutions.

Artificial neural network (ANN) methods in general fall within this category. Particularly interesting in the context of optimization are *recurrent network* methods based on *deterministic annealing*. In contrast to most other methods, these are not based on a direct exploration of the given discrete state space; instead, they utilize

an interpolating continuous (analogue) space, allowing for shortcuts to good solutions. Key concepts here are the *mean-field* (MF) approximation (Hopfield and Tank, 1985; Peterson and Söderberg, 1989) and *annealing*.

Although early versions were confined to problems encodable with a quadratic energy in terms of a set of binary variables, in the past decade the method has been extended to deal with more general problem types in terms of both variable types and energy functions, and has evolved to a general-purpose heuristic for combinatorial optimization. An appealing feature is that the basic MF dynamics is directly implementable in VLSI (see ANALOG VLSI IMPLEMENTATIONS OF NEURAL NETWORKS), facilitating hardware implementations.

Recurrent Networks

Recurrent networks appear in the context of associative memories (Hopfield, 1982) and difficult optimization problems (Hopfield and Tank, 1985; Peterson and Söderberg, 1989). Such networks resemble statistical models of magnetic systems ("spin glasses"), with an atomic spin state (up or down) seen as analogous to the "firing" state of a neuron (on or off). This similarity has been the source of much inspiration for neural network studies.

The archetype of a recurrent network is the Hopfield model (Hopfield, 1982), which is based on an energy function of the form

$$E(s) = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j \quad (1)$$

in terms of binary variables (or Ising spins, as used in magnetic models), $s_i = \pm 1$ (in some contexts, equivalent 0,1 spins are preferred), with symmetric weights w_{ij} . Owing to the identity $s_i^2 = 1$, diagonal components w_{ii} are redundant and can be assumed to vanish.

With an appropriate choice of weights determined by a set of stored patterns, the latter appear as local minima, satisfying

$$s_i = \text{sgn} \left(\sum_j w_{ij} s_j \right) \quad (2)$$

With a simple asynchronous dynamics based on iterating Equation 2, this system turns into a recurrent ANN, having the local minima as stationary points. In effect, this model serves as an associative memory (see COMPUTING WITH ATTRACTORS).

Update Modes

Note the importance of *asynchronous* update (in random order or sequentially), in which case $E(s)$ is a Lyapunov function and cannot increase. This guarantees the convergence toward a fixed point defining a local energy minimum.

Attempting to iterate Equation 2 in *synchrony* would not necessarily yield convergence to a local minimum. The system could wind up in a two-cycle instead, with a subset of the spins flipping signs on every update.

This behavior can be understood by viewing two consecutive synchronous updates of the N spins as a single sequential update of a system of $2N$ spins, $\{x_i, y_i\}$, where first the x are updated based on y (as $x_i = \text{sgn} \sum_j w_{ij} y_j$), and then the y are updated based on the new x . The corresponding energy $\hat{E}(x, y) = -\sum_{ij} x_i w_{ij} y_j$ is a Lyapunov function, and the extended system must converge to a fixed point (x^*, y^*) . If x^* and y^* are equal, they define a fixed point s^* (a local energy minimum) of the original system; otherwise a two-cycle results with s alternating between x^* and y^* . If x^* and y^* are maximally different ($x^* = -y^*$), they define two equivalent local *maxima* of E . Other, mixed cases can be seen as saddle points.

Consider also the related problem of minimizing $-E$ (i.e., maximizing E), obtained by flipping all signs in w . In terms of x, y , the corresponding update equations differ from the original only by replacing, say, y by $-y$. Thus, there is a one-to-one correspondence between sequences of states for s obtained for $-E$ and for E : one is obtained from the other by flipping the signs of every second state. This shows that with synchronous update, the system cannot really tell the difference between minimizing and maximizing E .

The undesirable behavior of synchronous update can be avoided by introducing stabilizing *self-couplings* in the form of positive diagonal elements large enough to make w a positive-definite matrix (and $E(s)$ a concave function); this, however, has the negative side effect of adding a multitude of stationary states that are not local minima.

Below the sequential update mode without self-couplings will be assumed where not otherwise stated.

Optimization with Recurrent Networks

Many types of optimization problems can be encoded in terms of a Hopfield model, with the energy function adapted to a specific problem by a dedicated choice of weights, such that global minima of $E(s)$ correspond to solutions. For simple problems, the recurrent network dynamics of iterating Equation 2 can be used to find a solution. For more difficult problems, however, the system will most likely get trapped in a nonoptimal local minimum close to the starting point, which is not desired. A more refined approach is needed to reach the global minimum, or at least a low-lying local minimum.

Stochastic Methods

A possible strategy is to employ a stochastic algorithm that allows for uphill moves, such as simulated annealing (SA) (see SIMULATED ANNEALING AND BOLTZMANN MACHINES). In this approach, a stochastic neighborhood search method is used in an attempt to generate a sequence of configurations distributed according to a Boltzmann distribution, $P(s) \propto e^{-E(s)/T}$, where T is an artificial temperature representing the noise level of the system, which is slowly decreased (annealing). With a very slow annealing rate, the system can avoid getting stuck in a local minimum, and produce a global minimum as $T \rightarrow 0$. Such a procedure can be very CPU-consuming, however.

The Mean-Field Equations

An alternative is given by MF annealing, where the stochastic SA method is approximated by a deterministic dynamics based on the MF approximation, defined as follows for a system of Ising spins.

The true Boltzmann distribution $P(s)$ is approximated by the direct product of single-spin distributions, $\prod_i p_i(s_i)$. Such a factorized distribution is characterized by the absence of correlations between the spins and is completely determined by the single-spin averages $v_i \equiv \langle s_i \rangle = p_i(1) - p_i(-1) \in [-1, 1]$. The parameters v_i are variationally determined so as to minimize the *free energy*,

$$F(v) = E(v) - TS(v) \quad (3)$$

where $E(v) \equiv \langle E(s) \rangle = -(\frac{1}{2}) \sum_{ij} w_{ij} v_i v_j$ is the average energy, while $S(v) \equiv -(\frac{1}{2}) \sum_i [(1 + v_i) \log(1 + v_i) + (1 - v_i) \log(1 - v_i)]$ is the entropy associated with the approximating distribution. Minimization of F with respect to v_i directly yields the *MF equations*,

$$v_i = \tanh(u_i/T), \text{ with} \quad (4)$$

$$u_i \equiv -\frac{\partial E(v)}{\partial v_i} \equiv \sum_j w_{ij} v_j \quad (5)$$

The analog *MF variables* v_i take values in the interval $[-1, 1]$, interpolating between the discrete spin states ± 1 , which is natural since they approximate the thermal spin averages $\langle s_i \rangle_T$.

Analog Network

The MF equations (Equation 4) can be solved by asynchronous iteration, analogously to the discrete Equation 2. The only difference is the replacement of the sharp step function $\text{sgn}(u_i)$ by a smooth sigmoid $\tanh(u_i/T)$, with an adjustable parameter $1/T$ controlling the gain: high T corresponds to very smooth sigmoids, while in the low- T limit the stepfunction of Equation 2 is recovered.

Most of the discussion on update modes in the context of Equation 2 also applies to the MF dynamics; thus, with asynchronous updating, the free energy $F(v)$ of Equation 3 defines a Lyapunov function guaranteeing the convergence to a fixed point defined by a local minimum of F .

Mean-Field Annealing

In MF (or deterministic) annealing, the fixed- T MF dynamics is slowly modified by lowering an initially high T , using, for example, a geometric annealing schedule.

For the quadratic Hopfield energy, the dynamics then will exhibit a behavior with two phases. At large temperatures, the system relaxes to a trivial fixed point v^o , with $v_i^o = 0$. As the temperature sinks below a critical value T_c , v^o becomes unstable and nontrivial fixed points emerge; as $T \rightarrow 0$ these are pushed toward discrete (± 1) values, representing a specific decision made as to the solution of the problem in question.

The position of the bifurcation point T_c can be determined by linearizing Equation 4 around v^o , that is, replacing the sigmoid function (\tanh) by its argument. With sequential updating without self-couplings, this yields a smooth tangent bifurcation at a temperature given by the largest positive eigenvalue of w (Peterson and Söderberg, 1989) (see also DYNAMICS AND BIFURCATION IN NEURAL NETS).

For an energy without the exact symmetry under $v \rightarrow -v$, as results, for example, from adding a linear energy term, a distinct bifurcation might be absent; then the high- T fixed point is only approximately zero, with the MF variables evolving continuously from smaller to larger values over a finite T interval. This is not a problem: a suitable initial T can still be estimated. Alternatively, an auxiliary spin variable can be introduced and multiplied by the linear term to restore symmetry.

Deterministic annealing yields a more efficient method for finding low-lying energy minima than setting $T = 0$ from the start (i.e., iterating Equation 2). Tracking a local minimum as T is lowered can guide the system to better low- T minima (see STATISTICAL MECHANICS OF NEURAL NETWORKS).

The Graph Bisection Problem

As an example application illustrating the abstract discussions heretofore presented, we will use graph bisection (GB). A graph with an even number N of nodes is to be divided into two halves of $N/2$ nodes each, such that the cut-size (the number of connections between the halves) is minimal (Figure 1A). The encoding is particularly transparent here because of the binary nature of the problem: with each node i a binary spin s_i is associated, to be assigned a value ± 1 representing whether the node will wind up in the left or right partition of Figure 1A. The graph is given in terms of a symmetric connection matrix J , such that an element J_{ij} equals 1 if vertices i, j are connected, and 0 if not (or if $i = j$). With this notation, the product $J_{ij}s_i s_j$ is non-zero only for a connected pair of nodes i, j , yielding 1 if they are put in the same partition and -1 if not. Thus, the cut-size is proportional to $-(1/2)\sum_{ij} J_{ij}s_i s_j$ plus an unimportant constant.

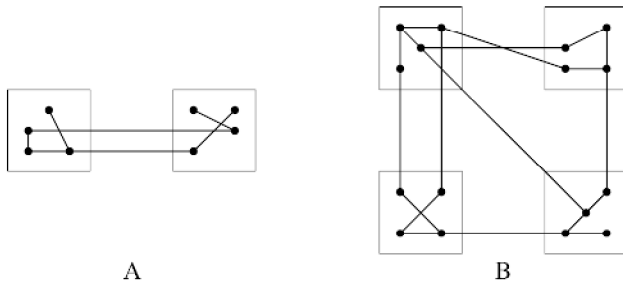


Figure 1. A, A graph bisection problem. B, A $K = 4$ graph partition problem.

In addition, one needs to take into account the global constraint of equal partition of the nodes, requiring $\sum_i s_i = 0$. This can be done by adding to the energy function a term that penalizes an illegitimate partition. A term proportional to $(\sum_i s_i)^2$ will do the trick.

Discarding the constant diagonal part $\sum_i s_i^2 = N$, we obtain a Hopfield energy function, Equation 1, with $w_{ij} = J_{ij} - \alpha(1 - \delta_{ij})$:

$$E = -\frac{1}{2} \sum_{ij} J_{ij} s_i s_j + \frac{\alpha}{2} \left(\left(\sum_i s_i \right)^2 - \sum_i s_i^2 \right) \quad (6)$$

where the constraint coefficient α sets the relative strength of the penalty term.

Equation 6 has a structure common in combinatorial optimization problems: $E = \text{Cost} + \text{Global constraint}$. The origin of the difficulty inherent in this kind of problem is very transparent here: the conflict associated with minimizing the two competing terms makes the system frustrated, which often leads to the appearance of many local minima.

For large random GB problems, MF annealing yields a distinctively better performance than simple iteration of Equation 2.

Recurrent Potts Networks

For GB and many other optimization problems, an encoding in terms of binary elementary variables is natural. However, there are many problems where the natural elementary decisions are of the type one-of- K with $K > 2$.

Early attempts to approach such problems with recurrent network methods were based on *neuron multiplexing* (Hopfield and Tank, 1985), where for each elementary K -fold decision, a set of K binary 0/1 neurons was used, with a *syntax* constraint requiring that precisely one of them be on (i.e., equal to 1) implemented in a soft way with a penalty term. In the original work on the traveling salesman problem, as well as in subsequent investigations on the graph partition problem (Peterson and Söderberg, 1989), this approach did not yield high-quality solutions in a parameter-robust way.

A more efficient encoding is based on K -state *Potts spins* with the syntax constraint built in. This confines the dynamics to the relevant parts of solution space and leads to a drastically improved performance.

MF Annealing with Potts Spins

A K -state Potts spin is a variable that has K possible values (states). For our purposes, the best representation is in terms of a K -dimensional vector $\mathbf{s} = (s_1, s_2, \dots, s_K)$, with the a th state given by the a th principal unit vector, defined by $s_a = 1$, $s_b = 0$ for $b \neq a$. These vectors point to the corners of a regular K -simplex (see Figure 2 for the case of $K = 3$). They are all normalized and mutually orthogonal, and fulfill in addition the syntax constraint $\sum_a s_a = 1$.

The MF equations for a system of Potts spins \mathbf{s}_i with a given energy function $E(\mathbf{s})$ in multilinear form ($\partial^2 E / \partial s_{ia} \partial s_{ib} = 0$) are derived in analogy to the Ising case: Approximate the Boltzmann distribution with a factorized Ansatz, $P(\mathbf{s}) = \prod_i p_i(\mathbf{s}_i)$, parameterized by the single-spin averages $\mathbf{v}_i \equiv \langle \mathbf{s}_i \rangle$. These are determined so as to minimize an associated free energy,

$$F(\mathbf{v}) = \langle E(\mathbf{s}) \rangle - TS(\mathbf{v}) = E(\mathbf{v}) - TS(\mathbf{v}) \quad (7)$$

where the last equality follows from multilinearity; the entropy S is given by $-\sum_{ia} v_{ia} \log(v_{ia})$. A local minimum of F satisfies the MF equations

$$v_{ia} = \frac{e^{u_{ia}/T}}{\sum_b e^{u_{ib}/T}} \quad (8)$$

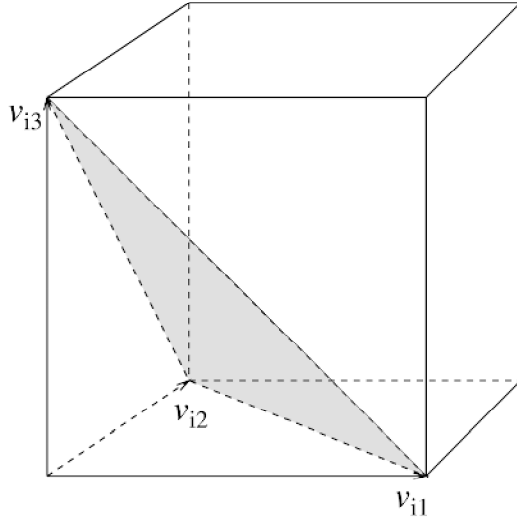


Figure 2. The cubic solution space corresponding to the neuron multiplexing encoding for $K = 3$, interpolating between the eight allowed spin states at the corners of the cube. With a Potts encoding, the solution space is restricted to the shaded triangle, interpolating between the three allowed Potts spin states at its corners.

with

$$v_{ia} \equiv -\frac{\partial E(\mathbf{v})}{\partial v_{ia}} \quad (9)$$

These result in *Potts MF neurons* v_i , approximating the thermal average of s_i , and satisfying $v_{ia} \geq 0$, $\sum_a v_{ia} = 1$ (for $K = 3$ the shaded region in Figure 2). A component v_{ia} represents a probability for the i th spin to be in state a . For $K = 2$ one recovers the formalism of the Ising case (with $v_i = v_{i1} - v_{i2} \in [-1, 1]$). As $T \rightarrow 0$, each MF neuron v_i is forced to approach a sharp spin state, defined by the index of the largest component of \mathbf{u}_i in Equation 8 (see also WINNER-TAKE-ALL NETWORKS).

Asynchronous iteration of the Potts MF equations in combination with annealing yields a deterministic annealing approach for Potts systems. As for an Ising system, a suitable initial temperature can be obtained, for example by means of a linear stability analysis.

The Graph Partition Problem

An illustration is given by K -fold *graph partition* (GP): The N nodes of a graph, defined by a symmetric connection matrix $J_{ij} = 0, 1$, $i \neq j = 1, \dots, N$, are to be grouped in K subsets of N/K nodes each, with a minimal cut-size (i.e., the number of connections between distinct subsets; see Figure 1B).

GP is naturally encoded with Potts spins, as follows. With each node $i = 1, \dots, N$, a K -state Potts spin, $\mathbf{s}_i = (s_{i1}, \dots, s_{iK})$, is associated, where a single nonvanishing component $s_{ia} = 1$ is to be chosen to represent the choice of subset a for node i . A suitable quadratic energy function (cf. Equation 6) is

$$E(\mathbf{s}) = -\frac{1}{2} \sum_{i,j=1}^N J_{ij} \mathbf{s}_i \cdot \mathbf{s}_j + \frac{\alpha}{2} \left(\left(\sum_{i=1}^N \mathbf{s}_i \right)^2 - \sum_{i=1}^N \mathbf{s}_i^2 \right) \quad (10)$$

where the first term is a cost term (cut-size), while the second is a penalty term with a minimum when the nodes are equally partitioned into the K subsets. Note that the diagonal contributions are subtracted in the second term to secure multilinearity.

Writing E as $-(1/2) \sum_{ij} w_{ij} \mathbf{s}_i \cdot \mathbf{s}_j$, we have for the input $\mathbf{u}_i = \sum_j w_{ij} \mathbf{v}_j$, in analogy to Equation 5.

Refinements and Generalizations

In this section, we discuss modifications and extensions of MF annealing, as well as complications that may arise in optimization applications and that require special care in one way or another.

Continuous-Time Methods

An alternative method for solving, say, the Ising MF equations (Equations 4 and 5) is to use a continuous-time formalism, based on $\dot{u}_i = -u_i + \sum_j w_{ij} v_j$, with $v_j \equiv \tanh(u_j/T)$. Indeed, such a formulation was used in the original work of Hopfield and Tank (1985). It is easily generalized to the Potts case. In both cases such a dynamics can also be directly implemented in VLSI.

A continuous-time formalism facilitates an alternative method for implementing a global constraint, such as $\sum_i s_i = 0$ in GB, with a linear term $\lambda \sum_i s_i$, where λ is a *Lagrange multiplier* to be dynamically adjusted such that a balanced stationary state results. Such methods are discussed in Platt and Barr (1988).

A naive discretization of the continuous-time system with a unit time step results in synchronous updating of Equation 4, with problems like an absent Lyapunov function and two-cycle behavior. However, with a small enough time step $\varepsilon \ll 1$, a stabilized discrete-time approach results, corresponding to synchronous updating of the inputs u according to $u_i = (1 - \varepsilon)u_i + \varepsilon \sum_j w_{ij} v_j$, defining a parallelizable alternative to the standard asynchronous discrete-time dynamics. For a sequential software implementation, however, the latter is far more efficient.

Nonquadratic Energy Functions

With a quadratic Potts energy $E(\mathbf{s})$, self-couplings can be avoided and multilinearity secured by removing all diagonal terms, $s_{ia}s_{ib} \rightarrow \delta_{ab}s_{ia}$; such a procedure can be generalized to any polynomial E . Although in principle any energy function of a finite number N of spins can be rewritten as a polynomial of at most degree N , this may be difficult in practice for large N with an energy in nonquadratic form.

An efficient and general alternative method for disarming self-couplings in a Potts system with a given generic energy function E is to simply replace the derivative in Equation 9 for the input by a difference:

$$u_{ia} = -\frac{1}{T} (E(\mathbf{v})|_{v_i=\mathbf{e}_a} - E(\mathbf{v})|_{v_i=\mathbf{0}}) \quad (11)$$

where \mathbf{e}_a is a unit vector in the a -direction. Whenever E is multilinear, Equations 9 and 11 are equivalent.

Inequality Constraints

In the optimization problems mentioned above, the constraints considered were all of the *equality* type, $g(s) = 0$, that could be implemented with quadratic penalty terms $\propto g(s)^2$. However, in many optimization problems, especially those of the resource allocation type, one has to deal with *inequalities*. An inequality constraint, $g(s) \leq 0$, can be implemented with a penalty term proportional to, e.g.,

$$\Phi(g) = g \Theta(g) \quad (12)$$

with Θ the Heaviside step function: $\Theta(x) = 1$ if $x > 0$ and 0 otherwise. Of course, such a nonpolynomial term in the energy requires the use of Equation 11.

Inequality constraints appear, for example, in the *knapsack problem*, where one has a set of N items i , with associated utilities c_i and loads a_{ki} . The goal is to fill a “knapsack” with a subset of the items such that their total utility is maximized, subject to a set of

M load constraints. In terms of binary spins $s_i \in \{1, 0\}$, representing whether or not item i goes into the knapsack, the total utility can be expressed as

$$U = \sum_{i=1}^N c_i s_i \quad (13)$$

and the load constraints as

$$\sum_{i=1}^N a_{ki} s_i \leq b_k, \quad k = 1, \dots, M \quad (14)$$

where $b_k > 0$ define load capacities, which can be seen as representing distinct limiting aspects of the knapsack (its height, width, etc.).

In Ohlsson, Peterson, and Söderberg (1993), a set of difficult random knapsack problems were successfully approached with an MF annealing method based on the energy function

$$E = -\sum_{i=1}^N c_i s_i + \alpha \sum_{k=1}^M \Phi\left(\sum_{i=1}^N a_{ki} s_i - b_k\right) \quad (15)$$

Scheduling and Constraint Satisfaction

Scheduling problems have a natural formulation in terms of Potts spins and can be approached with Potts MF annealing. A pure scheduling problem can have the following simple structure: For a given set of events, a time slot and a location are to be chosen, each from a set of allowed possibilities, such that no clashes occur. Such a problem consists entirely in fulfilling a set of basic no-clash constraints, $g = 0$, each of which can be handled with a non-negative penalty term, e.g., $\propto g^2$, that will vanish for a legal schedule.

In realistic scheduling applications, there often exist additional preferences within the set of legal schedules that lead to the appearance also of *cost terms*. A set of real-world scheduling problems was successfully dealt with in Gislén, Peterson, and Söderberg (1992), using a straightforward MF Potts formalism.

Pure scheduling is a special type of *constraint satisfaction* problem (CSP) where the entire object is to satisfy a set of constraints. Such problems have been much studied in computer science. INN is a modified MF annealing approach dedicated to CSP, where a particular kind of nonpolynomial penalty term is used, based on an information-theoretic analysis. In Jönsson and Söderberg (2001), INN was applied to a set of difficult K -SAT problems and shown to outperform a conventional MF annealing approach based on polynomial penalty terms.

For constrained optimization, a *hybrid approach* might be advantageous, using a conventional polynomial energy term for the cost part and nonpolynomial INN-type penalty terms for the constraints.

Routing Problems

Many network routing problems can be conveniently handled using a Potts MF approach. The basic idea can be illustrated with a simple shortest-path problem: Given a network of N nodes connected by arcs of given lengths, find the shortest path between nodes a and b , i.e., the shortest sequence of arcs leading from a to b .

This problem can be solved in polynomial time using, e.g., the Bellman-Ford (BF) algorithm (Bellman, 1958), where every node i estimates its distance D_{ib} to b , minimized with respect to the choice of a continuation node j among its neighbors (nodes directly connected to i via an arc of length d_{ij}):

$$D_{ib} = \min_j (d_{ij} + D_{jb}), \quad i \neq b \quad (16)$$

while $D_{bb} = 0$. Iteration of Equation 16 gives convergence in less than N steps, and D_{ab} can be read off.

Also, more complex routing problems, such as ones with several competing routing requests, can be formulated in terms of optimal neighbor choices that can be encoded by a set of Potts spins. The resulting system can then be handled with a Potts MF annealing algorithm. An appealing feature of such an approach is the *locality* inherited from BF: all information required for the neighbor choice is local to the node and its neighbors.

In Häkkinen et al. (1998) a set of complex routing problems in finite-capacity networks was approached in this manner, aided by a *propagator* formalism for monitoring global topological aspects of the fuzzy MF routes.

Mutual Assignment Problems

In certain classes of problems, one seeks an optimal one-to-one assignment between the elements in two sets of equal size N . Such an assignment can be encoded with a doubly stochastic 0/1-matrix s ,

$$s_{ij} \in \{0, 1\}, \quad i, j = 1, \dots, N \quad (17)$$

$$\sum_j s_{ij} = 1 \quad (18)$$

$$\sum_i s_{ij} = 1 \quad (19)$$

such that $s_{ij} = 1$ represents the mutual assignment of element i in the first set with element j in the other.

An example is the traveling salesman problem, where the goal is to minimize the total length of a closed tour connecting a set of N cities with given pairwise distances. This can be seen as finding an optimal mutual assignment between cities and positions in the tour.

In an early approach Hopfield and Tank (1985) to the traveling salesman problem, each component of s was considered an independent binary 0/1 spin, and the row and column sum constraints on s were softly implemented by means of penalty terms. In a refined MF annealing approach (Peterson and Söderberg, 1989), each row of s was taken as a separate Potts spin, while penalty terms were used for the column sum constraints (*row-Potts*; the opposite, *column-Potts*, is of course also possible), yielding a noticeable increase in performance.

Ideally, however, one would prefer a dedicated MF method for mutual assignments. Such an approach can indeed be devised, by using a single Potts spin with $N!$ components, one for each possible assignment. A problem with this approach is the inevitably non-polynomial time consumption for large N , which makes it infeasible for large problems.

For large mutual assignment problems, the best recurrent network method around appears to be *Softassign* (Yuille and Kossowski, 1994; Rangarajan, Gold, and Mjolsness, 1996), where both row and column sum constraints are formally implemented in an exact manner by means of Lagrange multipliers, yielding a formalism that can be seen as a synthesis between the row- and column-Potts MF approaches, although not strictly derived from a proper MF formalism. Softassign requires synchronous updating, in contrast to the row-Potts approach, where one row at a time can be updated; this yields instability problems that have to be remedied, such as with positive self-couplings. At low temperatures, it also suffers an inevitable slowing down of the row and column normalization procedure.

Hybrid Approaches

A large class of optimization problems can be viewed as *parametric assignment* problems, containing elements of both discrete assign-

ment and parametric fitting to given data, e.g., by using templates with a known structure. Then the assignment part can be encoded in terms of Potts neurons while the template part can be formulated in terms of a set of continuous adjustable parameters.

Also, certain pure assignment problems with a well-defined geometric structure can be cast in this form; a nice example is the *elastic net* algorithm (Durbin and Willshaw, 1987; Simic, 1990; Yuille, 1990) for planar traveling salesman problem, where a closed curve is allowed to move and deform elastically in the plane, with each city choosing a nearby point on the curve by means of an analog Potts MF neuron. As $T \rightarrow 0$, each chosen point is attracted to the respective city, while the remaining points on the curve are adjusting to form straight segments in between.

Discussion

For a large class of combinatorial optimization problems, a straightforward MF annealing approach can be used, based on an encoding in terms of Ising or Potts spins, with the following basic steps:

- Map the problem onto a recurrent network by a suitable encoding of solution space (in terms of a set of binary or Potts spins) and an appropriate choice of energy function, and derive the associated MF equations.
- Compute a suitable starting temperature (e.g., by means of a linear stability analysis of the asynchronous MF dynamics).
- Solve the MF equations iteratively while slowly lowering T .
- When the system has settled, the solutions are checked with respect to constraint satisfaction, if applicable. If needed, one may perform a simple corrective postprocessing or rerun the system (possibly with modified constraint coefficients).

This very general approach has been numerically explored for many different problem types, resulting in the following general picture. The MF annealing method, without excessive fine-tuning, consistently performs roughly in parity with dedicated problem-specific heuristics, designed to perform well for a particular problem class. Convergence is consistently achieved after a modest number (typically 50–100) of iterations, independently of problem size.

Modified variants of this method have been defined for specific problem types, such as INN for pure constraint satisfaction prob-

lems and Softassign for mutual assignment problems. For parametric assignment problems and for certain low-dimensional geometrical assignment problems such as the planar traveling salesman problem, hybrid methods can be used in which Potts MF neurons are combined with conventional analog parameters.

Road Map: Dynamic Systems

Background: Computing with Attractors

Related Reading: Cortical Hebbian Modules; Energy Functionals for Neural Networks; Phase-Plane Analysis of Neural Nets; Statistical Mechanics of Neural Networks

References

- Bellman, R., 1958, On a routing problem, *Q. Appl. Math.*, 16:87–90.
- Durbin, R., and Willshaw, D., 1987, An analog approach to the traveling salesman problem using an elastic net method, *Nature*, 326:689–691. ♦
- Gislén, L., Peterson, C., and Söderberg, B., 1992, Complex scheduling with Potts neural networks, *Neural Computat.*, 4:805–831.
- Häkkinen, J., Lagerholm, M., Peterson, C., and Söderberg, B., 1998, A Potts neuron approach to communication routing, *Neural Computat.*, 10:1587–1599.
- Hopfield, J. J., 1982, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, 79:2554–2558.
- Hopfield, J. J., and Tank, D. W., 1985, Neural computation of decisions in optimization problems, *Biol. Cybern.*, 52:141–152.
- Jönsson, H., and Söderberg, B., 2001, An information-based neural approach to constraint satisfaction, *Neural Computat.*, 13:1827–1838. ♦
- Ohlsson, M., Peterson, C., and Söderberg, B., 1993, Neural networks for optimization problems with inequality constraints: The knapsack problem, *Neural Computat.*, 5:331–339.
- Platt, J. C., and Barr, A. H., 1988, Constrained differential optimization, in *Neural Information Processing Systems* (Anderson, D. Z., ed.), New York: AIP, p. 55.
- Peterson, C., and Söderberg, B., 1989, A new method for mapping optimization problems onto neural networks, *Int. J. Neural Syst.*, 1:3–22.
- Rangarajan, A., Gold, S., and Mjolsness, E., 1996, A novel optimizing network architecture with applications, *Neural Computat.*, 8:1041–1060.
- Simic, P., 1990, Statistical mechanics as the underlying theory of “elastic” and “neural” optimizations, *Network*, 1:89–103. ♦
- Yuille, A. L., 1990, Generalized deformable models, statistical physics, and matching problems, *Neural Computat.*, 2:1–24.
- Yuille, A. L., and Kosowski, J. J., 1994, Statistical physics algorithms that converge, *Neural Computat.*, 6:341–356.

Optimization Principles in Motor Control

Tamar Flash, Neville Hogan, and Magnus J. E. Richardson

Introduction

Optimization theory has become an important research tool in our attempts to discover organizing principles that guide the generation of goal-directed motor behavior. It provides a convenient way to formulate a coarse-grained model of the underlying neural computation, without requiring specific details of the way those computations are carried out. Generally speaking, this application of optimization theory consists of defining an objective function that quantifies what is to be regarded as optimum (i.e., best) performance and then applying the tools of variational calculus to identify the specific behavior that achieves that optimum. This forces us to make explicit, quantitative hypotheses about the goals of motor actions and allows us to articulate how those goals relate to observable behavior. Not all motor behaviors are necessarily optimal,

but attempts to identify optimization principles can be useful for developing a taxonomy of motor behavior and gaining insight into the neural processes that produce motor behavior.

Ill-Posed Problems in Motor Behavior

Many optimization-based models in the literature have been developed to address the “excess degrees-of-freedom” problem. How does the motor system select the behavior it uses from the infinite number of possibilities open to it? In mathematical parlance, this is an “ill-posed” problem in the sense that many solutions are possible. For example, most limb segments are moved by a larger number of muscles than appear to be necessary. To reach for a cup of coffee, the hand may move along many different paths. The same figural form (e.g., the letter Z or an ellipse) may be drawn using a

wide variety of time profiles for the pen's position. The central question is how the nervous system chooses values for the large number of parameters that can be controlled. One appealing possibility is that the nervous system has evolved to select solutions that maximize the organism's fitness, i.e., that are optimal in some sense. More specifically, the hypothesis is that in performing a motor task, the brain produces coordinated actions that minimize some measure of performance (such as effort, smoothness, etc.). In this article we review several studies in which the validity of such ideas was examined in the context of planar upper limb movements. Similar ideas have been explored in the context of other effector systems and motor actions, such as whole body posture, gait, and various sporting activities, but they will not be considered here. The interested reader is referred to Winters and Crago (2000) for further information.

Arm Trajectory Formation

Our first topic is the kinematic aspects of movement. *Kinematics* refers to the time course of limb position, velocity, etc., while *dynamics* refers to variables such as forces and torques. In principle, even a single-degree-of-freedom movement (e.g., elbow rotation) can be performed in many different ways. Thus, while the hand path is constrained to follow a circular arc, its speed along the path may follow many different time profiles. One way to gain insight into the processes responsible for the selection of specific limb trajectories is to experimentally observe human movements. Patterns or invariances in the observed behavior suggest hypotheses about the way these movements are organized. Optimization theory provides a mathematical tool for concisely formulating and testing these hypotheses. The key step is the identification of an *objective function* that defines a measure of performance by assigning a cost to each member of the class of possible behaviors under study (e.g., arm trajectories). One member of this class (e.g., one trajectory) will then be selected to maximize or minimize that function. How the objective function is defined determines what aspects of the motor behavior are considered important.

Kinematic Versus Dynamic Objective Functions

In this article we will consider two different types of objective functions that have been proposed (out of the multitude of possibilities) as they reflect two major competing theories of how motor computations are organized. The first type of objective function is based solely on kinematic variables (e.g., limb position and its time derivatives). If a kinematic objective function can be found that leads to optimal trajectories that accurately reproduce the patterns of observed behavior, it implies that the brain ignores nonkinematic factors in selecting and producing that behavior. This would be consistent with a theory that, to produce movement, neural computations are organized hierarchically and executed by proceeding from the abstract (i.e., move to that light over there) to the particular (i.e., activate that set of motor neurons in this manner). The most compelling evidence supporting this idea is the observation that similar kinematic patterns are observed even when widely different musculoskeletal systems are involved in producing motor behavior. One's signature on paper is equally as recognizable and distinctive as one's signature on a blackboard, despite the enormous differences in the mechanics and physiology of the body parts used to produce it. Nevertheless, a troubling aspect of this theory is that it seems to imply that, at least at the higher levels of the postulated hierarchy, the brain does not take into account *any* dynamic considerations, such as the energy required, the loads on the limb segments, or the force and fatigue limitations of its peripheral neuromuscular system.

To circumvent this problem within the framework of optimization theory, a second type of objective function may be formulated based on dynamic variables (e.g., joint torques, muscle forces, etc., and their time derivatives). If a dynamic objective function can be found that leads to optimal trajectories that accurately reproduce the patterns of observed behavior, it implies that the brain considers dynamic factors in selecting and producing that behavior. It is also consistent with a theory that neural computations to produce movement are executed in parallel, taking all relevant factors (e.g., dynamics as well as kinematics) into account simultaneously.

Single-Joint Movements

As has been frequently observed, single-joint movements are characterized by single-peaked, bell-shaped speed profiles. This finding and the tendency of natural movements to be characteristically smooth and graceful led Hogan (1984) to suggest that motor coordination can be mathematically modeled by postulating that voluntary movements are made, at least in the absence of any other overriding concerns, to be as smooth as possible. For mathematical convenience (there are many other plausible measures of smoothness), maximizing smoothness was expressed as minimizing mean-squared average jerk, the third time derivative of position. In the single-joint case,

$$C = \int_{t_0}^{t_f} \left(\frac{d^3\theta}{dt^3} \right)^2 dt \quad (1)$$

where $\theta(t)$ is the joint angle, and t_0 and t_f are the initial and final movement times, respectively. Using variational calculus, the unique time history of joint positions that minimizes this performance measure may be derived analytically. It is described by the following quintic polynomial in time:

$$\theta(t) = c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5 \quad (2)$$

where c_i , $i = 0, \dots, 5$ are unspecified coefficients whose values are determined by the conditions at the beginning and end of the movement (boundary conditions). Originally, Hogan (1984) analyzed movements that start and end at rest and therefore assumed zero initial and final velocities and accelerations. Consequently, the predicted trajectories were characterized by symmetric bell-shaped speed profiles. For movements of different amplitudes and durations, the ratio of peak speed to average speed was invariant at 1.88. For a repetitive sequence of movements, speed profiles were again symmetric and this ratio was again invariant, but with a value of 1.57. These predictions appear to be in good agreement with observation. A constant ratio of peak speed to average speed has been reported by several researchers, with values between 1.60 and 1.90, depending on the conditions of measurement. However, a distinctive feature of these minimum-jerk movements is their symmetric speed profile, and that is not always observed experimentally. For example, when enhanced accuracy of target acquisition is demanded, an asymmetric speed profile is typically observed, with the peak speed occurring earlier in the movement. This indicates that the simple minimum-jerk theory may need to be modified. One possible way to account for this asymmetry is by adding (to the objective function) a term to minimize hand-to-target error integrated across the movement. An alternative is to modify the boundary conditions.

Another alternative is to use a dynamic objective function. This requires formulation of a model of neuromuscular and skeletal mechanics to relate dynamic variables (e.g., forces) to kinematics. Hasan (1986) proposed a minimum-effort theory of single-joint movement generation based on a model that described neuromuscular behavior as equivalent to a "spring-like" element driving the limb toward a neurally defined "equilibrium position," determined

by simultaneous activation of agonist and antagonist muscle groups. Minimization of effort was expressed as follows:

$$C = \int_{t_0}^{t_f} \left(\sigma(t) \left(\frac{d\beta}{dt} \right)^2 \right) dt \quad (3)$$

where σ is the joint stiffness (describing the rate of change of the restoring force generated by the “spring-like” element with its displacement from equilibrium) and $d\beta/dt$ is the time derivative of the equilibrium position. Thus, for single-joint movements, optimization theories using both kinematic and dynamic objective functions have been applied with success. A more telling test of these theories is found in multijoint movements.

Multijoint Movements and the Question of Coordinates

The kinematics of multijoint arm movements may be represented in a number of different ways, e.g., as a series of hand positions, joint angles, or muscle lengths. Each of these may be considered as alternative “coordinate frames” for describing the movement. The neural computations underlying multijoint arm movements may make use of any one (or even several) of these representations. Experimental observations of unconstrained human reaching movements are characterized by approximately straight hand paths and symmetric bell-shaped speed profiles that remain nearly invariant despite changes in movement direction, speed, and starting position. Because these features are evident only in the motions of the hand, and not in the movements of individual limb segments, it was proposed that the neural computations underlying movement production take place in terms of hand motion through extracorporeal space and not in terms of joint rotations.

Flash and Hogan (1985) showed that the maximum-smoothness theory reproduced all of these features, provided the objective function was expressed in terms of the Cartesian coordinates of the hand as follows:

$$C = \int_{t_0}^{t_f} \left(\left(\frac{d^3x}{dt^3} \right)^2 + \left(\frac{d^3y}{dt^3} \right)^2 \right) dt \quad (4)$$

where $x(t)$, $y(t)$ describe the hand position coordinates and t_f is the movement duration.

Minimizing this objective function yielded analytic expressions for the hand trajectories. For unrestrained point-to-point movements starting and ending at rest, the model predictions agreed closely with experimental data and successfully accounted for the invariance of hand trajectories under translation, rotation, amplitude and speed scaling.

In more complex curved movements, patterns were again evident in hand kinematics, but not in joint kinematics. When subjects were instructed to generate curved or obstacle-avoidance movements, although the hand paths appeared smooth, movement curvature was not uniform; the trajectories displayed two or more curvature maxima. The hand speed profiles also had two or more maxima, and the minima between adjacent peaks temporally corresponded to the peaks in curvature.

To describe curved and obstacle-avoidance movements, the maximum-smoothness model was extended by assuming that a small number of points along the path through which the hand should pass are specified (Flash and Hogan, 1985). The time of passage through those “via” points and the hand velocity at that time were not specified a priori but were predicted by the model. For the simplest case of one via point between the initial and final positions, the theory yielded explicit mathematical expressions for the hand motion (Flash and Hogan, 1985) that reproduced all the features of the experimental observations: distinct maxima in the speed profile with a minimum between them which coincided temporally with a curvature maximum; trajectory shape invariant under

translation, rotation, amplitude, and time scaling; and nearly equal durations of movement from the initial position to the via point, and from the via point to the final position. The latter observation was referred to as the *isochrony principle* (Viviani and Terzuolo, 1982), or the phenomenon that movement durations of large and small segments of a trajectory are roughly equal.

Minimum Torque Change Models

In contrast to the maximum-smoothness model, Uno, Kawato, and Suzuki (1989) postulated that movement selection optimizes the rate of change of actuator efforts, e.g., joint torques. Although minimizing jerk and minimizing the rate of change of joint torques appear conceptually similar (in a single-joint system with predominantly inertial dynamics they are proportional to one another), there are important differences. First, the objective function is based on dynamic variables: the rate of change of torque. Therefore, the predicted motion depends sensitively on the modeled dynamic behavior of the musculoskeletal system. Second, the objective function was formulated in terms of joint torques rather than functions of the hand’s coordinates, as is the case for minimum jerk. This implies that motor computations are based on a joint-space representation of behavior. Although (as outlined above) kinematic patterns are most evident in hand motions in extracorporeal space, approaches based on either joint or muscle spaces have the advantage that they can generate solutions to important aspects of the ill-posed motor control problems, such as kinematic redundancy (the apparent excess degrees of freedom) or actuator redundancy (the apparent excess of muscles). The maximum-smoothness model expressed in hand coordinates does not address these issues.

Initially, Uno et al. (1989) reported that the performance of the minimum torque change model surpassed that of the maximum-smoothness model. It appeared to account for the small but systematic curvature seen in point-to-point movements, and also for the larger curvature seen in movements that pass from the side to the front of the body. However, an independent study (Flash, 1990) and a later reappraisal co-authored by some of the original proponents of the minimum torque change model (Nakano et al., 1999) invalidated these conclusions: it was shown that a combination of too large an inertia and too small a viscosity contingently led to predictions compatible with experimental results.

However, in Nakano et al. (1999), a variant of the minimum torque change model, the minimum *commanded* torque change model, was introduced. In this model the commanded torque includes non-zero viscous terms that arise from biochemical and mechanical reaction processes within the muscles, and in this way both the muscles and the link dynamics are considered as controlled objects. Using more realistic, measured physical parameters, this second model was again able to reproduce the experimentally verified effects of curvature.

Motor Adaptation Studies

The most critical comparison of these two models arises from their fundamental differences. According to kinematically based optimization models, neural computations specify intended motions independently of movement dynamics or external load conditions. In contrast, dynamically based optimization models imply that external loads profoundly influence intended motions. For example, according to the minimum torque change models, movements in the presence of elastic loads should be more curved than unloaded movements, whereas the maximum-smoothness model predicts no effect.

Investigating motor adaptation to elastic loads, Uno et al. (1989) concluded that the behavior in the presence of the load was different from the unloaded case. Completely different results, however,

were obtained in another study in which static elastic loads were unexpectedly introduced during human reaching toward visual targets (Flash and Gurevich, 1997). In the first few trials following load application, movements were found to be misdirected and to miss the target, but after a small number of practice trials (five to seven), the loaded movements tended to follow straight hand paths with symmetric velocity profiles. In another study (Shadmehr and Mussa-Ivaldi, 1994), velocity-dependent force fields were used to perturb the motion, and the perturbed trajectories performed in the presence of the new force fields were again found to converge toward the ones seen in the unloaded case. In a third related study, Wolpert, Ghahramani, and Jordan (1995) used altered visual feedback conditions that caused an increase in the perceived curvature of aiming movements. This led to significant corrective adaptation of the movements actually produced: the hand movements became curved, thereby reducing the visually perceived curvature. These results support the notion that arm trajectories follow a kinematic plan formulated in extrinsic visual space, independent of movement dynamics or external force conditions. They are incompatible with the assumptions of dynamically based optimization models formulated in terms of intrinsic coordinates.

Furthermore, it should be noted that small deviations from straight-line movements do not necessarily imply planning in joint coordinates. Such phenomena are compatible with planning in kinematic space, but with perturbations due to the dynamics of the arm and neurally controlled muscles at the implementation stage (Flash, 1990). Conclusions with respect to the sensorimotor mapping that associate desired trajectories to motor commands were drawn based on motor adaptation studies. Shadmehr and Mussa-Ivaldi (1994) have analyzed the aftereffects observed when, after training in one region of the work space, subjects were asked to perform reaching movements at a nearby space. The patterns of aftereffects suggested that generalization from learning was in terms of intrinsic joint-based coordinates.

Relation to Physiology

The kinematic and dynamic objective functions discussed above are based on measures of smoothness in different coordinate frames. Both of these models have in common that they are *phenomenological* approaches. The controller (nervous system) and plant (arm) are treated as a *black box*, with the input the experimental task and the output the goal-f fulfilling movement. The success (or otherwise) of phenomenological theories in fitting experiment affords insight into which variables the central nervous system (CNS) might consider important in the movement planning process. Results have been presented above that support the idea that the high-level planning processes in the CNS might be in the coordinates of the hand's position.

The fact that movements are smooth, whether in hand or joint coordinates, has been interpreted as compatible with increasing the predictability of the trajectory or reducing the amount of information needed to internally represent motion plans (Hogan, 1984; Flash and Hogan, 1985). Smoothness maximization and the superposition of elemental movements to generate more complicated arm trajectories are also closely related to regularization-based approaches to learning from examples. Those approaches view learning as equivalent to identifying a function from sparse and noisy data. The trade-off between accurate data reproduction and "well-behavedness" of the mapping is achieved by maximizing the smoothness of the function.

Work has also been done on how the CNS might implement an optimization procedure such as minimum jerk. For example, it has been shown that a minimum-jerk movement planner can be directly

implemented by a radial basis function (RBF) network. Another implementation scheme was described by Hoff and Arbib (1992), who showed how the minimum-jerk principle could be converted into a real-time controller in which delays and noise effects could explain a number of experimental observations beyond the fitting of simple point-to-point trajectories. However, looking for neural circuits that can reproduce explicitly the calculations inherent in the phenomenological theories of minimum jerk and minimum torque might be a too literal interpretation of the success of such theories in reproducing experiment. The kinematic and psychophysical observations reported to date do not sufficiently constrain the possible movement-generating algorithms to distinguish the finer details of neural implementation. Nevertheless, these phenomenological theories serve as background, coarse-grained descriptors to which deeper, more biologically detailed theories must conform.

Recently, some effort has been made in grounding the optimization approach to motor control in a neurobiological context. It was noted that biological systems are corrupted by noise, the variance of which increases with the size of the signal (Harris and Wolpert, 1998). Hence, any preplanned movement is likely to be off-target when the motor program is run through the noisy neuromuscular system. As each goal-directed movement has some characteristic level of error, this suggests a natural optimization criterion: the CNS chooses movements that minimize the final error in the achievement of the motor task.

Harris and Wolpert (1998) analyzed the predictions of this hypothesis in the context of saccadic eye movements. The error in the final eye position was functionally minimized with respect to the control signal (using a linear model of the plant). It was found that small final error was achieved by low-bandwidth neuronal signals, corresponding to the smooth velocity profiles seen experimentally. This approach was also extended to arm movements in the particular case of two-joint motions in the plane (Harris and Wolpert, 1998). A large range of experimental results were successfully reproduced, including the small curvature seen in point-to-point movements. Furthermore, it was claimed that the predicted trajectory of the hand was, to a large degree, independent of the specifics of the model of the plant: the controlling neuronal signal adapts to produce similar output.

The role of noise in the coordination of movement has been further examined in the context of *optimal feedback control* (Todorov and Jordan, 2001). It was noted that the variability and redundancy inherent in, for example, the control of the human arm are often treated as problems to be overcome in the planning process. In their work on optimal feedback control, Todorov and Jordan showed that increased accuracy in the goal-specific parameters of movement can be obtained by allowing the variance to increase in the redundant variables. In fact, their model does not enforce a desired trajectory but corrects only those deviations that interfere with the task, a principle of *minimum intervention*. Despite this minimal formulation, experimentally observed features such as simplifying rules, control parameters, and synergies emerge as epiphenomena of the control process. The theory is supported by a number of exemplary experimental results and provides a satisfying interpretation of the role of variability and the so-called degrees-of-freedom problem.

Motion Planning for Three-Dimensional Movements

For completeness, we briefly mention recent work on motion in three dimensions (i.e., not confined to a plane). Compared with the success of optimization techniques in two dimensions, the use of cost function analysis is still in the investigative phase for this more general class of motion. It is known that point-to-point motions in

three dimensions are considerably more curved than in the plane. Nevertheless, there has been some success in predicting this more complex behavior using the techniques of the optimization approach. Hypothesized cost functions have included minimum kinetic energy (Soechting et al., 1995), in which it was also shown that a simple Donder's law rule that expresses a kinematic constraint on eye orientation does not apply to arm motions. Other models that incorporate a description of muscle dynamics and hypothesize the minimization of a metabolic energy cost or consider the effect of final posture of the arm have also been developed, representing attempts to deal with the acute degrees-of-freedom problem found in three-dimensional movements.

Discussion

One of the exciting challenges of brain theory is the need to deal with reality at the level of whole, functioning systems. Traditionally, scientific endeavor has advanced our state of knowledge by delving into finer and finer details of isolated pieces of reality—the essence of the reductionist approach. However, because of the limited amount known of these fine details and the difficulties involved in studying complex systems of many neurons, this bottom-up approach is severely limited in its ability to describe systemwide behavior: large-scale, strongly interacting systems exhibit characteristics that emerge primarily from interactions among their parts. To understand them, a top-down approach is far more effective, beginning at a coarse-grained macroscopic level and proceeding to finer levels of detail as their structure is discerned. Optimization theory provides a powerful set of mathematical tools that lend themselves well to a top-down approach to studying the brain. As we have reviewed in this article, optimization theory facilitates a rigorous approach, based on macroscopic observations of psychophysical behavior, to some fundamental and far-reaching questions about the structure of neural computations.

Road Map: Mammalian Motor Control

Background: Motor Control, Biological and Theoretical

Related Reading: Equilibrium Point Hypothesis; Limb Geometry, Neural Control; Sensorimotor Learning

Orientation Selectivity

Robert Shapley, David McLaughlin, and Michael Shelley

Introduction

The detection of edge information from within a visual scene is an essential component of visual processing. This processing is believed to be initiated in the primary visual cortex, where individual neurons are known to act as feature detectors of the orientation of edges within the visual scene. Individual neurons can have an *orientation preference* (which states that neuron's preferred orientation of the angle of edges) and an *orientation selectivity* (which measures the neuron's sensitivity as a detector of orientation).

This article considers mechanisms of orientation selectivity in the visual cortex. In V1 there is a transformation to orientation-tuned elements (Hubel and Wiesel, 1962). Along the visual pathway prior to V1, in the retina and lateral geniculate nucleus (LGN) of the thalamus, there is weak or no orientation selectivity in single cells. It has been thought from the time of its discovery that orientation selectivity, as an emergent property in visual cortex, must be an important clue to how the cortex works and why it is built

References

- Flash, T., 1990, The organization of human arm trajectory control, in *Multiple Muscle Systems: Biomechanics and Movement Organization* (J. Winters and S. Woo, Eds.), New York: Springer-Verlag, pp. 282–301.
- Flash, T., and Gurevich, I., 1997, Arm trajectory generation and stiffness control during motor adaptation to external loads, in *Self-Organization, Computational Maps and Motor Control* (P. G. Morasso and V. Sanguinetti, Eds.), Amsterdam: Elsevier, pp. 423–482.
- Flash, T., and Hogan, N., 1985, The coordination of arm movements: An experimentally confirmed mathematical model, *J. Neurosci.*, 5:1688–1703. ♦
- Harris, C. M., and Wolpert, D. M., 1998, Signal-dependent noise determines motor planning, *Nature*, 394:780–784. ♦
- Hasan, Z., 1986, Optimized movement trajectories and joint stiffness in unperturbed inertially loaded movements, *Biol. Cybern.*, 53:373–382.
- Hoff, B., and Arbib, M. A., 1992, A model of the effects of speed, accuracy, and perturbation on visually guided reaching, in *Control of Arm Movement in Space: Neurophysiological and Computational Approaches* (R. Caminiti, P. B. Johnson, and Y. Burnod, Eds.), *Experimental Brain Research Series* 22:285–306.
- Hogan, N., 1984, An organizing principle for a class of voluntary movements, *J. Neurosci.*, 4:2745–2754. ♦
- Nakano, E., Imamizu, H., Osu, R., Uno, Y., Gomi, H., Yoshioka, T., and Kawato, M., 1999, Quantitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model, *J. Neurophysiol.*, 81:2140–2155.
- Shadmehr, R., and Mussa-Ivaldi, F. A., 1994, Adaptive representation of dynamics during learning of a motor task, *J. Neurosci.*, 14:3208–3224. ♦
- Soechting, J. F., Buneo, C. A., Hermann, U., and Flanders, M., 1995, Moving effortlessly in three dimensions: Does Donder's law apply to arm movements? *J. Neurosci.*, 15:6271–6280. ♦
- Todorov, E., and Jordan, M. I., 2001, Optimal feedback control as a theory of motor coordination, available: <http://www-rcf.usc.edu/etodorov/>. ♦
- Uno, Y., Kawato, M., and Suzuki, R., 1989, Formation and control of optimal trajectory in human multijoint arm movement: Minimum torque-change model, *Biol. Cybern.*, 61:89–101. ♦
- Viviani, P., and Terzuolo, C., 1982, Trajectory determines movement dynamics, *Neuroscience*, 7:431–437.
- Winters, J. M., and Crago, P., 2000, *Biomechanics and Neural Control of Posture and Movement*, New York: Springer-Verlag. ♦
- Wolpert, D. M., Ghahramani, Z., and Jordan, M. I., 1995, Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study, *Exp. Brain Res.*, 103:460–470. ♦

the way it is. Much has been learned about the basic principles of cortical neurophysiology through intense investigations of orientation selectivity.

Models of Orientation Selectivity

Feedforward Models

There are two schools of thought about the explanation for cortical orientation selectivity: feedforward filtering, on the one hand, and strong excitatory corticocortical feedback on the other. The models of the latter, with sufficiently strong excitatory feedback, possess “attractor states” that are intrinsic to the nonlinear cortical dynamics.

Our view, based on experimental results and also on our own modeling, is somewhere in the middle; perhaps a good label for our view of the cause of orientation selectivity in V1 would be *recurrent network filtering*. However, the first view proposed his-

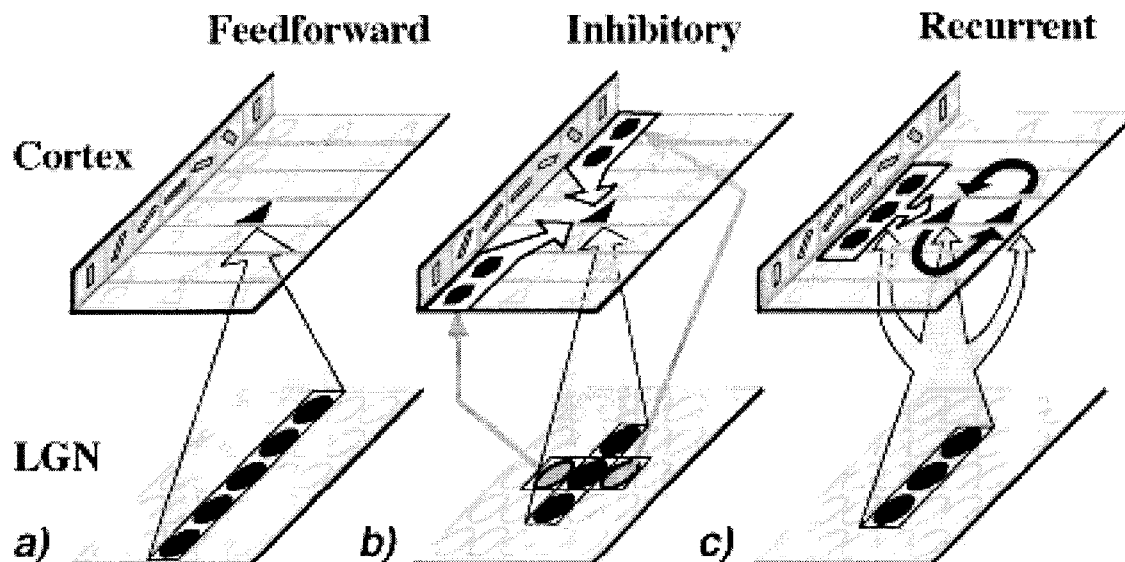


Figure 1. Sketch of different models for cortical orientation selectivity. At the cortical level, hexagonal shapes stand for inhibitory neurons and triangular shapes stand for excitatory neurons. The LGN cells are depicted as the circular shapes at the LGN level in the models. *A*, Feedforward. All cortical cells receive input from a row of LGN cells aligned in space. *B*, In inhibitory models, there is inhibitory input from orientation-tuned cortical neurons, indicated by the outlined white arrows. *C*, Recurrent excitatory

and inhibitory feedback models have both corticocortical inhibition (white arrows) and corticocortical excitation (black arrows) impinging on all cortical cells. These corticocortical interactions are supposed to greatly sharpen a broadly tuned input from LGN (indicated by the shorter row of aligned LGN cells). (From Somers, D. C., Nelson, S. B., and Sur, M., 1995, An emergent model of orientation selectivity in cat visual cortical simple cells, *J. Neurosci.*, 15:5448–5465. Reprinted with permission.)

torically and the first one discussed here is the feedforward view, which descended from the pioneering work of Hubel and Wiesel (1962). From the time of its publication, Hubel and Wiesel's feedforward model has been a dominant idea in this field. Figure 1, from Somers, Nelson, and Sur (1995), compares and contrasts the feedforward model in panel *A* with cortical interaction models in panels *B* and *C*. As shown in the figure, the HW model involves the addition of signals from LGN cells that are aligned in a row along the long axis of the receptive field of the orientation-selective neuron. In the HW model, this collection of LGN cells, taken together, sets the orientation preference and selectivity of that cortical cell onto which the LGN cells converge. The experiment on the cooling of cat V1 by Ferster, Chung, and Wheat (1996) is an important result that was interpreted to mean that there is substantial orientation tuning of the collective thalamic input to a cortical neuron, consistent with the HW model. In spite of this evidence, several authors agree that the HW model predicts rather little orientation selectivity, and therefore does not account for the visual properties of V1 cells (Sompolinsky and Shapley, 1997; Troyer et al., 1998; McLaughlin et al., 2000).

The reason for the shortfall of orientation selectivity in the HW model can be stated as follows. LGN cells have a low spontaneous rate but are quite responsive to visual stimuli, so their firing rate during visual stimulation clips at zero spikes per second. Because of this rectification, LGN cells act like nonlinear excitatory subunits as inputs to their cortical targets. Since the HW model simply adds up the LGN sources, and each of these responds to every orientation, the model's summation of the clipped LGN inputs would cause it to have a non-zero response at 90° from the optimal orientation. In fact, the HW model predicts that the total number of spikes elicited by a stimulus could be the same at 90° as at 0° , although the spikes would be more spread out in time at 90° (Sompolinsky and Shapley, 1997; Troyer et al., 1998). Computational simulations of the HW model have demonstrated that this analysis is correct (Sompolinsky and Shapley, 1997; McLaughlin et al.,

2000), as illustrated in Figure 2, where the orientation selectivity of an HW model is shown. But experimental observations establish that many cortical cells respond little or not at all at 90° from peak orientation, so we must conclude that the HW convergence mechanism is only part of the story of cortical orientation selectivity. In the literature on orientation selectivity, it is often stated that the nonlinearity of the neuronal spike threshold could cause major sharpening of the orientation tuning curve in the cortical cell response even if the convergent LGN input is as broad as in Figure

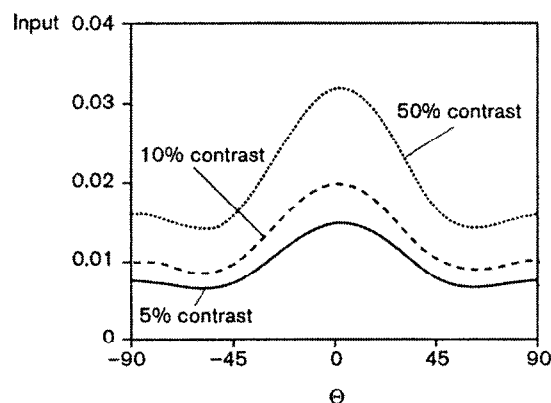


Figure 2. Orientation tuning of a feedforward model. Many LGN elements are assumed to converge via excitatory synapses onto single cortical cells in this model. No corticocortical excitation or inhibition modifies the feedforward excitation. Responses at three contrast levels are shown: 5% 10%, and 50% contrast. At all levels of contrast, there is a non-zero voltage generated 90° away from the preferred orientation. (After Sompolinsky and Shapley, 1997.)

2. This sharpening based on nonlinear thresholding is sometimes termed the iceberg effect. The iceberg effect is not a practical way to obtain sharpening, however. If the threshold is set so high that it causes the requisite significant sharpening, it will also diminish response magnitude a great deal, causing a loss of sensitivity. Also, thresholding works only at one stimulus contrast level. To obtain orientation tuning that does not broaden a great deal at high contrast, one needs another kind of mechanism.

One may wonder why the experiments of Ferster et al. (1996) are not decisive pieces of evidence for the hypothesis that LGN convergence causes a large amount of orientation selectivity. Here is our analysis of these experiments. The cooling experiments of Ferster et al. measured the first-order temporal Fourier component (F1) in the intracellular voltage response to a drifting grating from a neuron in a cooled cortex, and found it to be as tuned for orientation as in the warm cortex. But this does not account for the tuning of the mean spike rate, which is tuned as much as the F1 component of the spike rate. Also, the mean spike rate during a stimulus is the most often measured component of the neuron's response, and thus it is important to explain the tuning of this component. It is known from the work of Ferster et al. and others that the mean spike rate is an approximately linear function of mean membrane voltage (above a threshold value). So, to attempt to account for the orientation selectivity that is observed in the mean spike rate, it would have been necessary for Ferster et al. to measure also the orientation selectivity of the mean intracellular voltage response in the cooled cortex. They did not do this in the cooling experiments reported in 1996, probably for technical reasons. But suppose they had measured the DC response in the cooled cortex, and suppose the HW model does describe the LGN input to V1 neurons. Then we can predict that Ferster et al. would have found the mean (or DC) intracellular voltage response to be only weakly selective for orientation (based on the analysis of the HW model in the previous paragraph and Figure 2). Because the HW model predicts substantial membrane voltage response at 90° from peak, it cannot account for the sharp tuning of mean spike rate that is observed in experiments.

One can go further and analyze why Ferster et al. found so much orientation tuning for the F1 component. The only mechanism for orientation-selective response for an F1 response in the HW model is different spatial frequency resolution along the two axes of the elliptical receptive field. The reason is, the HW model has no inhibition; it is a purely excitatory model. Thus (if indeed the HW model applied), Ferster et al. probably observed orientation selectivity in the cooled cortex only because the spatial frequency of the grating they used was too high for the elliptical LGN array to resolve it along the long axis. However, the spatial frequency was not too high for the summed LGN input to resolve it along the minor axis. To put it another way, if orientation selectivity for the F1 Fourier component depended on feedforward LGN convergence as in the HW model, it would be very strongly dependent on spatial frequency, but in the normally functioning cortex, it is not, as reported 20 years ago by Jones and Palmer and then a few years after that by Webster and DeValois. The conclusion of all these considerations of the cooling experiment is that it is not decisive evidence for a feedforward explanation of orientation selectivity.

Cortical Inhibition

One possible addendum to the HW model that increases the orientation selectivity greatly is to add inhibition, either push-pull inhibition (Troyer et al., 1998) or some other kind of inhibition that is broadly distributed in orientation (Somers et al., 1995; Ben-Yishai, Bar-Or, and Sompolinsky, 1995; McLaughlin et al., 2000). But given what is known about V1, this inhibition must come through cortical interneurons rather than directly from the thalamic afferents. Such a model is diagrammed in Figure 1, panel B. Ex-

periments on intracortical inhibition in V1 have given mixed results. Initially, Adam Sillito's experiments with bicuculline suggested that intracortical inhibition might be necessary for orientation tuning. However, subsequent experiments by Sacha Nelson and colleagues at MIT, involving blocking inhibition intracellularly, have been interpreted to mean that inhibition of a single neuron is not necessary for that neuron to be orientation tuned. However, the role of intracortical inhibition has been supported by the work of A. B. Bonds and his collaborators. They have studied interactions between stimuli at different orientations, the effects of blocking activity in infragranular layers, and the effects of GABA on orientation selectivity. More recently, Ulf Eysel and his collaborators in Germany have accumulated a body of evidence in cat cortex for the important role of inhibition in causing a sharpening of orientation selectivity. Eysel and co-workers have blocked the lateral spread of cortical inhibition with local injection of inhibitory agonists that block local activity. When they do this, they often observe broadening of orientation tuning curves.

A theory of orientation tuning in cat cortex offered by Troyer et al. (1998) attempts to explain orientation tuning in terms of specific "push-pull" inhibition in which there is phase-specific inhibition superimposed on phase-specific excitation of the opposite sign. However, the main mechanism for sharpening of orientation tuning in the Troyer model is corticocortical inhibition that is broadly tuned for orientation. In the Troyer model there is moderately tuned LGN-convergent excitation from an HW mechanism, and then more broadly tuned inhibition that cancels out the wide-angle responses but leaves the tuning curve around the peak orientation relatively unchanged. Therefore, this model is one of a class of corticocortical interaction models for orientation selectivity.

More recently, we have developed a large-scale model of four hypercolumns in layer 4c α of macaque V1. The hypercolumn, a compact cortical region approximately 0.5 mm \times 0.5 mm in area, is the unit of cortical processing. All orientations are represented in a hypercolumn, by arrays of neurons of similar orientation preference arranged as if they were spokes of a pinwheel around a central singularity, the pinwheel center. This architecture has been deduced from the results of experiments with intrinsic optical imaging by Bonhoeffer and Grinvald (1993). Our model incorporates known facts about the physiology and anatomy of V1. This model accounts for many visual properties of V1 neurons, especially orientation selectivity. Inspired by the Somers et al. (1995) and Ben-Yishai et al. (1995) models, it seeks to account for the same set of phenomena as these models but with more biological realism. One novelty in our model is that the spatial strength of connections between neurons is taken to be the spatial density of synaptic connections revealed by anatomical investigations of cortex. The model places the "footprints" of synaptic excitation and inhibition on the pinwheel latticework that is revealed by optical imaging (Bonhoeffer and Grinvald, 1993). In our model the spatial scale of corticocortical excitation exceeds that of inhibition, as indicated by cortical neuroanatomy. In its focus on the visual-functional consequences of the pinwheel organization, our model is novel and original. Our model causes significant sharpening and also diversity of orientation selectivity, and produces simple cells. The most significant difference between this model and that of Troyer et al. (1998) is that in our model (the McLaughlin model), the inhibitory conductance input to a cell is phase-insensitive (the opposite of push-pull). This happens because inhibition of a model cell is a sum from many neural sources, and it is likely that each of these sources is a cortical inhibitory cell with a fixed phase preference different from those of neighboring neurons. It is also consistent with the measured phase insensitivity of measured inhibition (Borg-Graham, Monier, and Fregnac, 1998; Anderson, Carandini, and Ferster, 2000). Anderson et al. (2000) state that their data support a push-pull, that is, phase-sensitive inhibition model. However, a close scrutiny of their data reveals that much of the mea-

sured inhibitory conductance (in response to drifting gratings) is a phase-insensitive elevation of inhibition, as predicted by the McLaughlin model (a point discussed in Wielaard et al., 2001).

Cortical Excitation and Attractor Models

The idea that corticocortical excitatory feedback plays a crucial role in orientation tuning has been put forward most forcefully by theorists of brain function. Several well-known papers make the case for this corticocortical feedback. One, by Somers et al. (1995), presents an elaborate computational model for orientation tuning. Another paper in this genre is by Ben-Yishai et al. (1995), who offer an analytical model from which they make several qualitative and quantitative predictions. One of their important theoretical results is that one cannot predict contrast invariance of orientation tuning with feedforward models, but the feedback model of Ben-Yishai et al., with recurrent excitation and inhibition, does exhibit contrast invariance. Another of their results is that if recurrent feedback is strong enough, one will observe the “marginal phase” state in which V1 behaves like a set of attractors for orientation. Using a ring model that resembles the architecture of the Ben-Yishai model, Pugh et al. (2000) demonstrated that they could account for some of the important features of orientation dynamics that could not be explained by feedforward models, features such as the Mexican-hat tuning in orientation.

The attractor states of recurrent excitatory models are discussed not only by Ben-Yishai et al. (1995) but also by Tsodyks et al. (1999). Using intrinsic optical imaging of visual cortex, Tsodyks et al. found that there were patterns of spontaneous activity that resembled the patterns evoked during stimulation with oriented gratings. This provided some evidence for the idea that there were active states of cortical activity associated with orientation selectivity. The concept is that very weakly orientation-selective feedforward signals can be massively sharpened by strong recurrent excitatory feedback, causing the cortex to respond to any visual signal by relaxing into a state of activity governed by the pattern of corticocortical feedback. We believe that this theory, like the pure feedforward theory, has trouble explaining some important data, for instance, the existence of simple cells in which response waveforms follow the time course of the stimulus faithfully.

Bandwidth and Circular Variance

There are different ways to measure orientation selectivity, and they can tell us about different aspects of orientation selectivity. A traditional method is to determine the half-bandwidth of the tuning curve around the peak of the tuning. This indicates the shape of the tuning curve near the peak. However, there are important ques-

tions about mechanisms that depend on the global shape of the tuning curve at all orientations. Various vector-averaging measures have been devised by different investigators. We favor the use of circular variance, a measure that is used in circular statistics. If we write the spike rate as a function of angle as $m(\theta)$, the circular variance of $m(\theta)$ is:

$$CV[m] = 1 - \left| \frac{\int m(\theta) \exp(2i\theta) d\theta}{\int m(\theta) d\theta} \right|$$

Circular variance is $1 - \{\text{relative modulation of } m(\theta) \text{ as a function of } \theta\}$. The relative modulation is the ratio of the best-fitting Fourier component of the orientation tuning curve (with period equal to 180°), divided by the average response. For a flat tuning curve, $CV = 1$. For a very highly tuned tuning curve, CV approaches 0. CV reflects wide-angle responses that the bandwidth does not. Other investigators have also used global measures for selectivity that are related to circular variance.

Response Dynamics

In an attempt to provide a database to test models of orientation selectivity, Ringach, Hawken, and Shapley (1997) applied the sub-space reverse correlation method. The idea was to measure the time evolution of orientation selectivity extracellularly in single V1 neurons. One main result from the use of this technique is that there is evidence for a slightly delayed inhibition or suppression in orientation selectivity's time evolution. Also, in a few neurons one observes a progressive shift of the peak orientation with time. The presence of shifter cells provides a realization of the “marginal mode” (a name for one of the attractor states referred to above) predicted by Ben-Yishai et al. (1995), and thus provides some confirmation of the attractor states of the recurrent models with strong corticocortical excitation. However, such neurons are found to be the exception, not the rule, in the reverse correlation observations. What one does observe often are highly selective cells in the output layers of the cortex in whose activity there is the following pattern: a delayed suppression at the orientation that is the peak orientation early in the response. The suppression causes it to become the least preferred orientation late in the response. Thus, in neurons with such delayed suppression, one observes a change in the preferred orientation later in the response, also. But unlike the shifter neurons, the suppressed neurons have a rather sudden flip in preferred orientation, usually by as much as 90° .

The time dependence of orientation tuning is illustrated for one macaque 4α neuron in Figure 3, row *a*. This neuron, like many

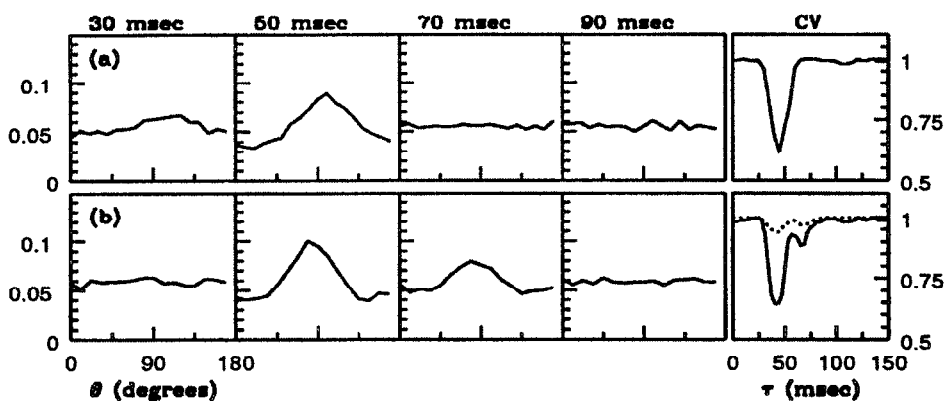


Figure 3. Time dependence of orientation tuning in a macaque 4α neuron and in the large-scale model of McLaughlin et al. (2000). In the first four panels, $p(\theta, \tau)$ is graphed. In the fifth panel the circular variance (CV) is shown. Row *a* is for a typical 4α neuron; row *b* is from the model. The dotted line in the rightmost panel of row *b* is the CV versus time for a pure feedforward model.

cells in the input layer 4c α , does not exhibit much late suppression. Figure 3 also illustrates that a feedforward model would produce little orientation selectivity in the reverse correlation experiment. This is illustrated in the rightmost panel of row *b*, in which circular variance (CV) is drawn for the full model (solid line) and for a pure feedforward model (dotted line).

Discussion

In our view, corticocortical inhibition is a crucial ingredient in the emergence of orientation selectivity in the visual cortex. The orientation preference of each neuron, and the orderly orientation preference map, are likely to be consequences of the pattern of feedforward convergence. However, the selectivity observed in steady-state experiments, and even more so in orientation dynamics experiments, cannot be achieved by a purely feedforward model. At present, we cannot yet evaluate the relative importance of corticocortical excitation in enhancing orientation selectivity. It may play a role for some neurons. For cortical simple cells, the results of our modeling indicate that corticocortical inhibition dominates, and that in simple cells, cortical excitation has to be relatively weak compared to the LGN excitation (see Wielaard et al., 2001). However, it is highly likely that corticocortical excitation is much more significant for the function of complex cells in V1.

Road Map: Vision

Related Reading: Ocular Dominance and Orientation Columns; Visual Cortex: Anatomical Structure and Models of Function

References

- Anderson, J. S., Carandini, M., and Ferster, D., 2000, Orientation tuning of input conductance, excitation, and inhibition in cat primary visual cortex, *J. Neurophysiol.*, 84:909–926.
- Ben-Yishai, R., Bar-Or, R. L., and Sompolinsky, H., 1995, Theory of orientation tuning in visual cortex, *Proc. Natl. Acad. Sci. USA*, 92:3844–3848.
- Bonhoeffer, T., and Grinvald, A., 1993, The layout of iso-orientation domains in area 18 of cat visual cortex: Optical imaging reveals a pinwheel-like organization, *J. Neurosci.*, 13:4157–4180.
- Borg-Graham, L. J., Monier, C., and Fregnac, Y., 1998, Visual input evokes transient and strong shunting inhibition in visual cortical neurons, *Nature*, 393:369–373.
- Ferster, D., Chung, S., and Wheat, H., 1996, Orientation selectivity of thalamic input to simple cells of cat visual cortex, *Nature*, 380:249–252.
- Hubel, D. H., and Wiesel, T. N., 1962, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *J. Physiol.*, 160:106–154.
- McLaughlin, D., Shapley, R., Shelley, M., and Wielaard, J., 2000, A neuronal network model of sharpening and dynamics of orientation tuning in an input layer of macaque primary visual cortex, *Proc. Natl. Acad. Sci. USA*, 97:8087–8092.
- Pugh, M. C., Ringach, D. L., Shapley, R., and Shelley, M. J., 2000, Computational modeling of orientation tuning dynamics in monkey primary visual cortex, *J. Comput. Neurosci.*, 8:143–159.
- Ringach, D., Hawken, M., and Shapley, R., 1997, The dynamics of orientation tuning in the macaque monkey striate cortex, *Nature*, 387:281–284.
- Somers, D. C., Nelson, S. B., and Sur, M., 1995, An emergent model of orientation selectivity in cat visual cortical simple cells, *J. Neurosci.*, 15:5448–5465.
- Sompolinsky, H., and Shapley, R., 1997, New perspectives on the mechanisms for orientation selectivity, *Curr. Opin. Neurobiol.*, 7:514–522. ♦
- Troyer, T. W., Krukowski, A. E., Priebe, N. J., and Miller, K. D., 1998, Contrast-invariant orientation tuning in cat visual cortex: Thalamocortical input tuning and correlation-based intracortical connectivity, *J. Neurosci.*, 18:5908–5927.
- Tsodyks, M., Kenet, T., Grinvald, A., and Arieli, A., 1999, Linking spontaneous activity of single cortical neurons and the underlying functional architecture, *Science*, 286:1943–1946.
- Wielaard, J., Shelley, M., McLaughlin, D. M., and Shapley, R. M., 2001, How simple cells are made in a nonlinear network model of the visual cortex, *J. Neurosci.*, 21:5203–5211.

Oscillatory and Bursting Properties of Neurons

Xiao-Jing Wang and John Rinzel

Introduction

Rhythmicity is a common feature of temporal organization in neuronal firing patterns. Historically, when recordings from *isolated* nerves became possible in the 1930s, systematic study of repetitive firing behaviors ensued. Arvanitaki (1939) and Hodgkin (1948, see citation in Rinzel and Ermentrout, 1998) identified three categories of crustacean axons by their rhythmic discharge patterns: those that fire repetitively over a wide (I) or narrow (II) range of frequencies and those whose firing hardly repeats (III). Later, Arvanitaki also pioneered the *Aplysia* preparation and discovered *bursting* oscillations, in which impulse clusters occur periodically, separated by phases of quiescence.

Since then, many other stereotypical single-neuron firing patterns, including a fascinating variety of endogenous oscillations, have been identified (Llinás, 1988; Connors and Gutnick, 1990). One wonders anew about categorizing neuronal firing modes and the criteria on which to base such a classification. In 1952, Hodgkin and Huxley showed that many spiking properties could be explained in terms of various active ionic currents across the cell membrane. Today, many types of ion channels are known, and some particular neuronal rhythms have been linked to selected subsets of channels. However, membrane potential oscillations with

apparently similar characteristics can be generated by different ionic mechanisms and by other biophysical factors, such as cable properties. In addition, a given cell type may display several different firing patterns under different neuromodulatory conditions. For these reasons, the visual appearance of particular voltage time courses and the presence of certain ionic mechanisms are insufficient bases for classification. A rational scheme should consider a cell's complete *repertoire* of dynamical modes and the nature of the transitions between modes.

Here we apply the mathematics of dynamical systems to describe precisely the dynamical modes of neuronal firing and the transformations between them. The approach was pioneered by FitzHugh (1961) with his phase space analysis of nerve membrane excitability. In this theoretical framework, membrane dynamics is described by coupled *differential equations*, e.g., à la Hodgkin and Huxley (cf. Rinzel and Ermentrout, 1998), the behavior modes by *attractors*, and the transitions between modes by *bifurcations*. The rest state is represented by a time-independent *steady state* and repetitive firing by a *limit cycle*. The transition from resting to oscillating typically occurs via either a *Hopf bifurcation* or a *homoclinic bifurcation* (Figure 1) (see, e.g., Rinzel and Ermentrout, 1998). The firing frequency versus applied current curves are qualitatively different in the two cases (minimum frequency being non-

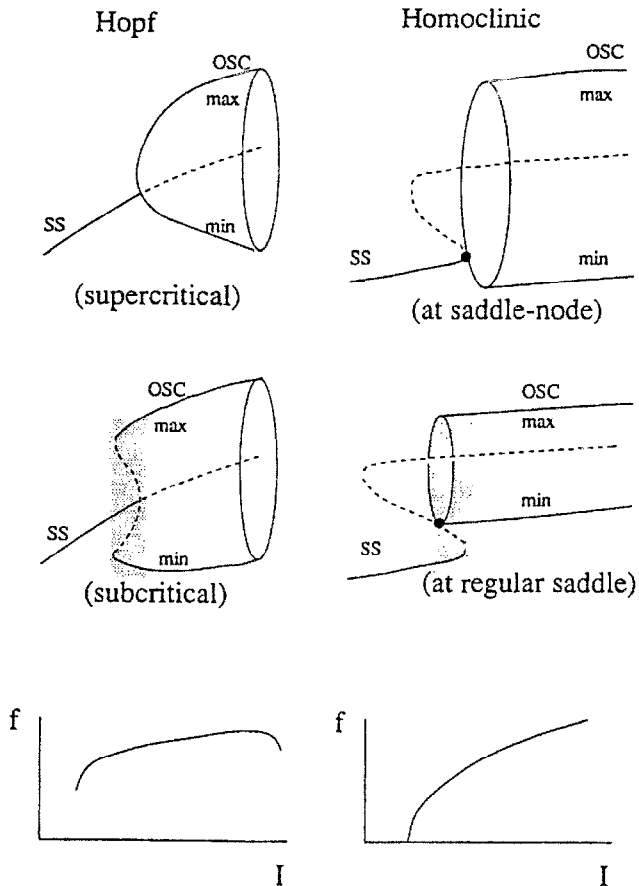


Figure 1. Schematic bifurcation diagrams from a steady state (SS) to an oscillatory firing state (OSC). The abscissa is a control parameter, such as the applied current intensity. The ordinate corresponds to the membrane potential, the repetitive firing state being indicated by the maximal and minimal amplitudes of the oscillatory membrane potential. The solid curve indicates stable and the dashed curve unstable. In the lowermost panels, the ordinate (f) is the frequency of repetitive firing. The left panels show Hopf bifurcation. At the onset of oscillation, the rhythmic amplitude is small and the frequency is finite. The bifurcation may be *supercritical*, where the new oscillatory branch is stable, or *subcritical*, where the new oscillatory branch is unstable and becomes stable at a turning point. The right panels show *homoclinic bifurcation*. It corresponds to the coalescence of an oscillatory state with an unstable steady state. This steady state can be either of saddle-node or saddle type. As this bifurcation point is approached, the amplitude of oscillation remains finite, while the rhythmic frequency tends to zero (the period diverging to infinity). In the case of a subcritical Hopf bifurcation or a normal homoclinic bifurcation, there is a range of parameter values where a steady-state attractor and an oscillatory attractor coexist (bistability, shaded region).

zero or zero, respectively) and might subserve an abstract basis for the distinction between the Arvanitaki-Hodgkin type II and type I axons. Our review generalizes this theoretical methodology to characterize various *bursting* oscillations in single neurons, elaborating on a qualitative classification scheme for bursting mechanisms proposed by Rinzel (1987).

Neuronal Bursting: Examples

We summarize some qualitative features of observed bursting patterns and then relate these to our classification scheme. We briefly mention conductance mechanisms that are *sufficient* to produce

some of these bursting oscillations. Although network synaptic interactions and dendritic cable properties influence bursting behavior, for the most part our discussion concerns an isolated, isopotential neuron. The main biophysical idea is that rhythmicity is generated by a depolarization process that is autocatalytic (positive feedback), followed by a *slower* repolarization process (negative feedback). These opposing processes may involve activation and slow inactivation of an inward ionic current, or a fast inward current and a slower outward current. Such features underlie action potential generation, and for bursting, there is at least another, *slower* negative feedback process.

The burst pattern of Figure 2A has a *square-wave* form, with abrupt periodic switching between rest (silent phase) and depolarized repetitive firing (active phase). Spiking here is due primarily to a *high-threshold* fast calcium current and a Hodgkin-Huxley-like potassium current. A minimal biophysical mechanism for square-wave bursting involves a calcium-activated potassium current, as originally proposed by Chay and Keizer in 1983. During the active phase, each calcium spike slightly increases $[Ca^{2+}]_i$, slowly turning on this current and eventually repolarizing the membrane to terminate the active phase. During the silent phase, the Ca^{2+} channels are closed, $[Ca^{2+}]_i$ decreases, and, as the potassium conductance deactivates, the cell slowly depolarizes until threshold for the next active phase is reached. Suggested alternative mechanisms for this type of bursting include slow inactivation by Ca^{2+} or by voltage of the Ca^{2+} current. Here, if spikes are abolished by pharmacologically blocking the calcium current, bursting is lost.

Although the dopamine-secreting neuron (Figure 2B) superficially appears to be a square-wave burster, we would not classify it as such. Its underlying slow wave persists even when action potentials are blocked. It appears to be of dendritic origin, and it drives somatic spiking via electrotonic interaction.

The bursting patterns of Figure 2C and 2D exhibit brief spike bursts riding on a slow *triangular wave*. Thalamocortical relay cells (Figure 2C) burst at the delta wave frequency (3 Hz) of quiet sleep (Steriade, McCormick, and Sejnowski, 1993), while the 5-Hz oscillation in inferior olivary cells (Figure 2D) is probably involved with movement tremor (see Llinás, 1988). Remarkably, in both cases, rhythmic bursting occurs for maintained hyperpolarizing rather than depolarizing stimuli. The underlying slow wave (due to a low-threshold calcium current) is unmasked when the fast action potentials are blocked, and is sometimes seen for modest hyperpolarizing inputs, even without blocking spikes. The Ca^{2+} current activates rapidly, below the voltage threshold for action potentials. Its inactivation by voltage, with a time scale like that of the triangular wave's depolarization, provides the slow negative feedback.

The *Aplysia* R15 neuron is the quintessential experimental model of an endogenous burster (Figure 2E). The sodium spike rate during a burst first increases and then decreases; hence there is *parabolic* bursting. Blocking these spikes reveals an underlying quasi-sinusoidal slow wave that is generated primarily by a Ca^{2+} current. This current activates more slowly and at lower depolarizations than that associated with the square-wave bursting of Figure 2A. Its slow activation and the slower $[Ca^{2+}]_i$ that inactivates it provide the two variables for a minimal model of a parabolic burster's underlying slow oscillator (see the next section).

Parabolic burst-like features are seen in the 10-Hz oscillations of mammalian thalamic reticular neurons (Figure 2F) during the spindle waves of quiet sleep (Steriade et al., 1993). The oscillation depends on a low-threshold calcium current, like that of triangular bursting. In addition to this current's slow inactivation, there is likely a second slow variable to support the parabolic pattern, e.g., $[Ca^{2+}]_i$ for activating a calcium-dependent potassium current in these cells.

A quite different kind of burst pattern consists of spike clusters interspersed with epochs of small-amplitude subthreshold oscilla-

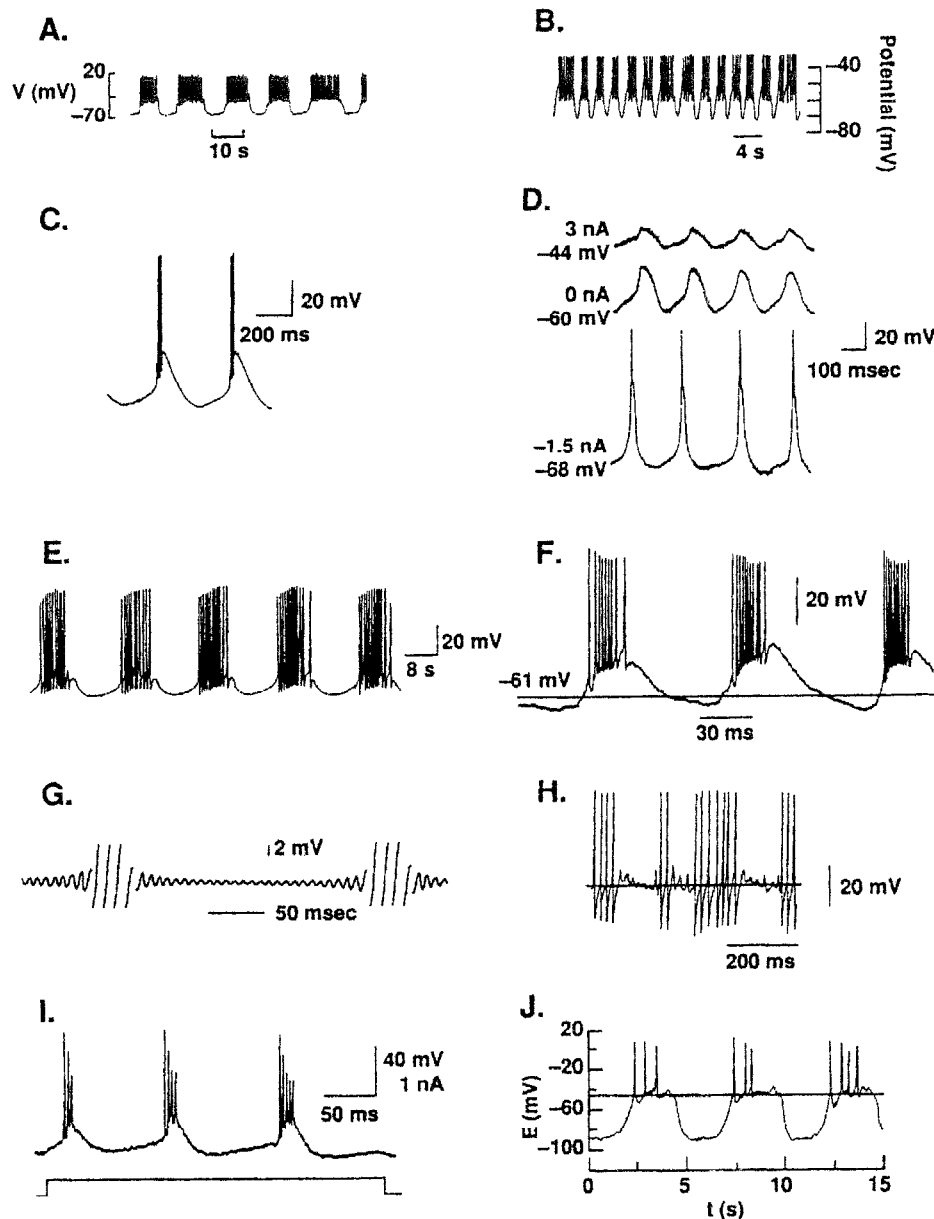


Figure 2. Examples of rhythmic bursting, showing the time courses of membrane potential, with the exception of *G*, which is extracellular voltage. See text for explanations. *A*, Pancreatic β -cell (From Sherman, A., Carroll, P., Santos, R. M., and Atwater, I., 1990, Glucose dose response of pancreatic beta-cells: Experimental and theoretical results, in *Transduction in Biological Systems* (C. Hidalgo et al. Eds.), New York: Plenum, p. 123; reprinted with permission.) *B*, Dopamine-containing neurons in the rat midbrain. (From Johnson, S. W., Seutin, V., and North, R. A., 1992, Burst firing in dopamine neurons induced by *N*-methyl-D-aspartate: Role of electrogenic sodium pump, *Science*, 258:665–667; reprinted with permission. Copyright 1992 by the AAAS.) *C*, Cat thalamocortical relay neuron. (From McCormick, D. A., and Pape, C.-H., 1991, Properties of a hyperpolarization-activated cation current and its role in rhythmic oscillation in thalamic relay neurons, *J. Physiol. Camb.*, 431:291–318; reprinted with permission.) *D*, Guinea pig inferior olivary neuron. (From Benardo, L., and Foster, R. E., 1986, Oscillatory behaviors in inferior olive neurons: Mechanism, modulation, cell aggregates, *Brain Res. Bull.*, 17:773–784; copyright 1986; reprinted with permission from Elsevier Science Ltd.) *E*, *Aplysia* R15 neuron. (From Lotshaw, D. P., Levitan, E. S., and Levitan, I. B., 1986, Fine tuning of neuronal electrical activity: Modulation of several ion channels by in-

tracellular messengers in a single identified nerve cell, *J. Exp. Biol.*, 124:302–322; reprinted with permission of Company of Biologists Ltd.) *F*, Cat thalamic reticular neuron. (From Mulle, C., Madariaga, A., and Deschênes, M., 1986, Morphology and electrophysiological properties of reticularis thalami neurons in cat: In vivo study of a thalamic pacemaker, *J. Neurosci.*, 6:2134–2145; reprinted with permission of the Society for Neuroscience.) *G*, *Sepia* giant axon. (From Arvanitaki, A., 1939, Recherche sur la réponse oscillatoire locale de l'axone géant isolé de *Sepia*, *Arch. Int. Physiol.*, 49:209–256; reprinted with permission.) *H*, Rat thalamic reticular neuron. (From Pinault, D., and Deschênes, M., 1992, Voltage-dependent 40 Hz oscillations in rat reticular thalamic neurons in vivo, *Neuroscience*, 51:245–258; copyright 1992; reprinted with permission from Elsevier Science Ltd.) *I*, Mouse neocortical pyramidal neuron. (From Agmon, A., and Connors, B. W., 1989, Repetitive burst-firing neurons in the deep layers of mouse somatosensory cortex, *Neurosci. Lett.*, 99:137–141; reprinted with permission.) *J*, Rat pituitary gonadotropin-releasing cell. (From Tse, A., and Hille, B., 1993, Role of voltage-gated Na^+ and Ca^{2+} channels in gonadotropin-releasing hormone-induced membrane potential changes in identified rat gonadotropes, *Endocrinology*, 132(4):1475–1481; reprinted with permission. © The Endocrine Society.)

tions (Figures 2G and 2H). The envelope of fast events slowly waxes and wanes, forming an approximate spindle or ellipse; hence the term, *elliptic bursting*. Here, the inactive phase is not totally silent but often shows small oscillations. The frequency of intraburst spiking is comparable to that of the interburst subthreshold oscillations. Only recently has this bursting pattern been reported for mammalian neurons and associated with important functional roles, such as the limbic system's theta rhythm (not shown), and the gamma fast oscillations (about 40 Hz) that occur intermittently with increased alertness and focused attention (Figure 2H). Experimental (Llinás, Grace, and Yarom, 1991) and computational (Wang, 1993) studies indicate that the 40-Hz elliptic bursts involve a persistent Na^+ conductance and a specific voltage-dependent transient K^+ conductance.

Some oscillations (Figures 2I and 2J) depend on the electrical cable properties of neuronal dendrites and intracellular sources of regenerative ion fluxes. The bursting behavior of some pyramidal neurons (Figure 2I) in neocortex (Connors and Gutnick, 1990) and in the hippocampus depends on high-threshold calcium channels located on the distal dendrites, while the faster sodium spikes are generated primarily in the perisomatic region (Williams and Stuart, 1999). Computer simulations suggest that a one-compartment description is inadequate, and that electrotonically distinct compartments must be explicitly modeled and analyzed (Traub et al., 1991; Pinsky and Rinzel, 1994). Figure 2J displays the bursting pattern of a pituitary gonadotropin-releasing cell. Although it resembles the square-wave form of Figure 2A, here the underlying slow rhythm is generated by a cytoplasmic second messenger system that leads to nonlinear, time-dependent calcium fluxes across the endoplasmic reticulum (ER) membrane and to oscillations in $[\text{Ca}^{2+}]_i$.

Bursting Systems Analysis: Fast/Slow Phase Space Dynamics

Since different bursters may have qualitatively similar patterns, a qualitative classification should not depend on quantitative properties such as the rhythm's period or its precise biophysical bases. Our general framework involves a *geometrical* analysis of the bursting dynamics for a model's differential equations (Rinzel, 1987; Rinzel and Ermentrout, 1998; Izhikevich, 2000). The model for an isopotential neuron may be written as:

$$\frac{dX}{dt} = F(X, Y) \quad (1)$$

$$\frac{dY}{dt} = G(X, Y) \quad (2)$$

where the vectors X and Y represent the variables with fast and slow time scales, respectively. Typically, the membrane potential is a fast variable, so Equation 1 might be the membrane's current balance equation:

$$C_m \frac{dV}{dt} = - \sum_i I_i + I_{\text{app}}$$

The other dynamic variables include the gating variables for specific ionic channels plus relevant second-messenger variables and ionic concentrations. Here we consider only one or two slow variables Y_k , which might be a slow voltage-dependent gating variable or $[\text{Ca}^{2+}]_i$, or both.

The fast/slow phase space dissection method (Rinzel, 1987; Rinzel and Ermentrout, 1998; Izhikevich, 2000) exploits the presence of two disparate time scales. For simplicity, suppose there is only one slow variable, Y . One first treats Y as a *control parameter* and considers the dynamics of Equation 1 as a function of Y . The fast

subsystem's various behavioral states are then summarized in a bifurcation diagram, plotting response amplitude, say V , versus Y , as in Figure 1, but where Y (instead of I) is the parameter. When the full system is considered, Y evolves slowly in time according to Equation 2, slowly sweeping through a range of values, while the fast subsystem slowly tracks its stable states (*attractors*). For example, an oscillatory state of the fast subsystem corresponds to the repetitive firing of a burst's active phase. During a silent phase, the fast subsystem would be following a pseudo-steady state of hyperpolarized V . To complete the description, one must understand the slow dynamics from Equation 2 in order to know where on the fast subsystem's bifurcation diagram Y will be increasing or decreasing. When the full system, Equations 1 and 2, is integrated and the resulting burst trajectory is projected onto the (V, Y) plane, it coincides with portions of the bifurcation diagram. Through visualization of this geometrical representation, one can make predictions about the qualitative behavior of bursting and the effects of various parameter changes.

1. *Square-wave bursting*. The prototypical fast/slow phase plane (Figure 3A) was originally developed for the Chay-Keizer model of β -cell bursting, where $[\text{Ca}^{2+}]_i$ was the slow, negative-feedback variable. For the fast/slow dissection, one first constructs the fast subsystem's bifurcation diagram by treating Y as a parameter. This yields the Z-shaped curve of steady states. The oscillatory state "surrounding" the upper branch corresponds to repetitive spiking of an active phase. It terminates by contacting the unstable middle steady-state branch, at a homoclinic bifurcation. The Z-curve's lower branch represents a stable steady state of hyperpolarization, as tracked during a burst's silent phase. In an intermediate range of Y values, there is bistability of the depolarized oscillation and the hyperpolarized steady state.

Next, Y is allowed to vary according to its kinetics. Bursting occurs if the slow kinetics dictate that Y increases (decreases) when the fast spike-generating subsystem is in its upper (lower) state, where the voltage-dependent channels are (are not) activated. The slow Y modulation induces abrupt switching between the two co-existing states and thus temporal alternation between a train of spikes and a resting phase, as seen in Figure 2A.

2. *Triangular bursting*. Figure 3B shows fast/slow phase planes associated with triangular bursting. A minimal model has one slow variable, and its fast subsystem has regimes of bistability, as in the square-wave case. Here, however, the steady-state curve has five branches, composed of two S-shaped portions in different V ranges. These S-curves correspond to the two sets of regenerative currents active in the subthreshold voltage ranges, such as in thalamic relay or inferior olivary cells. The depolarized oscillatory state (repetitive spiking) joins the middle steady-state branch at its right knee (a saddle-node homoclinic bifurcation). Different oscillation patterns occur, depending on whether the lower S's right knee extends rightward beyond that of the upper S's right knee. If this is not the case (Figure 3B, left), a slow subthreshold oscillation without fast spikes may occur. The alternative case (Figure 3B, right) corresponds to more intense hyperpolarizing input, when triangular bursting arises (Figure 2D). "Triangular" refers to the gradually falling V time course of the active phase, related to the middle branch's steep slope (Figure 3B).

3. *Parabolic bursting*. This bursting type has a smooth underlying slow subthreshold wave. Its generation requires at least two slow variables, one for positive feedback and the other for negative feedback. The minimal fast/slow phase plot has three dimensions: V and the two slow variables (Figure 3C, originally constructed for a model of the *Aplysia* R15 neuron; see Rinzel, 1987). Steady states of the fast subsystem are now represented by a *Z-surface*. Similarly, a surface describes the fast oscillatory (repetitive spiking) attractors. These periodic solutions disappear via homoclinic bifurcation as they contact the Z-surface, precisely at its lower knee, a saddle-

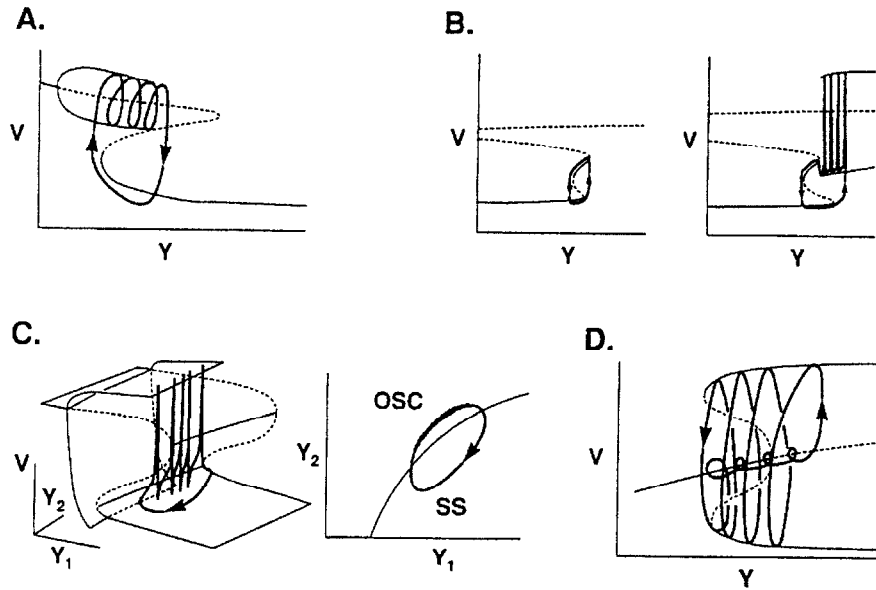


Figure 3. Fast- and slow-phase plot of bursting dynamics. The variable Y is a slow variable (there are two slow variables, Y_1 and Y_2 , in part C). In each case the bifurcation diagram is computed for the fast subsystem, with Y treated as a parameter and plotted in terms of the membrane potential (V) behavior as a function of Y . The solid curve shows stable and the dashed curve unstable branches. The oscillatory state of repetitive firing is represented by its maximum and minimum of V (cf. Figure 1). The heavy curves with arrows are bursting trajectories of the full system plotted on the (V, Y) plane or the (V, Y_1, Y_2) space. **A**, Square-wave bursting is based on a bistability of a steady state and a repetitive firing state in the fast subsystem and periodic switching between the two, induced by the slow-variable dynamics. **B**, Triangular bursting has a similar phase plot as in part A, but the fast subsystem's steady-state curve is quintic rather than cubic, with two branches of stable steady states. Depending on whether the stable repetitive

firing state overlaps with the lowermost steady-state branch, oscillations of the full system may be either purely subthreshold (left panel) or bursting (right panel). For simplicity, the repetitive firing state is shown only on the right panel, not on the left panel. **C**, Parabolic bursting is generated by an oscillation in a two-variable (Y_1 and Y_2) slow subsystem (right panel) that induces smooth periodic switching between a steady state (SS) and a repetitive spiking state (OSC) (which do not overlap) of the fast subsystem. **D**, Elliptic bursting involves a subcritical Hopf bifurcation in the fast subsystem. Bursting involves slow switching between a steady state and a repetitive firing state that are bistable in the fast subsystem. The silent phase exhibits damped or growing small oscillations as its trajectory passes through the Hopf bifurcation point. (Parts A and C–D are adapted from Rinzel, 1987; Part B from Rush, M., and Rinzel J., 1994, *Biol. Cybern.*, 71:281–291.)

node coalescence (cf. Figure 1). Here, the fast subsystem is monostable. The slow-variable phase plane is divided into two non-overlapping regions: one for the resting steady state and the other for the repetitive spiking regime of the fast subsystem.

Now, when the slow variables are allowed to vary, an oscillation may occur in this two-variable slow system (Figure 3C, right). If the slow oscillatory trajectory visits both of the fast subsystem's regimes, bursting occurs, with repetitive smooth switching between the resting and spiking states. As a burst begins and ends, its trajectory crosses a homoclinic bifurcation of the fast subsystem and spike frequency drops dramatically; hence the parabolic nature.

4. Elliptic bursting. A minimal model has only one slow variable (Figure 3D, originally constructed for a modified FitzHugh-type model; see Rinzel, 1987). The fast subsystem has bistability because of a *subcritical* Hopf bifurcation (cf. Figure 1) of periodic solutions from a monotonic steady-state curve. As in the square-wave case, during bursting the full system operates in the (V, Y) regime of bistability, repetitively switching between the steady state and the spiking state. A distinguishing feature, however, is that the “silent phase,” when the fast subsystem operates near its steady state, is no longer truly silent: it can display small oscillations that damp and then grow as the trajectory slowly passes through the Hopf bifurcation point, where the steady state is a spiral-type fixed point.

5. Complex bursting. The theoretical study of certain bursting types (Figures 2I and 2J) is relatively recent, and mathematical understanding of their mechanisms is just emerging. For analyzing the case of Figure 2I, one requires a minimal model of at least two

electrotonically separated compartments (Pinsky and Rinzel, 1994). Such a model can still be analyzed by the fast/slow dissection method, which shows that complex bursting of neocortical pyramidal cells can be described as the square-wave type (Kepecs and Wang, 2000). As for Figure 2J, one must take into account the interaction between second-messenger-mediated calcium fluxes from intracellular pools and voltage-dependent plasma membrane calcium currents.

The classification discussed here is based on various fast/slow phase plots. Although consistent with some of the wave-form phenomenology, the two may sometimes disagree. For instance, a system with the fast/slow phase plot of Figure 3C may burst with a slow wave that is less sinusoidal and more rectangular if one slow variable is much slower than the other. However, in contrast to a square-wave burster (Figures 2A and 3A), its slow wave may persist, even with the fast action potentials blocked.

Discussion

We have reviewed various neuronal bursting oscillations and, by using notions and analytic tools from the mathematics of dynamical systems, we discussed how these bursting patterns might be theoretically described and classified. Our examples are minimal for these categories. Indeed, one can imagine subcategories based on differences in the fast subsystem's bifurcation diagram. In summary, bursting in a single-compartment model typically involves some slow processes that induce repetitive switching between a relatively quiescent state and an active state of repetitive spiking

of a faster system. In the cases of square-wave, triangular, and elliptic bursting, one slow variable is sufficient, and the fast subsystem must be bistable. In the parabolic bursting case, the fast subsystem need not display bistability, and two slow variables are required.

The geometrical analysis by fast/slow dissection illustrates how novel and powerful theoretical approaches can emerge from fruitful interactions between neurobiology and the science of dynamical systems. Possible extensions might consider cable-like distributed systems with local burst-generating dynamics, or systems with many slow variables, or systems with complicated bifurcation diagrams, perhaps involving chaotic attractors. One can expect that dynamical systems methods, including fast/slow dissection, may also play a role in our understanding of neural networks with many synaptically coupled neurons, as long as there are disparate time scales in the system (Tabak et al., 2000).

Road Map: Biological Neurons and Synapses

Background: Dynamics and Bifurcation in Neural Nets

Related Reading: Hippocampal Rhythm Generation; Neuromodulation in Mammalian Nervous Systems; Sleep Oscillations

References

- Arvanitaki, A., 1939, *Les Variations graduées de la polarisation des systèmes excitables*, Paris: Hermann.
- Chay, T. R., and Keizer, J., 1983, Minimal model for membrane oscillations in the pancreatic beta-cell, *Biophys. J.*, 42:181–190.
- Connors, B. W., and Gutnick, M. J., 1990, Intrinsic firing patterns of diverse neocortical neurons, *Trends Neurosci.*, 13:99–104. ♦
- FitzHugh, R., 1961, Impulses and physiological states in models of nerve membrane, *Biophys. J.*, 1:445–466.
- Izhikevich, E. M., 2000, Neural excitability, spiking, and bursting, *Int. J. Bifurcat. Chaos*, 10:1171–1266. ♦
- Kepecs, A., and Wang, X.-J., 2000, Analysis of complex bursting in cortical pyramidal neuron models, *Neurocomputing*, 32:181–187.
- Llinás, R., 1988, The intrinsic electrophysiological properties of mammalian neurons: Insights into central nervous system function, *Science*, 242:1654–1664. ♦
- Llinás, R. R., Grace T., and Yarom, Y., 1991, *In vitro* neurons in mammalian cortical layer 4 exhibit intrinsic oscillatory activity in the 10- to 50-Hz frequency range, *Proc. Natl. Acad. Sci. USA*, 88:897–901.
- Pinsky, P., and Rinzel, J., 1994, Intrinsic and network rhythmogenesis in a reduced Traub model for CA3 neurons, *J. Computat. Neurosci.*, 1:39–60.
- Rinzel, J., 1987, A formal classification of bursting mechanisms in excitable systems, in *Proceedings of an International Congress of Mathematicians* (A. M. Gleason, Ed.), Providence, RI: American Mathematical Society, pp. 1578–1594.
- Rinzel, J., and Ermentrout, G. B., 1998, Analysis of neural excitability and oscillations, in *Methods in Neuronal Modeling: From Ions to Networks*, 2nd ed. (C. Koch and I. Segev, Eds.), Cambridge, MA: MIT Press, pp. 251–291. ♦
- Steriade, M., McCormick, D. A., and Sejnowski, T. J., 1993, Thalamocortical oscillations in the sleep and aroused brain, *Science*, 262:679–685. ♦
- Tabak J., Senn W., O'Donovan, M. J., and Rinzel, J., 2000, Modeling of spontaneous activity in developing spinal cord using activity-dependent depression in an excitatory network, *J. Neurosci.*, 20:3041–3056.
- Traub, R., Wong, R., Miles, R., and Michelson, H., 1991, A model of a CA3 hippocampal pyramidal neuron incorporating voltage-clamp data on intrinsic conductances, *J. Neurophysiol.*, 66:635–649.
- Wang, X.-J., 1993, Ionic basis for intrinsic 40 Hz neuronal oscillations, *NeuroReport*, 5:221–224.
- Williams, S. R., and Stuart, G. J., 1999, Mechanisms and consequences of action potential burst firing in rat neocortical pyramidal neurons, *J. Physiol.*, 521:467–482.

PAC Learning and Neural Networks

Martin Anthony and Norman Biggs

Introduction

In this article, we discuss the “probably approximately correct” (PAC) learning paradigm as it applies to artificial neural networks. The PAC learning model is a probabilistic framework for the study of learning and generalization. It is useful not only for neural classification problems but also for learning problems more often associated with mainstream artificial intelligence, such as the inference of Boolean functions. In PAC theory, the notion of successful learning is formally defined using probability theory. Very roughly speaking, if a large enough sample of randomly drawn training examples is presented, then it should be likely that, after learning, the neural network will classify most other randomly drawn examples correctly. The PAC model formalizes the terms “likely” and “most.” Furthermore, the learning algorithm must be expected to act quickly, since otherwise it may be of little use in practice.

There are thus two main emphases in PAC learning theory. First, there is the issue of how many training examples should be presented. Second, there is the question of whether learning can be achieved using a fast algorithm. These are known, respectively, as the *sample complexity* and *computational complexity* problems. This article provides a brief introduction to these problems. We highlight the importance of the Vapnik-Chervonenkis dimension, a combinatorial parameter that measures the expressive power of a neural network, and describe how this parameter quantifies fairly precisely the sample complexity of PAC learning. In discussing the computational complexity of PAC learning, we shall present a re-

sult that illustrates that in some cases, the problem of PAC learning is inherently intractable.

There are many variations on the basic PAC model that is the topic of this article, but there is insufficient space to explore these variations here. However, our discussion of the basic model serves as an introduction to the considerations that form the basis of recent extensions.

PAC Learning

Basic Definitions

In this section, we describe the basic PAC model of learning introduced by Valiant (1984). This model is applicable to neural networks with one output unit that outputs either the value 0 or 1; thus, it applies to *classification* problems. In the PAC model, it is assumed that the neural network receives a sequence of *examples* x , each labeled with the value $t(x)$ of the particular *target function* that is being “learned.” A fundamental assumption of this model is that these examples are presented independently and at random according to some fixed (but unknown) probability distribution on the set of all examples.

We first explain how to formalize the notion of generalization. Suppose that the set of all possible examples is $X = \mathbb{R}^n$ or $X = \{0, 1\}^n$, where n is the number of inputs to the network, and that the target function t can be computed by the neural network in

some state. A *training sample* for t of length m is an element \mathbf{s} of $(X \times \{0, 1\})^m$, of the form

$$\mathbf{s} = ((x_1, t(x_1)), (x_2, t(x_2)), \dots, (x_m, t(x_m)))$$

We shall denote by $S(m, t)$ the set of all training samples of length m for t . The learning algorithm accepts the training sample \mathbf{s} and alters the state of the network in some way in response to the information provided by the sample. It is desired that the function computed by the network when in the resulting state be an approximation to the target function.

Probability and Approximation

If $L(\mathbf{s})$ is the function computed by the network after training sample $\mathbf{s} \in S(m, t)$ has been presented and learning algorithm L has been applied, one way in which to assess the success of the learning process is to measure how close $L(\mathbf{s})$ is to t . Since there is assumed to be some probability distribution P on the set of all examples, and since t takes only the values 0 or 1, we may define the *error*, $er_P(h, t)$, of a function h (with respect to t) to be the P probability that a randomly chosen example is classified incorrectly by h . In other words,

$$er_P(h, t) = P(\{x \in X : h(x) \neq t(x)\})$$

The aim is to ensure that the error of $L(\mathbf{s})$ is “usually small.” Since each of the m examples in the training sample is drawn randomly and independently according to P , the sample vector \mathbf{x} is drawn randomly from X^m according to the product probability distribution P^m . Thus, more formally, we want it to be true that with high P^m probability, the sample \mathbf{s} arising from \mathbf{x} is such that the function $L(\mathbf{s})$ computed after training has small error with respect to t . This leads us to the following formal definition of PAC learning.

The learning algorithm L is a *PAC learning algorithm* for the network if for any given $\delta, \epsilon > 0$ there is a sample length $m_0(\delta, \epsilon)$ such that for all target functions t computable by the network and for all probability distributions P on the set of examples, we have

$$m \geq m_0(\delta, \epsilon) \Rightarrow P^m(\{\mathbf{s} \in S(m, t) : er_P(L(\mathbf{s}), t) > \epsilon\}) < \delta$$

In other words, provided the sample has length at least $m_0(\delta, \epsilon)$, then it is “probably” the case that after training on that sample, the function computed by the network is “approximately” correct. (We should note that the product probability distribution P^m is really defined not on subsets of $S(m, t)$ but on sets of vectors $\mathbf{x} \in X^m$. However, this abuse of notation is convenient and is unambiguous: for a fixed t , there is a clear one-to-one correspondence between vectors $\mathbf{x} \in X^m$ and training samples $\mathbf{s} \in S(m, t)$.) Note that the probability distribution P occurs twice in the definition: first in the requirement that the P^m probability of a sample be small, and again in the fact that the error of $L(\mathbf{s})$ is measured with reference to P . The crucial feature of the definition is that we require that the sample length $m_0(\delta, \epsilon)$ be independent of P and of t . It is not immediately clear that this is possible, but the following informal arguments explain why it can be done. If a particular example has not been seen in a large sample \mathbf{s} , the chances are that this example has low probability (with respect to P), and therefore misclassification of that example contributes little to the error of the function $L(\mathbf{s})$. In other words, the penalty paid for misclassification of a particular example is its probability, and, very loosely speaking, the two occurrences of the probability distribution in the definition can therefore “balance” or “cancel” each other.

The Finite Case

We shall show that if the network computes only a finite number of functions (for example, when the weights of a neural network

are restricted to a finite set of allowed values), then there is a PAC learning algorithm for the network.

We say that the learning algorithm L is *consistent* if, given any training sample $\mathbf{s} = ((x_1, t(x_1)), (x_2, t(x_2)), \dots, (x_m, t(x_m)))$, the functions $L(\mathbf{s})$ and t agree on x_i , for each i between 1 and m . Such a condition seems quite natural. We should note, however, that neither the standard on-line perceptron learning algorithm nor the on-line backpropagation algorithm is, in general, consistent. But the batch versions of these algorithms, in which one repeatedly cycles through the training sample until no further changes are required, are consistent algorithms.

Suppose that the network is capable of computing a total of M different functions, and let t be any one of these. If h is computable by the network and has error $\epsilon_h \geq \epsilon$ with respect to t and P , then the probability (with respect to the product distribution P^m) that h agrees with t on a random sample is clearly at most $(1 - \epsilon_h)^m$. This is at most $\exp(-\epsilon_h m)$, using a standard approximation. Thus, since there are certainly at most M such functions h , the probability that *some* function computable by the network has error at least ϵ and is consistent with a randomly chosen sample \mathbf{s} is at most $M \exp(-\epsilon m)$. Here we have used the “union bound” (see LEARNING AND GENERALIZATION: THEORETICAL BOUNDS). For any fixed positive δ , this probability is less than δ , provided that

$$m \geq m_0(\delta, \epsilon) = \frac{1}{\epsilon} \log \left(\frac{M}{\delta} \right)$$

This bound is independent of both the distribution and the target function.

This analysis shows that if a network computes only a finite number of functions, then there is a PAC learning algorithm for the network; moreover, *any* consistent learning algorithm for the network is a PAC learning algorithm. The argument fails if the network in question computes infinitely many functions, and it is not immediately clear that PAC learning is possible in such circumstances. In the next section, we present a theory that shows that, in many such cases, it is possible.

PAC Learning and the Vapnik-Chervonenkis Dimension

The Vapnik-Chervonenkis Dimension

In this section, we show how the problem of PAC learning can be addressed by means of a combinatorial parameter known as the Vapnik-Chervonenkis dimension (henceforth called the VC-dimension) (see also VAPNIK-CHERVONENKIS DIMENSION OF NEURAL NETWORKS). Suppose \mathcal{N} is a neural network that outputs 0 or 1, and suppose that \mathcal{N} accepts examples from a set X (for example, $X = \mathbb{R}^n$, where n is the number of inputs). We say that a set T of examples is *shattered* by \mathcal{N} if for each of the $2^{|T|}$ possible ways of dividing T into two disjoint sets T_1 and T_0 , there is *some* function f computable by \mathcal{N} such that $f(x) = 1$ if $x \in T_1$ and $f(x) = 0$ if $x \in T_0$. In what follows, it is sometimes convenient to say that x is a positive (respectively, negative) example of f if $f(x) = 1$ (respectively $f(x) = 0$). The *VC-dimension* of \mathcal{N} , denoted $\text{VCdim}(\mathcal{N})$, is defined to be the largest size of a set of examples shattered by \mathcal{N} . The VC-dimension may be thought of as a measure of the “expressive power” of the network, although Vapnik and Chervonenkis (1971) defined this parameter in a more general context and not specifically in the context of neural networks. It should be noted that the notion of Vapnik-Chervonenkis dimension is, in a sense, an extension to that of linear (or vector-space) dimension. Dudley (1978) proved that if \mathcal{F} is a vector space of real functions defined on a set X and if, for $f \in \mathcal{F}$, we define $f_+ : X \rightarrow \{0, 1\}$ by $f_+(x) = 1 \Leftrightarrow f(x) > 0$, then the VC-dimension of $\{f_+ : f \in \mathcal{F}\}$ is the linear dimension of \mathcal{F} .

It is instructive at this stage to determine the VC-dimension of the simplest neural network, the *simple real perceptron* \mathcal{P}_n on n inputs. This network consists of n real-valued inputs, each of which is connected by a weighted connection to the single, linear threshold, output unit. (The weights can be any real numbers.) It is clear that, for functions computable by \mathcal{P}_n , the sets of positive examples and negative examples are separated by a hyperplane.

Theorem 1. For any positive integer n , let \mathcal{P}_n be the simple real perceptron with n inputs. Then

$$\text{VCdim}(\mathcal{P}_n) = n + 1$$

Proof. Let T be any set of $n + 2$ examples. It can be shown that there is a nonempty subset T_1 of T such that, if $T_0 = T \setminus T_1$, then $\text{conv}(T_1) \cap \text{conv}(T_0) \neq \emptyset$, where $\text{conv}(A)$ denotes the convex hull of A . (This follows from Radon's theorem, which may be found in Grunbaum [1967], for instance.) It follows immediately that the sets T_1 and T_0 cannot be separated by a hyperplane; in other words, there can be no function f computable by \mathcal{P}_n such that $f(x) = 1$ if $x \in T_1$ and $f(x) = 0$ if $x \in T_0$. Therefore T is not shattered and the VC-dimension of \mathcal{P}_n must be at most $n + 1$. It remains to prove the reverse inequality. Let o denote the origin of \mathbb{R}^n and, for $1 \leq i \leq n$, let e_i be the point with a 1 in the i th coordinate and all other coordinates 0. Then \mathcal{P}_n shatters the set $T = \{o, e_1, e_2, \dots, e_n\}$ of $n + 1$ examples. To see this, suppose that $T_1 \subseteq T$. For $i = 1, 2, \dots, n$, let α_i be 1 if $e_i \in T_1$ and -1 otherwise, and let θ be $-1/2$ if $o \in T_1$, $1/2$ otherwise. Then it is straightforward to verify that if h is the function computed by the perceptron when the threshold is θ and the weights are $\alpha_1, \alpha_2, \dots, \alpha_n$, then $h(x) = 1$ if $x \in T_1$ and $h(x) = 0$ if $x \in T_0$. Therefore, T is shattered and, consequently, $\text{VCdim}(\mathcal{P}_n) \geq n + 1$. \square

Finite VC-Dimension Characterizes PAC Learning

We have observed that if \mathcal{N} computes only a finite number of functions, then any consistent learning algorithm is a PAC algorithm, and a value of $m_0(\delta, \epsilon)$ involving the number of computable functions can be determined. It turns out that, as far as PAC learning is concerned, it is not the size of the set of computable functions that is crucial but the VC-dimension of the network. More precisely, we have the following key result, due to Blumer et al. (1989) and Ehrenfeucht et al. (1989).

Theorem 2. If a neural network \mathcal{N} has finite VC-dimension $d \geq 1$, then any consistent learning algorithm L for \mathcal{N} is a PAC learning algorithm. Moreover, there is a constant K such that a sufficient sample length $m_0(\delta, \epsilon)$ for any such algorithm is

$$K\epsilon^{-1}(d \ln(\epsilon^{-1}) + \ln(\delta^{-1}))$$

On the other hand, there is a constant c such that for any PAC learning algorithm for \mathcal{N} , the sufficient sample length $m_0(\delta, \epsilon)$ must be at least $c\epsilon^{-1}(d + \ln(\delta^{-1}))$, for all $\epsilon \leq 1/8$ and $\delta \leq 1/100$.

In fact, an analogue of Theorem 2 holds for general classes of $\{0, 1\}$ -valued functions, and not simply those computable by neural networks.

VC-Dimension of Neural Networks

We now discuss some results on the VC-dimensions of certain types of network. A more detailed treatment of this topic may be found in VAPNIK-CHEVONENKIS DIMENSION OF NEURAL NETWORKS. First, we start with the feedforward linear threshold net-

work. The first part of the following result is due to Baum and Haussler (1989) and the second part is due to Maass (1993).

Theorem 3. There is $K > 0$ such that, if \mathcal{N} is any feedforward linear threshold network having W variable weights and thresholds and N threshold units, then $\text{VCdim}(\mathcal{N}) \leq KW \log N$. Furthermore, there is $c > 0$ such that some feedforward linear threshold networks having W weights and N threshold units have VC-dimension at least $cW \log N$; in other words, the upper bound is tight to within a constant.

Recent important work on the VC-dimension of neural networks includes that of Goldberg and Jerrum (1993) and Karpinski and MacIntyre (1995). In these papers, techniques from geometry and logic are used to study the VC-dimension of neural networks of certain types. In particular, the paper of Karpinski and MacIntyre, among other things, provides bounds on the VC-dimension of feedforward networks in which the output unit is a linear threshold and all other computational units have the standard sigmoid activation function given by $f(x) = 1/(1 + e^{-x})$. They obtain the following result.

Theorem 4. Suppose \mathcal{N} is a feedforward network with a linear threshold unit as output unit, and with the remaining N computational units having the standard sigmoid activation. If \mathcal{N} has W variable weights and thresholds, then

$$\text{VCdim}(\mathcal{N}) \leq (WN)^2 + 11WN \log_2(18WN^2)$$

The Computational Complexity of PAC Learning

Efficiency with Respect to Accuracy, Example Size, and Sample Length

Thus far, a learning algorithm has been defined as a function that maps training samples into hypotheses. We shall now be more specific about the computational effectiveness of this function. If the process of PAC learning by an algorithm L is to be of practical value, it should be possible to implement the algorithm quickly. We wish to quantify the behavior of a learning algorithm for a particular neural network architecture with respect to the size of the network. In particular, we wish to consider how the running time of the algorithm varies with the number n of inputs to the network: for a learning algorithm to be efficient, this running time should increase polynomially with n . However, there is another important consideration in any discussion of efficiency. Until now, we have regarded the accuracy parameter ϵ as fixed but arbitrary. It is clear that decreasing this parameter makes the learning task more difficult, and therefore the time taken to produce a probably approximately correct output should be constrained in some appropriate way as ϵ decreases; the appropriate condition is that the running time must be polynomial in $1/\epsilon$. Formally, we say that a learning algorithm L is *efficient with respect to accuracy ϵ , example size n and sample length m* if its running time is polynomial in the length m of the training sample and if there is a value of $m_0(\delta, \epsilon)$ sufficient for PAC learning that is polynomial in n and ϵ^{-1} .

Hardness Results

In complexity theory, two important classes of problems, RP and NP, are defined. The class RP is the class of all problems that can be solved by "randomized" algorithms in polynomial time, while NP is the class of problems that can be solved by nondeterministic Turing machines in polynomial time. (We refer the reader to the book by Cormen, Leiserson, and Rivest, 1990.) It is conjectured, and widely believed, that these classes are not the same; more precisely, it is believed that RP is a strict subset of NP. This is known

as the “RP \neq NP” conjecture. For fixed k , for each n , let \mathcal{P}_n^k be the neural network that consists of k linear threshold units, each connected to all of n inputs, the outputs of these threshold networks then being combined together by a hardwired AND gate. Thus, the network outputs 1 if and only if all k threshold units output 1. Blum and Rivest (1988) proved (essentially) the following result. (See also Anthony and Biggs, 1992.)

Theorem 5. Let \mathcal{P}_n^k be as described, where $k \geq 2$. If there is a PAC learning algorithm for \mathcal{P}_n^k that is efficient with respect to accuracy, example size, and number of inputs, then the “RP \neq NP” conjecture is false.

Thus, it is extremely unlikely that there is an efficient PAC learning algorithm for this surprisingly simple class of neural networks.

The point of this section has been to define what we might mean by an efficient algorithm, and to illustrate an approach (via computational complexity) to showing that certain learning problems are difficult. Therefore, we have presented a “negative” result. However, it should not be supposed that there are no positive results on efficient neural network learning. For example, there are efficient algorithms for perceptron learning, and also for more complex networks, including those in which the output of the network is a real number (and where we have to modify the definition of efficient learning from that given above, but in a fairly straightforward way). For instance, see Chapter 26 of Anthony and Bartlett (1999) for an efficient algorithm (due to Lee, Bartlett, and Williamson) for certain types of two-layer neural networks with linear threshold hidden units and a linear output unit.

Discussion

We have considered basic PAC learning as it applies to learning in artificial neural networks. There are two distinct aspects: the length of training sample to be used and the efficiency of learning. In other words, we have the *sample complexity* problem and the *computational complexity* problem. The Vapnik-Chervonenkis dimension of a neural network determines in a fairly precise way the length of sample sufficient for PAC learning. This dimension can, in many cases, be related to the structure of the network, as in the examples presented here. Techniques from computational complexity theory can be applied to show that in a number of cases, *efficient* algorithmic PAC learning is impossible unless the NP \neq RP conjecture is false.

There are a number of recent important extensions and generalizations of the PAC model that can be applied to artificial neural networks. For example, much attention has focused on PAC-type models of learning for networks in which the output is a real number rather than simply 0 or 1. The paper by Haussler (1992) was an important part of this development, and much work has subsequently been carried out on extending the PAC model to regression

and to classification by real-output neural networks. This involves generalized notions of VC-dimension. There is insufficient space here to do these recent developments justice, but the reader can consult the book by Anthony and Bartlett (1999) for details.

Road Map: Computability and Complexity

Related Reading: Learning and Generalization: Theoretical Bounds; Vapnik-Chervonenkis Dimension of Neural Networks

References

- Anthony, M., and Bartlett, P. L., 1999, *Neural Network Learning: Theoretical Foundations*, Cambridge, Engl.: Cambridge University Press. ♦
- Anthony, M., and Biggs, N., 1992, *Computational Learning Theory: An Introduction*, Cambridge, Engl.: Cambridge University Press. ♦
- Baum, E. B., and Haussler, D., 1989, What size net gives valid generalization? *Neural Computat.*, 1:151–160.
- Blum, A., and Rivest, R. L., 1988, Training a 3-node neural network is NP-complete, in *Proceedings of the 1988 Workshop on Computational Learning Theory*, San Mateo, CA: Morgan Kaufmann, pp. 9–18. (See also *Neural Netw.*, 1992, 5:117–127.)
- Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K., 1989, Learnability and the Vapnik-Chervonenkis dimension, *J. ACM*, 36:929–965.
- Cormen, T. H., Leiserson, C. E., and Rivest, R. L., 1990, *Introduction to Algorithms*, Cambridge, MA: MIT Press. ♦
- Dudley, R., 1978, Central limit theorems for empirical measures, *Ann. Probab.*, 6:899–929.
- Ehrenfeucht, A., Haussler, D., Kearns, M., and Valiant, L., 1989, A general lower bound on the number of examples needed for learning, *Inform. Computat.*, 82:247–261.
- Goldberg, P., and Jerrum, M., 1993, Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers, in *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory*, New York: ACM Press, pp. 361–369. (See also Goldberg, P., and Jerrum, M., 1995, Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers, *Machine Learn.*, 18(2/3):131–148.)
- Grunbaum, B., 1967, *Convex Polytopes*, London: Wiley.
- Haussler, D., 1992, Decision theoretic generalizations of the PAC model for neural net and other learning applications, *Inform. Computat.*, 100:78–150.
- Karpinski, M., and MacIntyre, A. J., 1995, Polynomial bounds for VC dimension of sigmoidal neural networks, in *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, New York: ACM Press, pp. 200–208. (See also Karpinski, M., and MacIntyre, A. J., 1997, Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks, *J. Comput. Syst. Sci.*, 54:169–176.)
- Maass, W., 1993, Bounds on the computational power and learning complexity of analog neural nets, in *Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, New York: ACM Press, pp. 335–344. (See also Maass, W., 1994, Neural nets with superlinear VC-dimension, *Neural Computat.*, 6:877–884.)
- Valiant, L. G., 1984, A theory of the learnable, *Commun. ACM*, 27:1134–1142.
- Vapnik, V. N., and Chervonenkis, A. Ya., 1971, On the uniform convergence of relative frequencies of events to their probabilities, *Theory Probab. Appl.*, 16:264–280.

Pain Networks

Marshall Devor

Introduction

Pain is an unpleasant sensory and emotional experience that arises in a conscious brain, typically in response to noxious stimuli. There is a growing consensus that the classical conception of how the

pain system works is incomplete, notably for its failure to adequately account for sensory abnormalities seen in patients with chronic pain. For example, people with nerve injury often report bizarre symptoms such as electric shock-like paroxysms, or severe burning pain in skin that is numb to the touch. In this article, I

outline a new synthesis, now emerging, that builds on the old scheme by addressing both normal and pathophysiological pain processes. There are many unsolved problems that could benefit from computational analysis but very little computational work has been done on pain to date.

The Pain System: Normal Functioning

Stimulus-Response Variability

The pain system encodes information on the intensity, location, and dynamics of strong, tissue-threatening stimuli. This sensory-discriminative function is shared with all sensory systems. Where pain differs is in the degree to which emotional-motivational and cognitive-evaluative variables can modulate the basic sensory message. The sight of blood may frighten you, but fright doesn't make red look like blue. In contrast, emotional and cognitive factors can render strong noxious stimuli painless. Consider, for example, the wounded soldier pulling his unconscious buddy out of the line of fire, or the placebo effect, in which belief that an inert pill contains analgesic ingredients is often enough to relieve pain. Even under

everyday conditions, shifts of attention and expectation cause normal, rational people to display wide variability in pain sensation. For this reason, pain professionals usually avoid speaking of pain stimuli (or pain receptors), preferring instead *noxious stimuli* (or *nociceptors*). A noxious stimulus may evoke more or less pain, depending on context; the degree of pain felt depends as much on system variables as it does on the stimulus itself (Wall, 1999).

Stimulus Encoding by Sensory Receptor Endings

The first step in sensory signaling is to encode the quality and location of the stimulus. This is done with spatial arrays of sensory receptor endings, each responsive to a specific type of stimulus at a specific location. Sensory receptors are the ends of axons of primary somatosensory neurons (primary afferents), the cell bodies of which are located near the spine (but not in the spinal cord), in the dorsal root ganglia (DRGs, Figure 1B). Each DRG neuron has a peripheral axon that travels in a nerve and terminates in a sensory transducer ending in skin, muscle, viscera, etc., and a central axon that travels in a dorsal root and ends synaptically in the spinal cord and/or the brainstem. The cell body itself is offset from the main

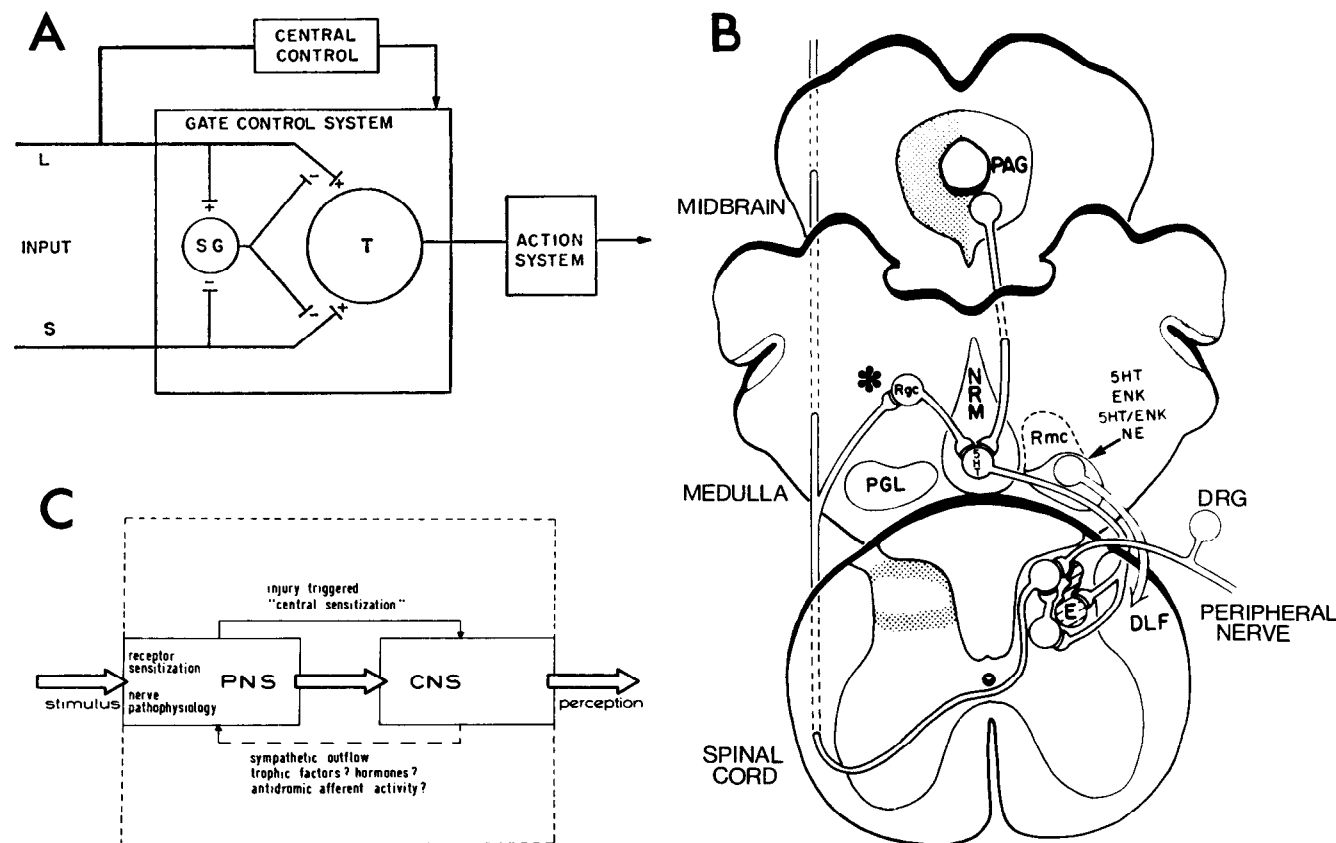


Figure 1. Three circuits for the modulation of pain signals. *A*, The original gate control system of Melzack and Wall (1965). Input from low-threshold (*L*) afferents and nociceptors (*S*) activates WDR transmission neurons (*T*) in the dorsal horn of the spinal cord. The former, but not the latter, activate substantia gelatinosa (*SG*) interneurons, which presynaptically inhibit the nociceptive input. Central control is also noted. *B*, Subsequently discovered details of the central control system (Fields and Basbaum, 1999). Activity in the midbrain PAG, relayed through specific medullary nuclei, including the nucleus raphe magnus (*NRM*) and the reticular magnocellular nucleus (*Rmc*), evokes synaptic inhibition on ascending pain-signaling neurons in the spinal cord partly via enkephalinergic (*E*) spinal interneurons. The

descending axons of the *NRM* and *Rmc*, which use as neurotransmitters 5-hydroxytryptamine (5-HT), enkephalin (ENK), and/or norepinephrine (NE), travel in the dorsolateral funiculus (DLF). A collateral branch of the ascending WDR pain-signaling neurons (*T*) contributes to descending inhibition via nucleus reticularis gigantocellularis (*Rgc*, asterisk) in a negative feedback loop. *C*, Pain amplification mechanisms associated with tissue and nerve damage. The pain signal is modulated in the PNS and CNS by the local processes noted, and by a combination of feedback and feedforward. (From Devor, M., and Seltzer, Z., 1999, Pathophysiology of damaged nerves in relation to chronic pain, in *Textbook of Pain*, 4th ed. [P. D. Wall and R. Melzack, Eds.], London: Churchill Livingstone, pp. 129–164.)

axon by a small stem. Nerve impulses run from the receptor ending, past the DRG, and on into the CNS without pause. Curiously, the cell is specially designed so that spikes will invade it. Given that there are no synapses in the DRG, why has nature gone to the trouble of arranging for this spike invasion?

Somatosensory afferents are modality specific. Some, *low-threshold mechanoreceptors* (LTMs), respond to gentle touch and vibration. Others, *nociceptors*, respond only to strong stimuli. Some nociceptors respond to a particular submodality, e.g., mechanical (pinch) or thermal (hot, cold), but most encode the combined intensity of strong mechanical, thermal, and irritant chemical stimuli. These *polymodal nociceptors* pose the dilemma of how we distinguish a pinprick from a bee sting. Some nociceptors do not respond to any stimuli at all unless they have been sensitized by tissue inflammation (*silent nociceptors*). LTMs mostly have heavily myelinated, fast-conducting A β axons. Nociceptors have slow-conducting axons either with thin myelin (A δ fibers) or with no myelin (C-fibers).

Within each modality, firing frequency encodes stimulus strength. LTMs, for example, respond to increasing pressure by accelerating their firing rate. LTMs also respond to noxious pinch. But since their firing rate saturates below the noxious range, they do not encode the intensity of noxious stimuli. Direct electrical microstimulation of LTM axons evokes a sensation of touch or pressure, but it does not (normally) evoke pain even at very high frequency (Vallbo et al., 1979). *Normally*, pain is felt only when activity is recruited in nociceptors; when stimulus strength approaches the noxious range, encoding is handed off from LTMs to nociceptors. I stress “normally” because there are system states where activity in LTMs does evoke pain (discussed in the next section).

Central Convergence

The spatial information inherent in the arrays of sensory receptor endings—in the skin, for example—is preserved by topographic (*somatotopic*) mapping of primary afferent axons onto the spinal cord, and thence onto all subsequent waystations up to the cortex. The coding of sensory quality is more complicated.

Extrapolating from the specificity of primary afferents, many investigators concluded in the past (and some still believe) that each somatosensory modality, including pain, remains separate and pathway-specific all the way to consciousness. This idea, the classical specificity theory of pain, should have been shaken by the discovery that most neurons in the spinal cord that receive synaptic input from nociceptors also receive low-threshold input; they are modality-convergent *wide dynamic range* (WDR) neurons. Indeed, some combine low-threshold and nociceptive skin input with proprioceptive and/or visceral input and hence are *multireceptive* neurons. Specificity theory was saved by the discovery of a small population of neurons in the most superficial part of the spinal dorsal horn that are (normally) nociceptive selective. But if these alone signal pain, what of the large majority of spinal WDR neurons that have convergent nociceptor input?

In fact, there is now abundant evidence that activity in WDR neurons can evoke pain sensation. But with the modality multiplexing of WDR neurons, how are the specific touch and pain sensations that we feel decoded? A number of schemes have been advanced. For example, the comparator model holds that activity in WDR neurons is interpreted as pain only if nociceptive-selective neurons are also active. Another proposal posits that stimulus quality is coded in discharge patterning. The most likely scheme, however, is that spike frequency is the key parameter. Touch is felt when WDR neurons fire slowly; pain is felt when they fire rapidly. Since each WDR neuron has its own encoding function (firing rate as a function of stimulus strength), increasingly strong stimuli pro-

gressively recruit WDR neurons with sequentially overlapping encoding functions. In this model, any given stimulus would produce a unique aggregate of neuronal activity when viewed across the entire population of neurons that map a particular patch of skin. Somatosensation is a symphony.

Ascending Pathways and Cephalic Representation

Textbook representations of ascending somatosensory pathways typically show two compact routes, the spinothalamic (anterolateral column) system for pain and temperature, and the dorsal column-medial lemniscus system for touch and vibration, each relaying through nuclei of the ventrobasal thalamus and ending mapwise in the somatosensory cortex. This vision, virtuous for being easy to teach and learn, corresponds to the ideology of specificity theory. It is also fundamentally misleading. Although some axons of the anterolateral column system reach the thalamus directly, most branch extensively, dropping terminals en route in numerous spinal, medullary, pontine, and mesencephalic structures. There are also massive projections to the cerebellum, and directly or indirectly to widespread areas of the limbic forebrain. In other words, much of the brain receives direct or nearly direct synaptic input from spinal cord WDR neurons. Ascending pain pathways are also dynamic. For example, cutting the anterolateral column usually relieves pain felt below the cut, but the pain usually returns within a few months. Any successful pain theory must take these facts into account.

Modulation and Gate Control

Peripheral Nervous System Sensitization

Sensory modulation is obvious from everyday experience. In sunburned skin, for example, gentle warming is painful. This primary hyperalgesia apparently results from peripheral sensitization, the fact that tissue inflammation can increase the gain (sensitivity) of the transduction process in nociceptive endings in skin, muscle, joints, and so forth, rendering them responsive to previously innocuous stimuli. Much of modern pain research is devoted to working out the molecular details of peripheral sensitization (Levine and Reichling, 1999; Woolf and Salter, 2000). Tissue inflammation is usually self-limiting, and hence the resulting pain is acute or subacute. Sustained inflammation, such as in rheumatoid arthritis, causes chronic pain. Most simple over-the-counter analgesics work by reducing peripheral sensitization. However, recent evidence suggests that these drugs may also have a significant CNS action.

Modulation in the CNS

Although peripheral sensitization yields pain in response to normally nonpainful stimuli (*allodynia*), it does not violate the spirit of the specificity theory in the sense that the signaling lines for the various modalities remain specific and independent. In 1965 Melzack and Wall presented the first real challenge to the specificity theory with their famous gate control theory of pain (Figure 1A). This theory began with the radical idea that pain is signaled primarily by populations of convergent WDR neurons, and added that the convergence itself is dynamic and modulated in an ongoing fashion by afferent input from the periphery, and by descending control from the brain. For example, input along low-threshold A β afferents inhibits the response of WDR neurons to simultaneously arriving nociceptive input, “closing a gate on pain.” This explains the relief obtained from gently rubbing tender skin. Central control was also a key feature of the original gate control model (Figure 1A), but the details emerged only later (Fields and Basbaum, 1999). Although ideas about the circuitry of spinal gating/modulation have changed over the years, the fundamental concept has been vindicated by a great richness of CNS modulatory processes that form the basis of the new synthesis.

Descending Inhibition

The midbrain periaqueductal gray matter (PAG in Figure 1B) is a nodal point for a descending inhibitory control circuit through which the brain gates ascending nociceptive information. Electrical stimulation of the PAG, or microinjection of opiates (e.g., morphine), activates nuclei in the medulla that contain cells rich in the neurotransmitters serotonin (5-HT; e.g., NRM in Figure 1B) and norepinephrine (NE). These cell groups in turn give rise to a compact bundle of axons that descend in the dorsolateral part of the spinal cord (DLF in Figure 1B). Activity in these descending axons inhibits the response of dorsal horn WDR neurons to noxious input, while responses to innocuous input are largely unaffected. Activation of this midbrain-medullospinal inhibitory circuit by morphine or by endogenous morphine-like neurotransmitters (enkephalin, endorphin) appears to be largely responsible for the antinociception obtained from opiates. Moreover, there is accumulating evidence that this system is also responsible for the stress-induced analgesia shown by the heroic soldier noted earlier, and for the placebo effect (Benedetti, Arduino, and Amanzio, 1999). We are talking about the neurology of belief and anticipation.

Central Sensitization

Spinal gating involves excitation and not just inhibition. For example, I noted the fact that inflammation may sensitize nociceptor endings, yielding primary hyperalgesia. However, tenderness around such injuries often spreads to a much larger area of surrounding skin where there is no inflammation and no nociceptor sensitization. There has long been evidence that such “secondary hyperalgesia” is due to impulses entering the CNS along LTM A β touch afferents. This idea of A β pain was strongly resisted, as it violates the most fundamental dogma of specificity theory, namely, that pain can only be evoked by A β and C nociceptors. However, in recent years the evidence has become increasingly compelling.

It turns out that a momentary noxious input is enough to trigger a system state, called *central sensitization*, in which A δ touch input transiently elicits pain (Raja et al., 1999; Woolf and Salter, 2000). Moreover, this state can be maintained indefinitely so long as a persistent source of nociceptive input is present (Gracely, Lynch, and Bennett, 1992). Central sensitization is particularly important as an amplifier in touch/brush-evoked pain (in contrast to thermal-evoked pain). It now appears that most of the tenderness we feel after everyday bumps and scrapes is due to A β touch afferents. A working model of central sensitization is given in Figure 2 (see legend).

Neuropathic Pain

The strongest impetus for revising classical ideas about pain came from observations of chronic pain in patients, particularly the bizarre and intractable pain associated with damage to peripheral nerves, dorsal roots, and the CNS. Such *neuropathic pain* is paradoxical in the sense that when a sensory conduction channel is compromised, sensation is expected to be reduced, not augmented. Perhaps the most striking example is phantom limb pain in amputees, but neuropathic pain also includes such common conditions as limb pain in diabetics (diabetic neuropathy), postherpetic neuralgia (shingles), sciatica, and many instances of cancer pain.

Nerve Pathophysiology/Ectopia

The key to understanding neuropathic pain is the realization that nerves do not behave like copper telephone cables. When an axon is cut across, or stripped of its myelin, the neuron reacts with path-

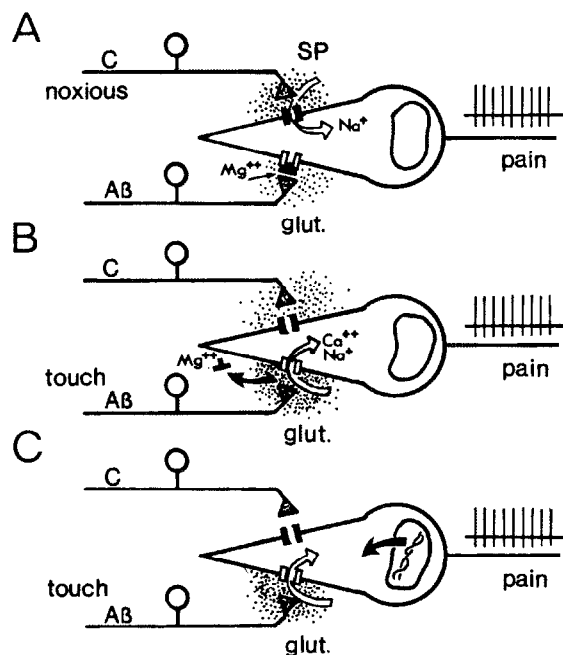


Figure 2. A proposed central sensitization mechanism for triggering touch-evoked pain (Woolf and Salter, 2000). *A*, Normally, activity in peripheral C-nociceptors activates spinal WDR neurons by means of excitatory amino acid and peptide neurotransmitters, probably including substance P (SP). This triggers an ascending pain signal. Touch input, carried along low-threshold A β afferents, evokes release of the neurotransmitter glutamate (glut). However, this drives the WDR neurons minimally because the NMDA-type (*N*-methyl *D*-aspartate) glutamate receptors on the postsynaptic dendrites are blocked at normal membrane potentials by Mg²⁺ ions. *B*, Intense noxious C-input produces prolonged (tens of seconds) SP-evoked depolarization. This displaces the Mg²⁺ block, enabling the NMDA receptors. Now, glutamate released from A β touch afferents can strongly activate the WDR neurons and hence evoke pain and tenderness. Ca²⁺ entering the WDR neurons through the enabled NMDA receptor channels may trigger phosphorylation of the channels, due to activation of a Ca²⁺-dependent protein kinase (PKA), sustaining the touch-evoked pain state for hours. *C*, More speculatively, a change in gene expression triggered by tissue or nerve injury could prolong the central sensitization state indefinitely.

ophysiological changes that may render it intrinsically resonant (Amir et al., 2002) and electrically hyperexcitable (Figure 3). The result is ongoing and stimulus-evoked firing that originates at abnormal (ectopic) sites, notably in the region of injury and/or the sensory cell body in the DRG. Other pathophysiological processes, such as nonsynaptic neuron-to-neuron coupling, may augment the ectopia (Devor and Seltzer, 1999).

Ectopic hyperexcitability appears to result primarily from injury-provoked remodeling of membrane electrical properties in the axon and/or sensory cell body, due to changes in gene expression and vectorial trafficking of expressed proteins (Devor and Seltzer, 1999; Waxman et al., 1999). Early simulations of the process indicated the potential importance of Na⁺ channel accumulation (Figure 3C), but ultimately, excitability is due to the complex integrated action on numerous channel types and subtypes (Amir et al. 2002). Ectopic firing in injured afferents contributes to pain in two ways. First, it injects an abnormal afferent impulse barrage into the CNS. Second, it may trigger and maintain central sensitization. In the sensitized state, A β touch input from the skin, and also A β activity from ectopic sources, is felt as pain (tenderness to touch and spontaneous pain).

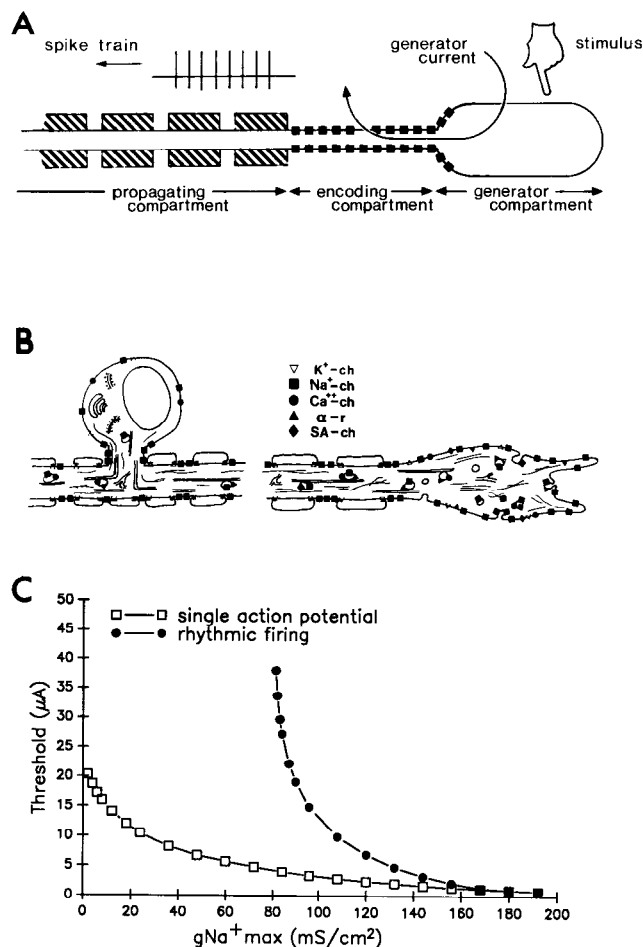


Figure 3. Stimulus transduction and encoding at normal sensory endings, and ectopic electrogenesis at sites of nerve injury, depends on the precise regulation of membrane electrical properties in the cell body and at the axon end. *A*, Applied stimuli create a generator current that is encoded into a spike train in a patch of membrane rich in Na⁺ channels (black squares in encoding compartment). *B*, The membrane channels (-ch), receptors (-r), and other proteins responsible for electrogenesis are synthesized in the cell body and transported down the axon. These include K⁺, Na⁺, and Ca²⁺ ion channels, α-adrenoreceptors (α-r), and mechanosensitive stretch-activated (SA) channels, among others. In the presence of nerve injury, channels and receptors accumulate in the membrane of the cut axon end and sprouts (right), rendering them hyperexcitable and a source of ectopia and neuropathic pain. *C*, Numerical simulation demonstrating that the accumulation of voltage-sensitive Na⁺ channels (gNa⁺max) sharply reduces the threshold for rhythmic firing, but has less of an effect on the threshold for evoking individual nerve impulses. (From Devor, M., and Seltzer, Z., 1999, Pathophysiology of damaged nerves in relation to chronic pain, in *Textbook of Pain*, 4th ed. [P. D. Wall and R. Melzack, Eds.], London: Churchill Livingstone, pp. 129–164.)

Where Is Pain?

I opened this article by characterizing pain as an unpleasant percept aroused in a conscious brain by noxious stimuli. So far I have referred to the signal acquisition apparatus of the pain system, and to neural pathways that transmit and modulate nociceptive signals. But where does the experience of pain actually occur?

First, it is safe to exclude the PNS and spinal cord, on the grounds that quadriplegics experience pain, including pain referred to anesthetic parts of the body (phantom body pain). Surprisingly,

the somatosensory cortex also appears nonessential, despite the fact that cortical neurons in a number of different regions respond to noxious stimuli (Peyron, Laurent, and Garcia-Larrea, 2000). For example, while lesions in the occipital cortex produce blindness, even extensive damage to the postcentral gyrus and other cortical projections does not preclude pain sensation, even in parts of the body whose cortical representation has been destroyed. More important, cortical seizure activity, and direct electrical stimulation of the somatosensory cortex, rarely evoke pain.

A part of the anterior cingulate gyrus has been implicated recently in pain perception on the grounds that the magnitude of activations there correlates with felt pain rather than with stimulus intensity when the pain percept is manipulated by hypnosis and distraction (Rainville et al., 1997). However, this correlation may simply reflect the operation of descending spinal modulatory pathways (Figure 1*B*) and hence attentional modulation of the ascending nociceptive signal. A final candidate is the collection of brainstem and subcortical forebrain regions that receive and modulate nociceptive signals (Devor and Zalkind, 2001). Few would argue that pain behavior, both withdrawal reflexes and more complex escape responses, may be organized subcortically. But can perception occur outside of the cortex? Might pain experience have arisen early in vertebrate evolution, in parallel with pain behavior and before cortical domination of the brain?

Perspective

Pain, particularly persistent pain, remains a medical health problem of the first order: witness the prominence of alternative approaches to treating pain, built largely on the therapeutic failures of conventional medicine. It is also a remarkable basic science challenge. Situated at the interface of body and mind, only a few synapses intervene between the biophysics of stimulus transduction and the magic of conscious perception (Devor and Zalkind, 2001).

At each level of analysis there are problems that could be fruitfully approached using computational methods. For example, in the periphery, the fine diameter of C-fiber endings precludes direct electrophysiological measurement. Testable hypotheses concerning the ionic mechanisms of transduction, encoding, sensitization (e.g., during inflammation), and ectopia could be provided by theoretical analysis. At the level of spinal processing are issues such as the modes of convergence that go into synthesizing natural receptive fields, the problem of how WDR neurons encode specific sensations, and mechanisms of functional plasticity (e.g., gate control and central sensitization). Finally, theoretical analysis might provide insights into the higher-level functions that control descending pain inhibition and ultimately pain experience. In the visual pathway, the extraction of meaning (Is it a passing cloud or a rhinoceros charging?) requires a massive analytical apparatus. In the pain system, meaning is much closer at hand.

Road Map: Other Sensory Systems

Related Reading: Emotional Circuits; Motivation; Somatosensory System; Somatotomy: Plasticity of Sensory Maps

References

- Amir, R., Liu, C.-N., Kocsis, J. D., and Devor, M., 2002, Oscillatory mechanism in primary sensory neurons, *Brain*, 125:421–435.
- Benedetti, F., Arduino, C., and Amanzio, M., 1999, Somatotopic activation of opioid systems by target-directed expectations of analgesia, *J. Neurosci.*, 19:3639–3648.
- Devor, M., and Seltzer, Z., 1999, Pathophysiology of damaged nerves in relation to chronic pain, in *Textbook of Pain*, 4th ed. (P. D. Wall and R. Melzack, Eds.), London: Churchill Livingstone, pp. 129–164. ◆
- Devor, M., and Zalkind, V., 2001, Reversible analgesia, atonia, and loss of consciousness on bilateral intracerebral microinjection of pentobarbital, *Pain*, 94:101–112.

- Fields, H. L., and Basbaum, A. I., 1999, Central nervous system mechanisms of pain modulation, in *Textbook of Pain*, 4th ed. (P. D. Wall and R. Melzack, Eds.), London: Churchill Livingstone, pp. 309–329. ♦
- Gracely, R., Lynch, S., and Bennett, G., 1992, Painful neuropathy: Altered central processing, maintained dynamically by peripheral input, *Pain*, 51:175–194.
- Levine, J., and Reichling, D., 1999, Peripheral mechanisms of inflammatory pain, in *Textbook of Pain*, 4th ed. (P. D. Wall and R. Melzack, Eds.), London: Churchill Livingstone, pp. 59–84.
- Melzack, R., and Wall, P., 1965, Pain mechanisms: A new theory, *Science*, 150:971–979.
- Peyron, R., Laurent, B., and Garcia-Larrea, L., 2000, Functional imaging of brain responses to pain: A review and meta-analysis, *Neurophysiol. Clin.*, 30:263–288.
- Rainville, P., Duncan, G. H., Price, D. D., Carrier, B., and Bushnell, M. C., 1997, Pain affect encoded in human anterior cingulate but not somatosensory cortex, *Science*, 277:968–971.
- Raja, S. N., Meyer, R. A., Ringkamp, M., and Campbell, J. N., 1999, Peripheral neural mechanisms of nociception, in *Textbook of Pain*, 4th ed. (P. D. Wall and R. Melzack, Eds.), London: Churchill Livingstone, pp. 13–44.
- Vallbo, A. B., Hagbarth, K. E., Torebjork, H. E., and Wallin, B. G., 1979, Somatosensory, proprioceptive, and sympathetic activity in human peripheral nerves, *Physiol. Rev.*, 59:919–957.
- Wall, P. D., 1999, *Pain: The Science of Suffering*, London: Weidenfeld and Nicholson. ♦
- Waxman, S. G., Dib-Hajj, S., Cummins, T. R., and Black, J. A., 1999, Sodium channels and pain, *Proc. Natl. Acad. Sci. USA*, 96:7635–7639.
- Woolf, C. J., and Salter, M. W., 2000, Neuronal plasticity: Increasing the gain in pain, *Science*, 288:1765–1769.

Past Tense Learning

Amit Almor

Introduction

The English past tense, a seemingly simple linguistic phenomenon that has come to epitomize the latest round in the centuries-old debate between rationalists and empiricists, exemplifies the processes that more generally handle the formation of words and their structure. These processes have been traditionally studied by the linguistics field of “morphology,” which has assumed a level of representation that is based on the smallest meaning-bearing linguistic units, called *morphemes*. The main goal of linguistic morphology has been to systematically identify morphemes and describe the principles that govern the way they are used to form words (see Spencer and Zwicky, 1998, for current issues in linguistic morphology). Morphemic representation lies in between the phonological level, where the basic representations consist of individual sounds (phonemes), and whole words. Some morphemes are identical to whole words. For example, each of the words *base* and *ball* consists of one morpheme (/base/ and /ball/, respectively), and the word *baseball* consists of these two morphemes combined. Not all morphemes are words, however; the word *dislike* consists of the non-word morpheme /dis/ and the word morpheme /like/, and the word *liked* consists of the stem morpheme /like/ and the non-word past tense morpheme /ed/. The relation between meaning and word form is not always transparent. One reason is that the meaning of some morphemes can only be traced etymologically. For example, the meaning of *-mit* in *permit*, *submit*, and *remit* is not transparent to modern day English speakers but can be traced back to the Latin suffix *mittere* (roughly meaning, “to let go”). Moreover, even when the meanings of the component morphemes are known and can be ascertained (as in /break/fast/, i.e., the meal that breaks the fast) it is not clear that these meanings are transparent and accessible enough to affect actual word use. Finally, even when the meanings of the component morphemes are transparent and readily accessible, the meaning of the resulting compound is not consistently entailed. Although there is much regularity in how the meanings of morphemes are related to word meanings (e.g., *housedog* is a kind of dog and a *doghouse* is a kind of house) there are many exceptions. *Hotdog* is not a kind of dog, *sweetbread* is not a kind of bread, and *hammerhead* is not a kind of head. Thus, although the relation between word form, word meaning, and morphemes seems to encompass much regularity, it is not trivial and cannot be easily captured by simple combinatorial rules.

This may be differentially true for different kinds of morphological processes. Most morphological theories distinguish between inflectional and derivational processes. Inflectional processes generally involve elements of word structure that are related to grammar, such as markings for tense, number, gender, and case. Languages vary in the extent to which this grammatical information is morphologically encoded. In Hebrew, verbs are marked for tense, number, gender, and argument structure, resulting in a complex and highly regular verbal inflectional system. In English, verbs are only marked for tense and number, and in Mandarin Chinese, verbs are marked for neither tense nor number. Inflectional processes apply in accordance with sentence structure (in English, the subject noun of a sentence has to agree in number with the verb; a failure to do so results in ungrammaticality) and do not alter the core meaning or grammatical class of the inflected stem. Inflectional processes can be viewed as generating classes of systematically related word forms from a basic stem/root form via a small set of morphological operations. Inflectional processes are also highly productive in the sense that they apply to all the words in the language with new words “automatically” receiving a so-called regular “default” treatment. For example, all verbs in English are marked for tense and new verbs generally receive the regular *-ed* past tense suffix (e.g., *faxed*, *emailed*).

In contrast, derivational processes are more open ended and involve the creation of new words from other words and morphemes, often resulting in changes in grammatical category. For example, in English, the suffix *-ly* can be used to form adverbs from adjectives: *glad-gladly*, *poor-poorly*, etc. Derivational processes do not apply across the board (e.g., the suffix *-ly* cannot be added to adjectives such as *tall*, *old*, *young*).

Historically, the more regular nature of inflectional processes suggested that the relation between morphemes, meanings, and word forms could be more easily explained in the area of inflectional morphology than in the area of derivational morphology. It also suggested that the underlying computational machinery might involve rules. Although in recent years derivational processes have begun to attract as much attention as inflectional processes, it has been in the area of inflectional morphology that most current ideas and theories were formed.

Regulars and Rules

Indeed, for many years, regular inflection such as the *-ed* English past tense suffix was used as the showcase example for symbolic

rules in mental computation. Two main observations were most often cited. The first is that speakers readily apply the regular rule to new words they haven't heard before, regardless of the new words' similarity to other words that conform to the regular pattern. This observation has received much empirical support from numerous studies using variants of Berko's (1958) well-known "Wug test" in which a subject is asked to complete a fragment like: "Here is one *wug*. These are two____." Adults and children from about age three usually apply the regular English plural suffix and respond "wugs" despite never having heard the word *wug* before. Similar results have been obtained with the English past tense in that subjects aged three and older readily inflect a novel verb with the regular *-ed* suffix. The second observation thought to support symbolic rules was children's over-regularization errors. It has long been observed that children exhibit what may seem like a paradoxical decline in language performance starting at around three years of age. At this age, children who may have previously used irregular forms correctly suddenly start inappropriately regularizing many irregular forms. A child who may have already used the past tense form *went* in her speech might suddenly start using *goed* or *wented*. This over-regularization lasts well into the elementary school years. This learning pattern is usually referred to as a "U-shaped learning curve" because when plotted against age, children's performance worsens and then improves, thus resembling the shape of the letter U. U-shaped learning was believed to reflect children's switch from rote memorization of both regular and irregular past tense forms to the use of symbolic rules. By this account, at around age three, children's ability to use rules matures and they start applying rules across the board, occasionally producing over-regularization errors.

The Past Tense Debate

The view that regular inflection shows the working of an underlying rule was challenged by Rumelhart and McClelland's (1986) landmark connectionist model. This model was trained to map phonological representations of stem forms into output phonological representations of past tense forms. By using a general error-correcting learning algorithm, the statistical relations between the input stem phonemes and the output past form phonemes were encoded in the weights of the links. These learned associations enabled the model to generalize and produce the regular past tense form for novel verbs it was not trained on, thus exhibiting behavior that could be described as rule-governed even though no rules were involved in producing this behavior. Remarkably, without implementing any underlying rules, the model's course of learning mimicked the U-shaped curve characteristic of children. The model thus illustrated how the generalization ability previously thought to be the signature of an underlying rule mechanism can arise in a connectionist network without explicit rules. This model had a substantial effect on the fields of psycholinguistics and cognitive science. After two decades in which statistical learning was considered inadequate as an explanation of language (mainly due to an influential critique of statistical learning by Chomsky, 1959), Rumelhart and McClelland's work showed that statistical learning can provide a viable alternative to symbolic rules.

Rumelhart and McClelland's challenge to symbolic rules met with considerable criticism, starting with Pinker and Prince (1988). Critics argued that the behavior of the model diverges in important ways from human behavior. For example, unlike humans, the model did not generalize well to novel forms that have an unusual sound (e.g., the model mapped the stem *tour*, which was not in the training set, to the past tense *toureder*.) Critics also argued that the most impressive feat of the model, its U-shaped learning curve, is the result of an implausible and carefully engineered training regime that does not parallel the input to the child or the child's own

output. Indeed, many critics argued that in order for a connectionist model to properly handle regular inflections, it must implement rules albeit using connectionist machinery. While this view concedes that such "implementation models" can help explain how rules are represented and executed by the brain, it nevertheless maintains that such models do not provide a theoretical alternative to rule based theories (for a summary of the criticism of Rumelhart and McClelland's model, see Pinker, 1999).

The combination of successes and problems of the Rumelhart and McClelland model led many researchers to propose that while connectionist models may provide an adequate account for how irregularly inflected forms are processed, the processing of regular inflections must be driven by a rule-based mechanism. This "dual mechanism" approach maintains both a rule-based mechanism and a connectionist memory system living side by side. The latter contains memorized mappings from stem to inflected forms for all the irregular mappings but, being nonselective, may also include frequently encountered (and possibly irregular sounding) regular mappings. To explain how processing in the two components is coordinated, this view also stipulates a "blocking mechanism" that allows an activated memorized inflected form to block the usage of the default rule (for a detailed presentation of the dual route model, see Pinker, 1999.)

The critique of Rumelhart and McClelland's model and the subsequent development of the dual mechanism approach provided the road map for much of the research that has followed since. A new generation of connectionist models attempted to address the design flaws and empirical shortcomings of the original Rumelhart and McClelland model as well as to broaden the scope of the empirical data covered. Many models added hidden layers and clean-up units to allow for a wider range of learnable mappings. Other models included semantic representations, which enabled them to address more than one type of inflection and to assign different inflections to homophonic stems. For example, MacWhinney and Leinbach (1991) developed a model that handled both noun plurals and verb past tenses and that could further distinguish past tense forms of homophones (*break-broke* vs. *brake-braked*). Plunkett and Marchman's (1993) model addressed many of the developmental issues related to the U-shaped learning curve.

Although these and many other models have successfully addressed many of the problems in the original Rumelhart and McClelland model, they nevertheless met with new criticism from dual mechanism theorists. One line of criticism continued to be directed at specific design aspects of various models, with some models accused of stealthily implementing the past tense rule. For example, the MacWhinney and Leinbach's (1991) model was accused of implementing a rule because, in addition to the regular input to output connections, it had special connections between the corresponding phonological units in the input and output (Pinker, 1999). The value of this line of criticism is questionable because, as connectionist theorists are quick to admit (e.g., Seidenberg, 1997), implemented models require many simplifying assumptions. Broadly criticizing models for making these assumptions without considering their specific implications risks missing important insights about the connection between behavior and underlying computation.

Moreover, this kind of criticism can be equally directed at the dual mechanism account, which makes its own set of problematic assumptions. In particular, it does not make clear how the blocking mechanism works, how and why it develops, and why is it not triggered by the many regulars that were already learned before the rule mechanism matured. As is often the case with theoretical constructs that are hard to explain or motivate, blocking is assigned the status of an innate mechanism. This not only relieves the theory from having to explain how this mechanism works and develops, but also adds one more marvel to the bag of wonders that cannot

be explained without an arsenal of highly specialized language-specific mental machinery.

Besides debating details of implementation, much research has focused on behavioral and neuropsychological differences between the processing of regulars and irregulars and on the implications of such differences for the single and dual mechanism theories.

Frequency of Inflected Forms

One way in which regulars and irregulars differ has to do with how the frequency of inflected forms affects their processing. The use of irregularly inflected forms is strongly affected by their frequency, and to the extent that regularly inflected forms show frequency effects, these effects are quite small (see Pinker, 1999, for details). According to the dual mechanism view, these differences indicate that regulars and irregulars are processed by two separate mechanisms. This is because the memorized irregular mappings are strengthened each time an irregularly inflected form is encountered, but the application of the regular symbolic rule is not sensitive to the properties of the individual stems or inflected forms. The small frequency effects with regularly inflected forms found in some experiments is explained as a result of special circumstances that encourage the use of the memory system, for example when subjects are presented with a list containing an unnaturally high number of irregular forms.

Single mechanism accounts can also explain differences in the frequency dependence of regulars and irregulars. In these systems, frequency by regularity interactions can arise because “regular” input to output mappings, which are shared by a large number of different inputs, are less dependent on the frequency of individual input-output mappings, whereas “irregular” mappings that are shared by only a small number of inputs are more sensitive to the frequency of individual mappings (Seidenberg and McClelland, 1989). By this account, regularity itself, and by extension frequency effects, fall along a continuum defined by the number and similarity of input-output mappings.

A frequency effect that can better distinguish between the two theories is related to whether the frequency of regular forms makes their production less or more prone to interference by similarly sounding irregulars. Empirical evidence shows that such irregularization errors are more likely for low frequency regular verbs than for high frequency regular verbs and that the latency of correct regular production for high frequency regulars is less affected by interference from phonologically similar irregulars than the latency of correct responses for low frequency regulars (Long and Almor, 2000). These findings are compatible with a single mechanism view in which regulars and irregulars are processed similarly such that more frequent items are processed more quickly and with fewer errors than less frequent items. These results are not compatible with the dual mechanism view. Because, by this account, the only kind of *regularly* inflected forms that are stored in the memory system and are therefore prone to interference are high frequency regulars, this account falsely predicts that competition between similarly sounding irregular and regular stems would only occur for high frequency regulars but not for low frequency regulars (Pinker, 1999, page 303, fn 22). Obviously, the dual mechanism account can be easily modified to say that regular past tense production can benefit from (rather than be hindered by) the existence of a stored mapping in the associative network. Alternatively, the dual mechanism account could be modified to say that regular mappings of stems that rhyme with irregularly inflected stems are stored in the associative network regardless of frequency. The apparent ease with which such post hoc modifications can be applied to the theory highlights its underspecification and in particular the vagueness of the blocking mechanism.

Further frequency-related arguments have been made on the basis of inflectional systems in other languages such as German and Hebrew in which, unlike English, the default inflection was argued not to apply to the majority of stems (see Pinker, 1999, chapter 8, for details). According to dual mechanism theorists, an inflection that applies to a minority of the stems in the language but that is productively applied to new forms is not compatible with the statistical learning of connectionist models. Some of the relevant data, however, has been called into question (Bybee, 1995). In particular, even if the stems that undergo the so-called regular inflection are not the majority of stems, the combined instances of these stems can still be more frequent than the instances of other inflections. Although to date the implications of inflections in other languages have not been fully explored, they may eventually help resolve some of these issues in ways that are not possible using the morphologically impoverished English.

Neurological Impairments and Imaging

Findings from different neurological impairments that selectively affect the processing of either regulars or irregulars, as well as imaging findings of different brain activation patterns associated with processing regulars vs. irregulars, have also been cited as evidence for separate mechanisms. Marslen-Wilson and Tyler (1997) found that although some aphasic patients seem to lose the knowledge that regular stems and inflected forms are related (exposure to *walked* did not speed their subsequent response to *walk*) but not the knowledge that uninflected regular forms are related to their stems (exposure to *found* speeded their subsequent response to *find*), other aphasic patients show exactly the opposite loss pattern. Other researchers found similar double dissociations in other neurological disorders. Aphasic patients with agrammatism, patients with Parkinson’s disease, and children with Williams syndrome have been reported to have more trouble with regulars than irregulars, while anomie aphasics, patients with Alzheimer’s disease, and some children with SLI have been reported to have more trouble with irregulars than regulars (see Pinker, 1999, chapter 9; and Ullman, 2001, for details). Functional imaging studies have also revealed temporal as well as location differences in the brain activation accompanying the processing of regulars vs. irregulars. Although the exact temporal patterns and brain areas vary from study to study (and sometimes from one experiment to the next within the same study), it seems that the processing of regulars more strongly involves left frontoparietal cortex, whereas the processing of irregulars involves tempoparietal areas in both hemispheres. Dual mechanism supporters have interpreted these dissociations as indicating that the grammar areas in the brain handle regular processing, and lexical semantics areas handle irregular processing. An interesting variation of this account has been proposed by Ullman (2001), who views regular processing as relying on procedural memory, the mental storage for skills and other routine mental operations, and irregular processing as relying on declarative memory, the storage for events and idiosyncratic facts. Ullman’s view differs from that of many other dual mechanism theorists because inflectional rules are not considered a specialized language mechanism any more than bicycle riding skills are considered a specialized bicycling mechanism. Arguably, the only language-specific machinery in Ullman’s account is the blocking mechanism.

Although these neurological dissociations seem to favor the dual mechanism view, connectionist models that include both semantic and phonological representations can also show selective impairments to regulars or irregulars as a result of certain artificial lesions. These dissociations occur because irregulars depend more on semantics than on phonology, whereas regulars depend more on phonology than on semantics. By this argument, brain impairments

that affect regulars more than irregulars simply involve more damage to phonological than to semantic representations while brain impairments that affect regulars more than irregulars involve more damage to semantic than to phonological representations (Joanisse and Seidenberg, 1999).

Discussion

Despite the active controversy surrounding the mental processing of regular inflections, most researchers agree that the mental processing of irregular inflections is not rule governed but rather works much like a connectionist network. It is therefore important to remember that Rumelhart and McClelland's model and its successors are important even if dual mechanism theories are correct and regular inflection is rule based.

Despite failing to flesh out important parts of their models, dual mechanism proponents enjoy the advantage that rules provide an intuitively appealing explanation to regular behavior. Rules, however, need not be part of the language processing mechanism, as is assumed by the dual mechanism account. Instead, rules could be meta-linguistic. People are clearly able to consciously identify regularities and describe them with explicit rules that can then be deliberately followed. This is often used in second-language learning when students are explicitly instructed about certain "rules." Meta-linguistic rules may also be used in tasks that require overt responses such as the "Wug" test, which requires people to inflect a novel form. The underlying language processing system may be only involved insofar as it supplies probabilistic input to the meta-linguistic rule (as in the "Wug" test), or as it is modified as a result of being exposed to the output of the explicit rule (as in second-language learning). This means that the linguistic processes underlying inflection should be studied by tasks that do not require making a choice of inflection but that instead rely on activation-based measures such as priming between inflected and stem forms. Comparing the results of such studies to the results of explicit response tasks would provide a better assessment of whether rules are meta-linguistic.

Road Map: Linguistics and Speech Processing

Related Reading: Connectionist and Symbolic Representations; Imaging the Grammatical Brain; Language Processing; Speech Processing; Psycholinguistics

References

- Berko, J., 1958, The child's learning of English morphology, *Word*, 14:150–177.
- Bybee, J., 1995, Regular morphology and the lexicon, *Lang. Cognit. Proc.*, 10(5):425–455.
- Chomsky, N., 1959, Review of B. F. Skinner's *Verbal Behavior*, *Language*, 35:26–58.
- Joanisse, M. F., and Seidenberg, M. S., 1999, Impairments in verb morphology after brain injury: a connectionist model, *Proc. Natl. Acad. Sci. USA*, 96(13):7592–7597.
- Long, C., and Almor, A., 2000, Irregularization: The interaction of item frequency and phonological interference in regular past tense production, in *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 310–315.
- MacWhinney, B., and Leinbach, J., 1991, Implementations are not conceptualizations: Revising the verb learning model, *Cognition*, 40(1–2):121–157.
- Marslen-Wilson, W. D., and Tyler, L. K., 1997, Dissociating types of mental computation, *Nature*, 387(6633):592–594.
- Pinker, S., 1999, *Words and Rules: The Ingredients of Language* (1st Ed.), New York: Basic Books. ♦
- Pinker, S., and Prince, A., 1988, On language and connectionism: Analysis of a parallel distributed processing model of language acquisition, *Cognition*, 28(1–2):73–193.
- Plunkett, K., and Marchman, V., 1993, From rote learning to system building: Acquiring verb morphology in children and connectionist nets, *Cognition*, 48(1):21–69.
- Rumelhart, D. E., and McClelland, J. L., 1986, On learning the past tenses of English verbs, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 2, *Psychological and Biological Models*, (J. L. McClelland, D. E. Rumelhart, and P. R. Group, Eds.), Cambridge, MA: MIT Press. ♦
- Seidenberg, M. S., 1997, Language acquisition and use: Learning and applying probabilistic constraints, *Science*, 275(5306):1599–1603. ♦
- Seidenberg, M. S., and McClelland, J. L., 1989, A distributed, developmental model of word recognition and naming, *Psychol Rev*, 96(4):523–568.
- Spencer, A., and Zwicky, A. M., 1998, *The Handbook of Morphology*, Malden, MA: Blackwell. ♦
- Ullman, M. T., 2001, A neurocognitive perspective on language: The declarative/procedural model, *Nature Rev. Neurosci.*, 2(10):717–726.

Pattern Formation, Biological

James D. Murray

Introduction

The generation of biological spatial patterns is fundamental to many disciplines, among them bacteriology, developmental biology, physiology, neurobiology, epidemiology, ecology, and tumor growth. In population biology, patchiness in population densities is the norm rather than the exception. In developmental biology, groups of previously identical cells follow different developmental pathways, depending on their position. The rich spectrum of mammalian coat patterns and the patterns found on fishes, reptiles, molluscs, and butterflies reflect developmental processes that are still not fully understood; Figure 1 shows some examples. Stationary patterns as well as a wide variety of waves have been observed in chemical reactions. Ocular dominance stripes reflect patterns in the connectivity of the visual cortex, while hallucination patterns can be partially explained as activity patterns in the visual cortex. These patterns have been used by shamans for millennia.

The discovery in 1998 of anti-angiogenic drugs, such as Angiostatin and Endostatin, opened up exciting new possibilities for anticancer therapy. The new therapy is based on a revolutionary idea put forward by Dr. Judah Folkman in the 1970s, namely, that if tumors are starved of nutrients, they will die, and that starvation could be achieved if angiogenesis (genesis of blood vessels) in the tumor could be prevented. Only recently (1998) has the success of this concept been reported: anti-angiogenic drugs stopped tumor growth in mice. This result refocused attention on the patterning process of angiogenesis and network formation of endothelial cells in extracellular matrix; it is one of the exciting research areas in pattern formation. In this article we briefly discuss a mechanical model for generating such networks. The model's mechanism is firmly based on known biology. A comparison of the model's predictions with experimental results shows remarkable correlations. The book on vascular morphogenesis edited by Little, Mironov,

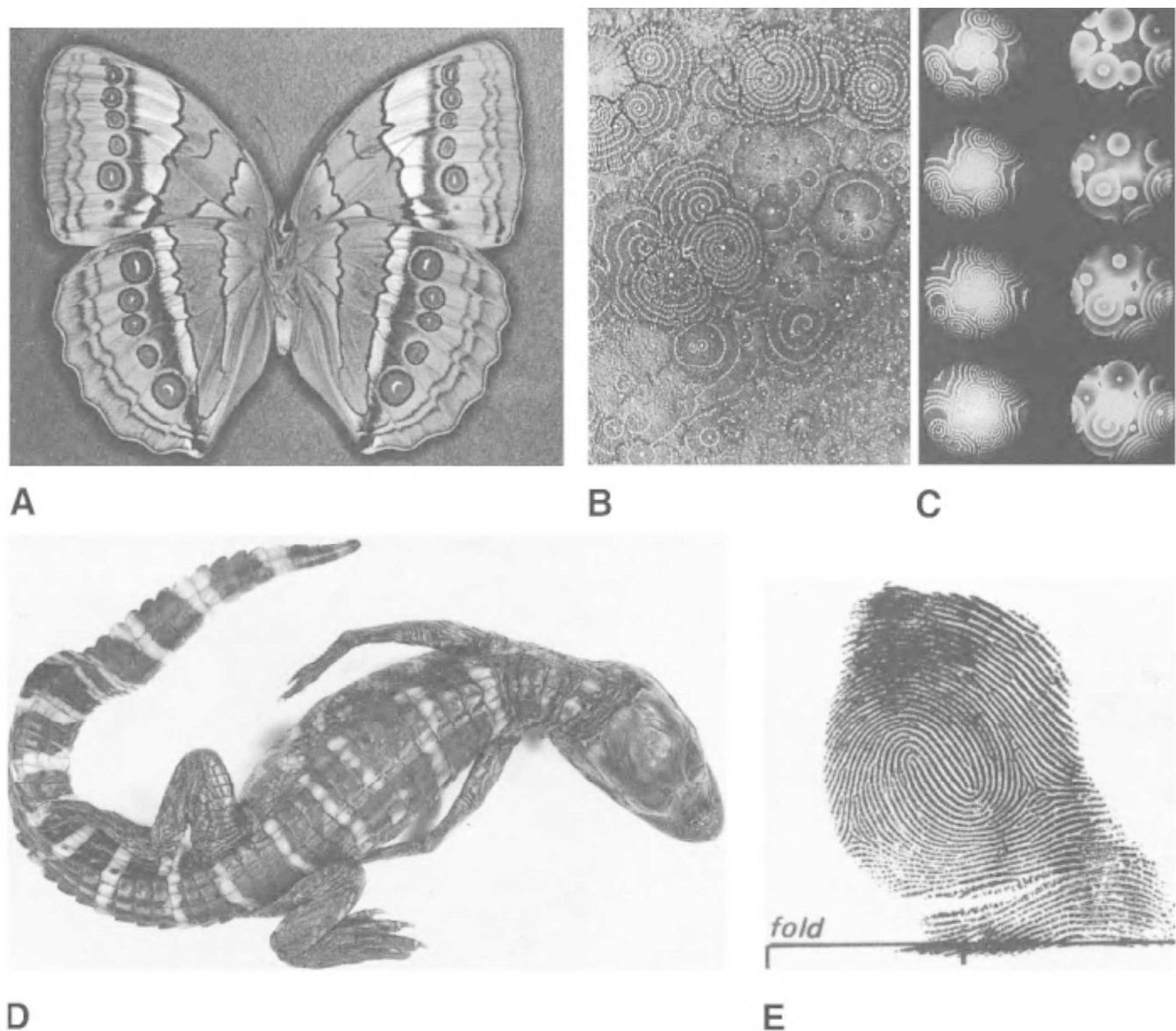


Figure 1. A small sampling of the diverse spatial patterns for which model mechanisms have been proposed. *A*, The butterfly (*Stichophthalama camadeva*) shown exhibits most of the basic pattern elements observed in butterfly wings. (Photograph courtesy of H. F. Nijhout.) *B*, Example of moving and stationary bands of amoebae of the slime mold *Dictyostelium*

discoideum. (Photograph courtesy of P. C. Newell.) *C*, Circular and spiral waves in the Belousov-Zhabotinskii reaction (Photograph courtesy of A. T. Winfree.) *D*, Stripes on an alligator (*Alligator mississippiensis*). (Photograph courtesy of M. W. J. Ferguson.) *E*, Typical human fingerprint.

and Sage (1998), with a foreword and brief description by Folkman (1998) of anti-angiogenic therapy as opposed to conventional therapy, provides an extensive review of current knowledge in the field. In this article we also touch on a new approach for predicting brain tumor growth in which angiogenesis does not play a role.

Although biological patterns occur on a wide range of spatial scales, spanning the molecular, cellular, individual, and population levels, a common feature is that macroscopic patterns result from microscopic interactions. Although genes play a crucial role in developmental outcomes, they do not actually produce the pattern. Mathematical models have been proposed for the mechanisms that generate these biological patterns, based on the principle of interactions between the relevant components. We describe two of the main classes, reaction-diffusion models and mechanical models,

and briefly mention some others. Each model can exhibit spontaneous pattern formation; that is, patterns develop in homogeneous environments without particular initial conditions, boundary conditions, or other external forces to drive them. The patterns are therefore self-organizing and symmetry breaking. The cancer cell patterns discussed in the brain tumor diffusion model are closely related, but there the patterns are not self-organizing.

Our goal is to develop a mechanism from knowledge of biology and to determine (1) the range of parameters attending this mechanism in which pattern formation is expected, (2) the nature of the pattern (steady, oscillating, or moving through space), (3) the scale of the pattern, and (4) perhaps the most important, the relation of the theoretical results, conclusions, and predictions to the actual biomedical problem. Relating model predictions to actual problems

frequently suggests new experimental avenues and new biological insights. One of the major strategies to investigate such a relationship is as follows. First, a suggested biological mechanism is translated into a set of mathematical equations (the model). An appreciation of the pattern formation potential of existing models is invaluable here, and we hope that this brief survey will be useful in that regard. Once the model has been specified, we determine a homogeneous steady state and use linear stability analysis to determine whether perturbations of such an unpatterned state will grow or decay. For parameters supporting pattern formation, we isolate unstable modes and use the dominant modes to predict the scale and shape of the pattern. Although we generally cannot estimate all the parameters from the experiment, nevertheless we can use our knowledge of the biology to suggest broad ranges for some of the key parameters. The development of sophisticated mathematical models bearing little or no relation to the underlying biology, even if they are interesting mathematically, is of scant interdisciplinary value.

Reaction-Diffusion Models

To date, reaction-diffusion models have been the most widely studied of the models we discuss. They have been applied with effect, for example, in developmental biology, tumor growth, wound healing, population biology, epidemiology, neurophysiology, chemistry, and physics (see Murray, 2002). The variables, which depend on time and space, may be a type of molecule or cell; we refer to them generally as species. Species disperse and react, and these two processes are independent. Developing expressions for local interactions between species and their flux, and invoking conservation laws, we obtain the general form

$$\frac{\partial n}{\partial t} = f(n) + D \nabla^2 n \quad (1)$$

where $n(x, t)$ is the vector of species densities, f is the vector of reaction terms, D is the diffusion coefficient matrix, and ∇^2 is the diffusion operator in one, two, or three space dimensions; t denotes time. Initial conditions and boundary conditions must also be specified.

The patterns we are most interested in are stable, stationary ($\partial n / \partial t = 0$), inhomogeneous solutions to Equation 1. For a single species in a single spatial dimension, it can be shown that a homogeneous steady state cannot be destabilized by diffusion: in two dimensions this is not always the case. Two-species models are very much more interesting, while three-species (and higher) models have not been studied in any depth, even though in most pattern formation situations several species are involved. The aim in practical modeling is to isolate the key dependent variables.

Two-Species Models

In 1952, Alan Turing (of Turing machine fame) suggested that the differential diffusion of two interacting species could act to destabilize a homogeneous steady state. Since diffusion is usually thought of as a stabilizing (or smoothing) force, this was a startlingly original idea. It has since been supported both mathematically and experimentally (see Murray, 2002, who describes many applications and gives numerous references).

Consider the two-species system in one dimension, x , given by

$$\frac{\partial A}{\partial t} = f(A, B) + d_A \left(\frac{\partial^2 A}{\partial x^2} \right), \quad \frac{\partial B}{\partial t} = g(A, B) + d_B \left(\frac{\partial^2 B}{\partial x^2} \right) \quad (2)$$

in which a steady state exists at (A_0, B_0) , where $f(A_0, B_0) = g(A_0, B_0) = 0$. Linearizing around (A_0, B_0) , we get

$$\begin{aligned} \frac{\partial A}{\partial t} &= f_A A + f_B B + d_A \left(\frac{\partial^2 A}{\partial x^2} \right), \\ \frac{\partial B}{\partial t} &= g_A A + g_B B + d_B \left(\frac{\partial^2 B}{\partial x^2} \right) \end{aligned} \quad (3)$$

where $f_A = \partial f / \partial A$ is evaluated at (A_0, B_0) , and so on. We assume that the steady state is stable in the absence of spatial interaction (diffusion) and therefore that $g_B < 0$ without loss of generality.

We look for solutions of the form $e^{\lambda t + i k x}$ and generate a *dispersion relation* relating eigenvalues, λ , to modes, k (the growth rate of mode k is $\text{Re}(\lambda(k))$). The dispersion relation provides conditions for unstable modes to exist (see Murray, 2002, vol. II, for a full exposition and many other practical uses of dispersion relations):

$$f_A + g_B < 0 \quad \Delta = f_A g_A - f_B g_A > 0$$

($g_B < 0$ without loss of generality)

$$d_A > 0, \quad \delta f_A + g_B > 0, \quad (\delta f_A + g_B)^2 > 4 \delta \Delta \quad (4)$$

where $\delta = d_B / d_A$

Some necessary conditions for Turing instability are (1) the self-inhibiting species, B ($g_B < 0$), must diffuse at the higher rate, and (2) A must be self-activating ($f_A > 0$). Also, f_B and g_A must have opposite signs. The species that promotes growth of the other is the *activator*, and the other species is the *inhibitor*. The two possible cases are illustrated schematically in Figure 2. In case 1, A is the activator (which is also self-activating), while the inhibitor, B , diffuses at a higher rate and inhibits not only A but also itself. In case 2, B is the activator, again self-inhibiting, and again diffuses at a higher rate. It can be shown that in case 1 the two species occur at high or low density together (Figure 2C), whereas in case 2, A is at high density where B is low, and vice versa (Figure 2D).

We now give two analogies, one ecological for conceptual simplicity, for the mechanisms underlying Turing instabilities and the evolution of spatial patterns. Consider case 1, and refer to Figure 3A, which are sketches of the local phase plane. Let A be prey to a predator, B . How can patterns arise as in Figure 2C when predators disperse more rapidly than their prey? Suppose there were an area of increased prey density. In the absence of diffusion, this would be damped out after a temporary increase in both populations. However, with high predator dispersal, it is possible that the local increase in predators partially disperses and hence is not strong enough to push the prey population back toward equilibrium. Furthermore, when predators disperse, they lower the prey density in neighboring regions and cause the opposite effect. It is thus possible to have alternating clumps of high and low population density of both species.

Consider the second type of dynamics (Figures 2B, D, and 3B). Suppose now that A is a slowly dispersing, *autocatalytic* ($f_A > 0$) predator and B is its prey. In an area of high prey density, without diffusion, predator numbers would increase at the expense of prey, and eventually both populations would return to the steady state. However, there is a transient increase in the predator population and a reduction in the prey population to below its steady-state value. The resulting net influx of rapidly dispersing prey from neighboring regions would cause the predator population to drop in those regions while prey flourished. A pattern can become established in which areas of few predators and many prey supply prey to areas that contain few prey and large numbers of predators.

The dispersion relation also indicates the scale on which a pattern occurs, through the wavelength, l_c , of the fastest growing mode:

$$l_c = \frac{2\pi(d_B - d_A)^{1/2}}{[(\delta + 1)((-f_B g_A)\delta)^{1/2} - f_A + g_B]^{1/2}}, \quad \text{where } \delta = \frac{d_B}{d_A} \quad (5)$$

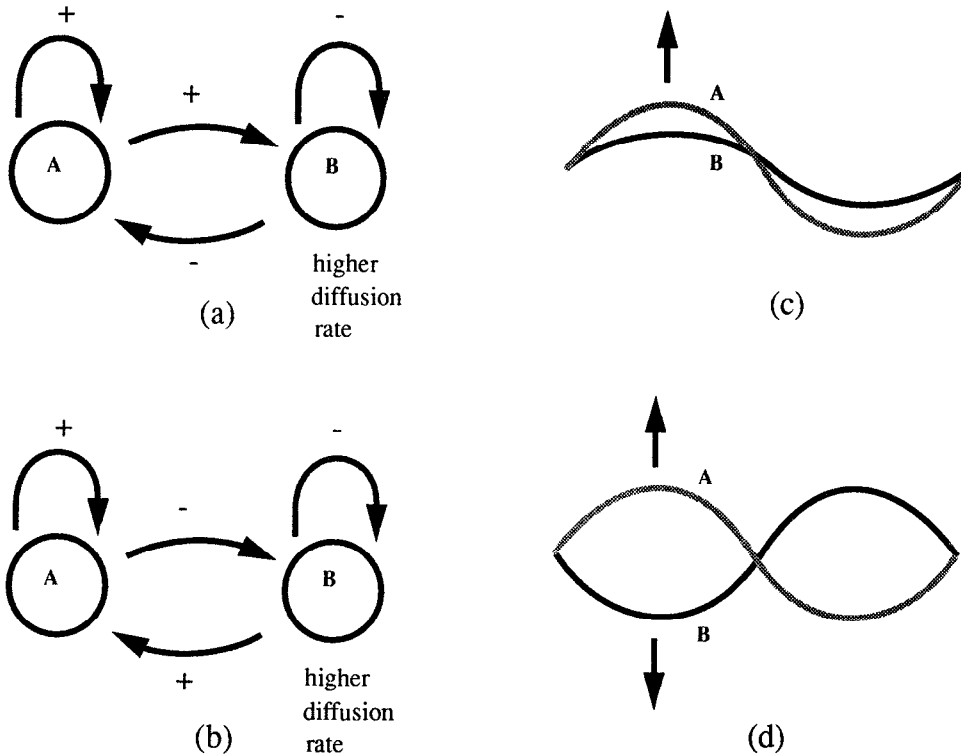


Figure 2. Some interactions support diffusion-driven instabilities. In part A, self-activating A also activates B, which inhibits both species. The resulting spatial pattern is shown schematically in part C. In part B, self-activating A now inhibits B but is itself activated by B. The resulting pattern is shown in part D. Corresponding reaction phase planes are shown in Figure 3.

(see Murray, 2002, vol. II). Equation 5 indicates that Turing instabilities can occur on a broad range of spatial scales. For large δ , $l_e \approx 2\pi[(d_A d_B)/(-f_B g_A)]^{1/4}$.

Mathematical analysis can also provide insight into the effect of boundary size and shape on pattern formation. In models for animal coat patterns (Murray, 2002, vol. II, and references there), one finds that only crosswise stripes can occur in long narrow domains, whereas spots can occur on wider domains. This is a possible explanation for why animals that have spots over most of their bodies (e.g., leopards) tend to have hooped patterns, or no pattern at all, on their tails; it also explains why a striped animal cannot have a spotted tail. The qualitative form of the pattern is governed by the size and shape of the animal at the time the pattern is determined.

In Turing's theory of morphogenesis, the developmental concept is that a chemical two-species system generates a landscape of chemical concentration to which cells react differentially and hence form spatial patterns. These chemicals are referred to as morphogens. Although these types of models have prompted enlightening experiments, their use has been limited because of the illusive nature of the morphogens in real biological situations. This view is changing; see the important paper by Lander, Nie, and Wan (2002).

Virtual Brain Tumors: Predicting Growth and Enhancing Medical Imaging

A medical use of classical diffusion models is that of predicting the growth (and control) of brain tumors (gliomas). This involves the spatial spread from an initial source of cancerous cells. Gliomas differ from most other cancers by their diffuse invasion of the surrounding normal tissue. Although medical imaging has increased the detection of gliomas, it has still a long way from defining accurately enough the degree of tumor cell invasion peripheral to the bulk of the tumor mass. This inadequacy of current medical imaging is substantiated by the fact that even extensive surgical re-

moval (resection) or local irradiation of gliomas is followed by tumor recurrence at or near the edge of the resection bed.

Since the mid-1990s, E. C. Alvord, M.D. (neuropathology), J. D. Murray (applied mathematics), and their research group at the University of Washington have developed mathematical models to quantify the spatiotemporal growth and invasion of gliomas in three dimensions throughout a virtual human brain with a resolution of 1 mm³ in which, latterly, the anatomically correct distributions of gray and white matter have been defined (Swanson, Alvord, and Murray, 2000, 2002; see also Murray, 2002, vol. II, for a full review). These mathematical models quantify the extent of tumorous invasion of individual gliomas to a degree beyond the capability of present medical imaging, including even microscopy. They also quantify the consequences of certain control therapies such as chemotherapy and resection.

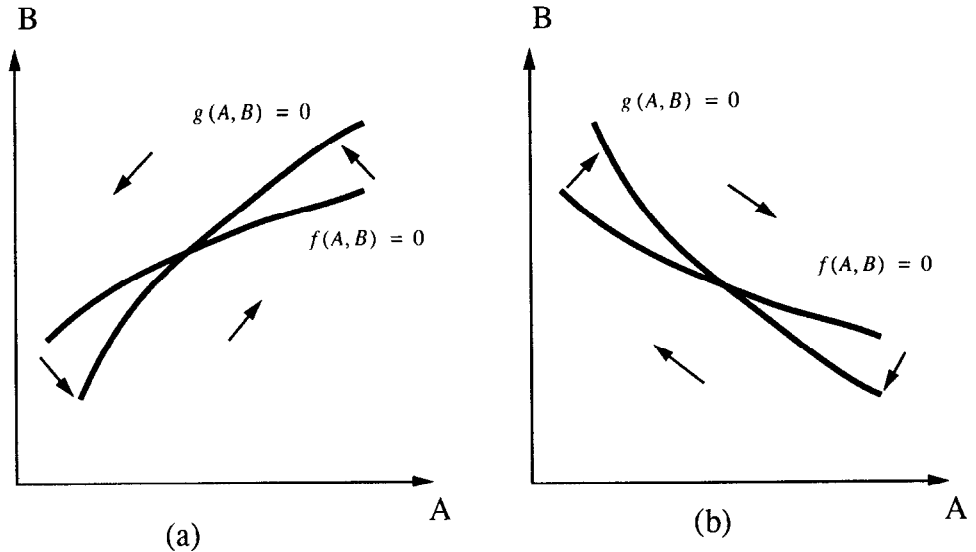
We briefly describe their approach here. Our basic model can vary the two key characteristics, proliferation and cell motility (diffusion), in heterogeneous, anatomically accurate brain tissue. We allow the motility coefficient to differ depending on the local tissue composition.

Our basic mathematical model quantifying the differential motility of gliomas in gray and white matter is quantified mathematically by

$$\frac{\partial c}{\partial t} = \nabla \cdot (D(\mathbf{x})\nabla c) + \rho c, \quad (6)$$

where $c(\mathbf{x}, t)$ is the concentration of tumor cells at location \mathbf{x} and time t . $D(\mathbf{x})$, a function of position \mathbf{x} in the brain, is the diffusion coefficient defining the random motility of the glioma cells with $D(\mathbf{x}) = D_g, D_w$, constants for \mathbf{x} in gray and white matter, respectively. ρ represents the net proliferation rate of the glioma cells. This term predicts exponential growth of the cancer cells, but within the survival period of the patient it has been shown to be a very good approximation. The diffusion coefficient in white matter

Figure 3. Reaction phase planes for diffusion-driven instabilities. Part A shows the phase plane corresponding to parts A and C of Figure 2. Part B shows the phase plane corresponding to parts B and D of Figure 2. The steady state is at the intersection of the two null clines. Arrows represent the direction of change due to local species interaction.



is larger than that in gray matter; that is, $D_w > D_g$. The model assumes that the tumor has grown to about 4,000 cells as a local mass before it begins to diffuse and the model Equation 6 applies.

For every medical imaging technique there is a threshold of detection below which glioma cells are not detectable. A tumor boundary detected by enhanced computed tomography (CT) corresponds to a tumor cell concentration of only about 8,000 cells/mm³. Mathematically, of course, in our virtual tumor, we can set the detection at any level we choose.

Based on the medical literature, the models assumed that a medical diagnosis is made when the volume of an enhanced CT-detectable tumor has reached a size equivalent to an average 3 cm in diameter and that death occurs when the volume reaches an average 6 cm in diameter. The difference between these two times can be defined as the survival time of the hypothetical or virtual patient. We can estimate these times from the model results, and they agree well with the average survival times for each grade of gliomas we have studied.

Crucial to the models' use is the ability to determine reasonable estimates of the critical parameters, the growth rate ρ and the diffusion coefficient D . We have estimates, but they vary quite widely. Some of the current research is focused on determining the values for individual patients, and these values can be then used in prognosis and treatment protocols.

Figure 4 shows three perpendicular cross-sections (coronal, sagittal, and horizontal or axial) of the virtual human brain, intersecting in a point marked by an asterisk in the superior frontal region where the virtual tumor originates. The gray and white matters of the brain domain appear gray and white, respectively. In each image, a single thick black curve defines the edge of the tumor that the model suggests would be detectable on enhanced CT, associated with a threshold of detection of 8,000 cells/mm³. The outermost profile corresponds to an arbitrary threshold of detection 80 times more sensitive than enhanced CT (i.e., 100 cells/mm³). The left column of images in Figure 4 represents the tumor at the time of detection, defined as an enhanced CT-detectable tumor with average diameter of 3 cm, while the right column represents the tumor at the time of death, defined by an enhanced CT-detectable tumor with average diameter of 6 cm. The simulations clearly reveal the subthreshold invasion of the tumor well beyond the detectable portion of the tumor. No matter the extent of resection, the mathematical model indicates that the gross tumor will ultimately recur and kill.

Unlike real patients with real gliomas, virtual patients with virtual gliomas can be analyzed by letting any particular factor vary while keeping all the other determining factors constant. Such isolation techniques, of course, require a mathematical model that has sufficient complexity to contain a realistic number of variables. The recent availability of simulated MRI, with proportions of gray and white matter accurately indicated, permitted the development of this model, which is sufficiently complex to allow different diffusion rates in gray and white matter (e.g., a 5-fold increase in diffusion or migration in white matter).

Murray-Oster Mechanical Models

The Murray-Oster mechanical theory is based on the fact that mesenchymal cells exert significant traction forces on the surrounding extracellular matrix, thereby influencing their movement and the density of the surrounding matrix. The model simply consists of conservation equations for the cells and matrix and a force balance equation for the interaction between the cell-generated forces and the viscoelastic resistive properties of the matrix. An extensive pedagogical review is given in Murray (2002, vol. II). Harris, Stopak, and Warner (1984) first presented graphic experimental confirmation of some of the predictions made with the original model system.

The overriding feature of the Murray-Oster mechanical theory of pattern formation is its simplicity (in spite of its relative mathematical complexity): complicated patterns arise solely as a consequence of the interaction of the cell traction forces and the viscoelastic properties of a deformable matrix in which they are embedded. We give an intuitive description of the process below. The theory has been widely used in a variety of problems, especially in wound healing studies (for example, Tranquillo and Murray, 1993; Olsen, Maini, and Sherratt, 1998; and the review by Sherratt and Dallon, 2002).

The cell-matrix network patterns found experimentally (Vernon et al., 1995) when vascular endothelial cells are placed on a basement extracellular membrane provide a unique theoretical and experimental opportunity to determine the important components in the morphogenetic process of the complex network formation. Figure 5 shows an example of typical patterns obtained from the model mechanism and experiment. In these experiments, endothelial cells were placed on a gelled basement membrane matrix (BMM, Ma-

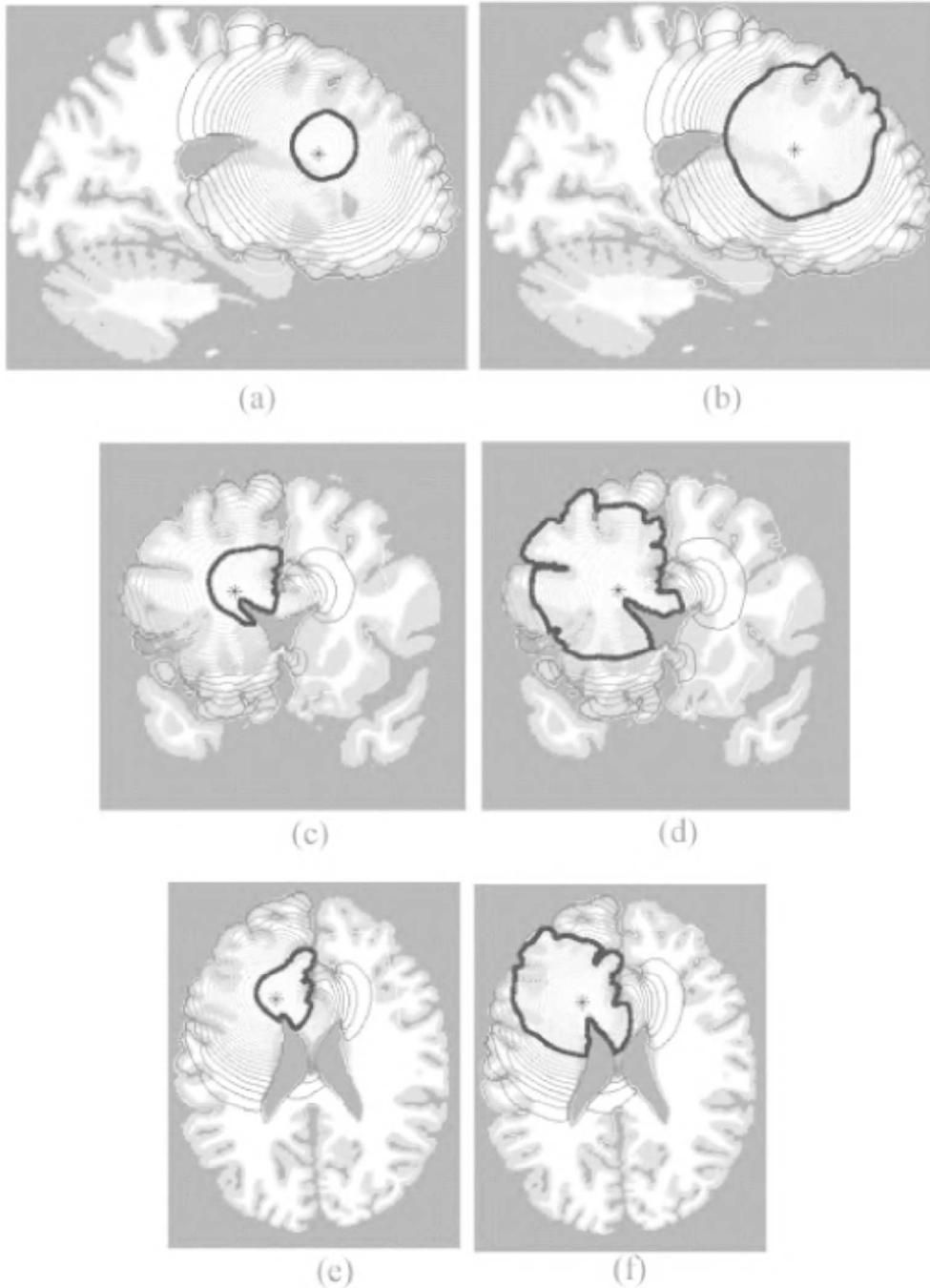


Figure 4. Sections of the virtual brain in sagittal, coronal, and horizontal planes that intersect at the site of the brain tumor (glioma) originating in the thalamus denoted by an asterisk. A thick black contour defines the edge of the tumor detectable by enhanced computed tomography. Cell migration was allowed to occur in an anatomically accurate three-dimensional representation of the human brain.

trigel). In the course of a few hours, aggregates started to form, and after 24 hours all of the BMM had been pulled into an aggregate network.

In view of the importance of angiogenesis in cancer therapy, we describe a version of the theory that generates networks of cells and matrix and that has been suggested (Manoussaki et al., 1996; Murray et al., 1998) as the possible mechanism of network formation (Vernon et al., 1995). The model shows unequivocally that cell-matrix contact guidance plays a crucial role. There is no cell proliferation in the case of the cell-Matrigel patterns mimicking early network patterns of angiogenesis, in which the complex strand-like structures of matrix are formed. The pattern forms

purely from cellular traction forces and their interaction with the fibrous matrix: there are no external forces, cell proliferation, or other complicating factors. We can thus isolate, in our models, the central mechanical interaction underlying a range of biological phenomena in development and other contexts. There are a number of well-determined parameters, such as cell density, matrix density, matrix thickness, pattern wave number, time scale, and some parameters that we can confine to a range of possible values, as well as the Young modulus and the magnitude of cellular traction forces.

The conservation equations in this (experimental) situation are

$$\frac{\partial n}{\partial t} + \nabla \cdot [nv - D(\epsilon)\nabla n] = 0, \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0 \quad (7)$$

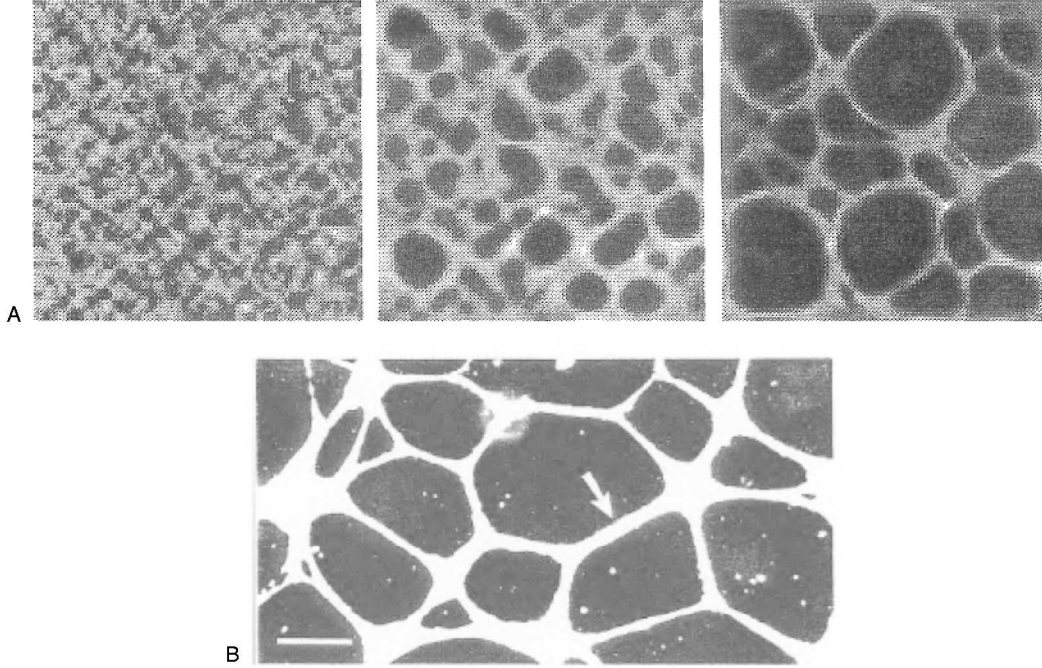


Figure 5. Cell density network formation when cells are embedded in a matrix gel: white areas denote high cell densities. *A*, Initially the cells were uniformly distributed and the model equations simulated numerically with the parameters obtained from the literature and related experiments: the time scale for the pattern formation is 24 hours; the complete square is $800 \mu\text{m}$.

B, Experimental patterns obtained with bovine aortic endothelial cells cultured on Matrigel. Each cord (example arrow) consists of many cells; the image is viewed by darkfield illumination (bar = $200 \mu\text{m}$). (Photograph courtesy of R. B. Vernon.)

where cells (n) and matrix (ρ) convect as a single phase with material velocity $v = Du/Dt$. $D(\varepsilon)$ is the cell motility tensor, dependent on the matrix strain, ε , defined by

$$\varepsilon = \frac{1}{2} (\nabla \cdot u + \nabla \cdot u^T)$$

where u is the vector of matrix displacement and u^T its transpose.

The force balance equation simply says that the traction forces generated by the cells, $T(n, \rho)$; the induced response stress of the matrix, σ ; and the substrate anchoring, $(s/\rho)(\partial u/\partial t)$, where s is the strength of the fluid-like drag force of the matrix attachment, are in equilibrium, namely

$$\nabla \cdot [\sigma + T(n, \rho)] = \frac{s}{\rho} \frac{\partial u}{\partial t} \quad (8)$$

The detailed form of $D(\varepsilon)$ and each term in the force equation have to be specified; they are crucial elements in the modeling.

This particular model is analyzed in detail by Manoussaki et al. (1996) and by Murray et al. (1998). They took

$$\sigma = \tau \frac{n}{1 + \alpha n^2}$$

where α is a positive parameter: this form reflects the reduction in traction as the cell density becomes large. Tranqui and Tracqui (2000; see earlier references there) have developed experimental techniques to determine the detailed form and estimate the parameters.

The derived biased diffusion $D(\varepsilon)$ is

$$D(\varepsilon) = D_0 \begin{pmatrix} 1 + \frac{\varepsilon_{xx} - \varepsilon_{yy}}{2} & \frac{\varepsilon_{xy} + \varepsilon_{yx}}{2} \\ \frac{\varepsilon_{xy} + \varepsilon_{yx}}{2} & 1 - \frac{\varepsilon_{xx} - \varepsilon_{yy}}{2} \end{pmatrix}$$

where D_0 is the motility coefficient when there is no strain present.

The viscoelastic matrix stress must account for the density and reorientation of the matrix fibers as well as their strength. This was taken to be made up of a viscous and an elastic part given by a Voigt form, namely

$$\sigma = \frac{E(\varepsilon)}{1 + \nu} \left(\varepsilon + \frac{\nu}{1 - 2\nu} \theta I \right) + \left(\mu_1 \frac{\partial \varepsilon}{\partial t} + \mu_2 \frac{\partial \theta}{\partial t} I \right)$$

where E is the strain-dependent elastic modulus and ν is the Poisson ratio, which measures how a strip of gel will contract in one direction when stretched in the transverse direction. μ_1, μ_2 are the shear and bulk viscosities, $\theta = \nabla \cdot u$ is the dilation tensor, and I is the unit tensor.

Intuitively we can see how the mechanism generates the observed complex cellular and matrix patterns. Cells initially uniformly distributed on the matrix (essentially a two-dimensional geometry in the experiments) pull the matrix and hence cause stress lines to appear, resulting in realignment of the matrix fibers. The cells can move more easily along the directions of stress and form regions of higher cell density. These clusters of cells in turn generate higher traction forces because of their higher cell density, which in turn increases the deformation of the matrix and enhances “highways” of matrix along which the cells move more freely. In this way a network of fiber bundles is formed, giving rise to the quasi-hexagonal network observed in experiments (see Figure 5). The formation of these fiber bundles/highways we associate with the process of angiogenesis, or, in the context here, vasculogenesis.

Murray et al. (1998) analyzed the model using parameters estimated from the literature. Figure 5 illustrates some time-evolving network patterns together with one of the typical networks found experimentally by Vernon et al. (1995). Various other hypothetical scenarios were studied that also compared well with subsequent experiments. Clearly, no patterns form if there is no cell traction,

as found experimentally. The case for such a mechanical model of cell-matrix network patterns is further strengthened by the way the patterns are formed in vitro and mathematically. The evolution is from irregular polygons that increase in size and decrease in number, with the smallest polygons pinching off and disappearing. Also, on thicker gels, larger polygons form. They also found, surprisingly, that if the cell traction is sufficiently high, networks formed even in the *absence* of biased diffusion. The effect of matrix thickness on the patterns also reflected what was observed experimentally.

This mechanical model was the first mathematical description of cell-matrix interactions for the formation of network patterns in which all of the component variables were measurable.

Discussion

We have seen how two fundamentally different types of model can give rise to patterns. Here we briefly mention other mechanisms that have been studied.

Negative and Long-Range Diffusion

If, in a reaction-diffusion equation, the diffusion coefficient was negative, this would intuitively cause clumping (imagine viewing a movie of diffusion in reverse). Although such a problem is ill-posed mathematically, this can be rescued by adding a biharmonic term, as in the Cahn-Hilliard equation (Cahn, 1968). Murray (2002, vol. I) shows that a biharmonic term is a natural practical modification when fluxes have a long-range component (they depend on densities in a neighborhood of the reference point). Such long-range effects are present in the Murray-Oster mechanical model via the extracellular matrix and the finger-like filopodia of the cells.

Chemotaxis

Chemotaxis is the name given to the process whereby cells move up or down a chemical (chemoattractant or chemorepellent) gradient. Typically, chemoattractant (c) is secreted and degraded by cells (n), and the cells respond to gradients in c with a convection speed of $\chi(\partial c/\partial x)$, with typical equations, in one dimension, given by

$$\frac{\partial n}{\partial t} = D_n \frac{\partial^2 n}{\partial x^2} - \frac{\partial}{\partial x} \left(n \chi \frac{\partial c}{\partial x} \right), \quad \frac{\partial c}{\partial t} = f(n, c) + D_c \frac{\partial^2 c}{\partial x^2} \quad (9)$$

where χ is the chemotactic parameter (in fact usually a function of c), the D s are diffusion coefficients, and $f(n, c)$ is the source function of c . For spatial patterns to form, cells must diffuse at a lower rate than the chemoattractant, and the chemoattractive (destabilizing) force need only be slightly stronger than the diffusive (stabilizing) force. There is a single dimensionless parameter that determines whether or not a pattern will form.

Bacteria such as *E. coli* and *Salmonella* exhibit strong chemotactic responses and have been the subject of intense study since the early 1990s. The experimental work of Budrene and Berg (1995; see earlier references therein) graphically illustrates the complex regular patterns that can be formed. This work was the basis for the mathematical models that reflect the detailed biology by Tyson and her colleagues, including the experimentalists (Tyson, Lubkin, and Murray 1998; see other references there and Murray, 2002, vol. II, for a full survey). This specific paper shows the minimum requirements a model and the biology must have to produce the observed patterns.

Cross-Taxis

Two species can exhibit taxis with respect to one other:

$$\begin{aligned} \frac{\partial A}{\partial t} &= D_1 \frac{\partial^2 A}{\partial x^2} - \chi_1 \frac{\partial}{\partial x} \left(A \frac{\partial B}{\partial x} \right), \\ \frac{\partial B}{\partial t} &= D_2 \frac{\partial^2 B}{\partial x^2} - \chi_2 \frac{\partial}{\partial x} \left(B \frac{\partial A}{\partial x} \right) \end{aligned} \quad (10)$$

(if $\chi_1 > 0$ and $\chi_2 > 0$, for example, constant here, the two species move up each other's gradient). Although not greatly studied, such models are known to be susceptible to blow-up (see Murray, 2002, vol. II, for references and a brief analysis).

Mathematical Techniques

We have barely scratched the surface of the mathematical analysis of pattern formation. Close to bifurcation (loss of stability), it is possible to analyze the behavior of small-amplitude patterns. For example, one can determine whether two basic spatial patterns, hexagonal and striped, are stable with respect to each other (Murray, 2002, vol. II). We have only briefly confronted the fact that pattern formation generally takes place on a finite domain, perhaps with a particular geometry. At the very least, this reduces the number of modes that can occur: we refer the reader to Murray's (2002, vol. II) modeling of mammalian coat pattern formation as a graphic example. The most important aspect of mathematical models for biological pattern formation must remain, however, their close relation to the real world of biology.

Road Map: Dynamic Systems

Related Reading: Cooperative Phenomena

References

- Budrene, E. O., and Berg, H. C., 1995, Dynamics of formation of symmetrical patterns by chemotactic bacteria, *Nature*, 376:49–53.
- Cahn, J. W., 1968, Spinodal decomposition: The 1967 Institute of Metals Lecture, *Trans. Metall. Soc. AIME*, 242:167–180.
- Folkman, J., 1998, Foreword, in *Vascular Morphogenesis: In Vivo, In Vitro, In Mente* (C. D. Little, V. Mironov, and E. H. Sage, Eds.), Boston: Birkhäuser, pp. vi–ix.
- Harris, A. K., Stopak, D., and Warner, P., 1984, Generation of spatially periodic patterns by a mechanical instability: A mechanical alternative to the Turing model, *J. Embryol. Exp. Morphol.*, 80:1–20.
- Lander, A. D., Nie, Q., and Wan, F. Y. M., 2002, Do morphogen gradients arise by diffusion? *Devel. Cell*, 2:786–796.
- Little, C. D., Mironov, V., and Sage, E. H., Eds., 1998, *Vascular Morphogenesis: In Vivo, In Vitro, In Mente*, Boston: Birkhäuser.
- Manoussaki, D., Lubkin, S. R., Vernon, R. B., and Murray, J. D., 1996, A mechanical model for the formation of vascular networks in vitro, *Acta Biotheoret.*, 44:271–282.
- Murray, J. D., 2002, *Mathematical Biology*, 3rd ed., vol. I: *An Introduction*; vol. II: *Spatial Models and Biomedical Applications*, New York: Springer-Verlag. ♦
- Murray, J. D., Manoussaki, D., Lubkin, S. R., and Vernon, R. B., 1998, A mechanical theory of *in vitro* vascular network formation, in *Vascular Morphogenesis: In Vivo, In Vitro, In Mente* (C. D. Little, V. Mironov, and E. H. Sage, Eds.), Boston: Birkhäuser, pp. 178–188.
- Olsen, L., Maini, P. K., and Sherratt, J. A., 1998, Simple modelling of extracellular alignment in dermal wound healing, *J. Theor. Med.*, 1:175–192.
- Sherratt, J. A., and Dallon, J. C., 2002, Theoretical models of wound healing: Past successes and future challenges, *Compt. Rend. Acad. Sci. (Paris) (Life Sciences)* (in press).
- Swanson, K. R., Alvord, E. C., and Murray, J. D., 2000, A quantitative model for differential motility of gliomas in grey and white matter, *Cell Prolif.*, 33:317–329.
- Swanson, K. R., Alvord, E. C., and Murray, J. D., 2002, Virtual brain tumors (gliomas) enhance the reality of medical imaging and highlight inadequacies of current therapy, *Br. J. Cancer*, 86:14–18. ♦

- Tranqui, L., and Tracqui, P., 2000, Mechanical signalling and angiogenesis: The integration of cell-extracellular matrix couplings, *Compt. Rend. Acad. Sci. (Paris) (Life Sciences)*, 323:31–47.
- Tranquillo, R. T., and Murray, J. D., 1993, Mechanistic model of wound contraction, *J. Surg. Res.*, 55:233–247.
- Turing, A. M., 1952, The chemical basis of morphogenesis, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 237:37–72.

- Tyson, R., Lubkin, S. R., and Murray, J. D., 1998, A minimal mechanism for bacterial patterns, *Proc. R. Soc. Lond. B*, 266:299–304. ♦
- Vernon, R. B., Lara, S. L., Drake, C. J., Iruela-Arispe, M. L., Angello, J. C., Little, C. D., Wight, T. N., and Sage, E. H., 1995, Organized type I collagen influences endothelial patterns during “spontaneous angiogenesis in vitro”: Planar cultures as models of vascular development, *In Vitro Vasc. Dev. Biol.*, 31:120–131.

Pattern Formation, Neural

Paul C. Bressloff and Jack D. Cowan

Introduction

In studying the large-scale functional and anatomical structure of cortex, two distinct questions naturally arise: How did the structure develop and what forms of spontaneous and stimulus-driven neural dynamics are generated by such a cortical structure? It turns out that in both cases, the Turing mechanism for spontaneous pattern formation plays an important role. Turing originally considered the problem of how animal coat patterns develop. He suggested that chemical markers in the skin make up a system of diffusion-coupled chemical reactions among substances called morphogens. Turing showed that in a two-component reaction-diffusion system, a state of uniform chemical concentration can undergo a diffusion-driven instability, leading to the formation of a spatially inhomogeneous state (see PATTERN FORMATION, BIOLOGICAL). Wilson and Cowan (1973) proposed a nonlocal version of this mechanism based on competition between short-range excitation and longer-range inhibition. In the neural context, interactions are mediated not by molecular diffusion, but by long-range axonal connections; hence the term *nonlocal*. Since then, this neural version of the Turing instability has been applied to many problems concerning the dynamics (Bressloff and Cowan, 2002; see also DYNAMICS AND BIFURCATION IN NEURAL NETS) and development (Swindale, 1980) of cortex. In the former case, pattern formation occurs in neural activity; in the latter, it occurs in synaptic weights. In most cases, there exists some underlying symmetry in the model that plays a crucial role in the selection and stability of the resulting patterns.

Feature Selectivity and Tuning

Probably the simplest example of neural pattern formation is that of orientation tuning in the ring model of a cortical hypercolumn (Somers, Nelson, and Sur, 1995). The one-population version of this model consists of a continuous distribution of neural populations labeled by their orientation preference $\phi \in [0, \pi)$. The state of the network is expressed in terms of an activity variable $a(\phi, t)$ evolving according to the equation

$$\tau \frac{\partial a}{\partial t} = -a + w \circ \sigma[a] + h \quad (1)$$

where τ is a time constant, $w \circ \sigma$ signifies the convolution

$$(w \circ \sigma[a])(\phi) = \int_0^\pi w(\phi - \phi') \sigma[a(\phi')] \frac{d\phi'}{\pi} \quad (2)$$

and σ denotes a firing rate function (typically taken to be a smooth sigmoid function of activity). The weight distribution w represents nonlocal neural interactions within the hypercolumn, whereas h denotes an external drive generated by some oriented visual stimulus.

First, consider the case of constant external drive, $h(\phi) = h_0$, and suppose that \bar{a} is a homogeneous fixed-point solution of Equation 1, that is, $\bar{a} = w \circ \sigma[\bar{a}] + h_0$. Linearizing about this fixed point by setting $a(\phi, t) = \bar{a} + u(\phi)e^{2t/\tau}$ leads to the eigenvalue equation

$$(\lambda + 1)u(\phi) = \mu w \circ u(\phi) \quad (3)$$

where $\mu = \sigma'[\bar{a}]$. Expanding the π -periodic functions $u(\phi)$ and $w(\phi)$ as Fourier series

$$w(\phi) = \sum_{n=-\infty}^{\infty} W_n e^{2in\phi}, \quad u(\phi) = \sum_{n=-\infty}^{\infty} U_n e^{2in\phi} \quad (4)$$

we can diagonalize the eigenvalue equation to find the discrete set of eigenvalues

$$\lambda_n = -1 + \mu W_n \quad (5)$$

for integer n , with corresponding eigenfunctions $U_n e^{2in\phi}$. We assume that $w(\phi)$ is a real, symmetric function of ϕ so that $W_{-n} = W_n$ with W_n real. It follows that for sufficiently small μ (corresponding to a low activity state \bar{a}), $\lambda_n < 0$ for all n , and the fixed point \bar{a} is stable. However as μ increases beyond a critical value μ_c , the fixed point becomes unstable, owing to excitation of the eigenfunctions associated with the largest Fourier component of w . We refer to such eigenfunctions as *excited modes*.

Two examples of Fourier spectra are shown in Figure 1. In the first case, W_1 is maximal, so $\mu_c = W_1^{-1}$, and the excited modes are of the form

$$u(\phi) = z e^{2i\phi} + z^* e^{-2i\phi} = |z| \cos(2[\phi - \phi_0]) \quad (6)$$

with complex amplitude $z = |z| e^{-2i\phi_0}$. Since these modes have a single maximum around the ring, the network supports an activity profile consisting of a tuning curve centered about the point ϕ_0 (see Figure 1C). The location of this peak is arbitrary and depends only on random initial conditions, reflecting the rotation invariance or *symmetry* of the weight distribution w . This follows from the assumption that w depends only on the difference between the orientation preferences ϕ and ϕ' as in Equation 2. Such a symmetry is said to be spontaneously broken by the action of the pattern-forming instability. Also note that since the dominant component is W_1 , the distribution w is excitatory (inhibitory) for neurons with sufficiently similar (dissimilar) orientation preferences. (This is analogous to the Wilson-Cowan “Mexican hat” function; see below.) On the other hand, when the inhibitory component is weakened such that W_0 is maximal, the network undergoes a bulk instability at the critical point $\phi_c = W_0^{-1}$ in which no particular orientation ϕ_0 is favored, since the excited eigenmode reduces to U_0 .

So far, we have shown how linear stability analysis can be used to establish the growth of an inhomogeneous state from a homogeneous state through a pattern-forming instability. However, as

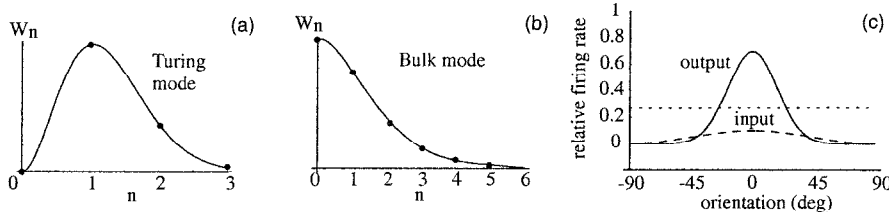


Figure 1. Orientation tuning in the ring model. Fourier spectrum W_n of local weight distribution $w(\phi)$ with (A) a maximum at $n = 1$ (Turing mode) and (B) a maximum at $n = 0$ (bulk mode). C, Tuning curve generated by a pattern-forming instability. Recurrent excitation and inhibition amplifies a weakly biased external input. The dotted line is baseline output without orientation tuning.

the new state increases in amplitude, the linear approximation breaks down, so one has to use nonlinear theory to investigate whether or not a stable pattern ultimately forms. Sufficiently close to the bifurcation point at $\mu = \mu_c$, where the homogeneous state becomes unstable, we can treat $\varepsilon = \mu - \mu_c$ as a small parameter and carry out a perturbation expansion of Equation 1 in powers of $\varepsilon^{1/2}$ (see DYNAMICS AND BIFURCATION IN NEURAL NETS). First, Taylor expand the nonlinear term $\sigma[a]$ about the fixed point \bar{a} :

$$\sigma[a] = \sigma[a_0] + \gamma_1(a - a_0) + \gamma_2(a - a_0)^2 + \gamma_3(a - a_0)^3 + \dots$$

where $\gamma_n = n!^{-1} d^n \sigma[a_0] / da^n$. Second, substitute the series expansion

$$a(\phi, t) = a_0 + \varepsilon^{1/2} [z(t)e^{2im\phi} + z^*(t)e^{-2im\phi}] + O(\varepsilon)$$

into Equation 1 and equate equal powers of $\varepsilon^{1/2}$. A standard perturbation calculation then yields a nonlinear ordinary differential equation at order $\varepsilon^{3/2}$ for the amplitude $z(t)$ of the excited modes (Bressloff and Cowan, 2002):

$$\frac{dz}{dt} = z(\mu - \mu_c - \Lambda|z|^2) \quad (7)$$

where

$$\Lambda = -\frac{3\gamma_3}{\gamma_1^2} - \frac{2\gamma_2^2}{\gamma_1^2} \left[\frac{W_2}{1 - \gamma_1 W_2} + \frac{2W_0}{1 - \gamma_1 W_0} \right] > 0 \quad (8)$$

It follows that if $\mu > \mu_c$, then the fixed point $z = 0$ is unstable, and there is a new branch of stable tuning curves with arbitrary phase and steady-state amplitude $|z| = (\mu - \mu_c)/\Lambda$.

Now suppose that there is a weakly biased external drive of the form

$$h(\phi) = h_0[(1 - \kappa) + \kappa \cos(2(\phi - \Phi))] \quad (9)$$

representing a visual stimulus with orientation Φ . Assuming that $\kappa = O(\varepsilon^{3/2})$, one finds that there is an additional term $h_0\kappa e^{-2i\Phi}$ on the right-hand side of the $O(\varepsilon^{3/2})$ amplitude equation (Equation 7). The phase of the stable inhomogeneous state is now equal to Φ . Hence, when the network is operating in the tuned or *Turing mode*, recurrent interactions amplify a weakly biased input, leading to an orientation tuning curve whose peak coincides with the stimulus orientation; the circular symmetry of the network is now explicitly broken by the presence of a biased input. This amplification mechanism has received recent experimental support in that optical imaging of visual cortical responses in the presence of voltage-sensitive dyes reveals responses sharply tuned for orientation whose amplitude then grows in a manner consistent with the onset of the Turing mode (Sharon and Grinvald, 2002).

It is also possible to extend the above ideas to incorporate other stimulus features for which cortical neurons exhibit tuned responses, including spatial frequency, color, and motion. An interesting question then arises, namely, “What is the appropriate symmetry of the associated pattern forming instability?” For example, it has recently been suggested that orientation and spatial frequency preferences can be combined by replacing the ring network, which

has circular symmetry, by a network with the topology of the surface of a sphere (Bressloff and Cowan, 2002). Motivated by recent optical imaging data on orientation and spatial frequency preference maps (Issa, Trepel, and Stryker, 2000), such feature preferences are represented by angular coordinates on the sphere. (Note that such a coordinate system refers to the feature preferences; it does not imply that the actual distribution and connections of visual cortex neurons fit on a sphere—in fact, they fit on a plane.) High and low spatial frequency preferences are located at the poles of the sphere, which are identified with the singularities of the orientation preference map, commonly referred to as *orientation pinwheels*. Cortical amplification through spontaneous breaking of spherical (rather than circular) symmetry leads to a sharply tuned, contrast-invariant response to both stimulus features.

Geometric Visual Hallucinations

Geometric visual hallucinations are seen by many observers after taking hallucinogens such as LSD, cannabis, or mescaline; on viewing bright flickering lights; on waking up or falling asleep; in “near death” experiences; and in many other syndromes. The Chicago neurologist Klüver organized such images into four groups called *form constants*: (I) tunnels and funnels, (II) spirals, (III) lattices, including honeycombs and triangles, and (IV) cobwebs, all of which contain repeated geometric structures. Figure 2A shows their appearance in the visual field. Note in particular the difference between the first two *noncontoured* images, which consist of alternating regions of light and dark, and the *contoured* nature of the last two images.

Ermentrout and Cowan (1979) provided a first account of the generation of visual hallucinations, based on the idea that some disturbance such as a drug or flickering light can destabilize the primary visual cortex (V1), inducing spontaneous pattern of cortical activity that reflects the underlying architecture of V1. They studied interacting populations of excitatory and inhibitory neurons distributed within a two-dimensional cortical sheet. Modeling the evolution of the network in terms of a set of Wilson-Cowan equations, they showed how spatially periodic patterns such as stripes, squares, and hexagons bifurcate from a low-activity homogeneous state via a Turing instability. They then noted that there is an orderly retinotopic mapping of the visual field onto the surface of cortex, with the left and right halves of visual field mapped onto the right and left cortices, respectively. Except close to the fovea (the center of the visual field), this map can be approximated by a complex logarithm (Schwartz, 1977) as illustrated in Figure 2B. Applying the inverse of this retinocortical map, they showed that when the periodic cortical patterns are mapped back into visual field coordinates, noncontoured hallucinatory images such as the form constants (I) and (II) of Figure 2A are reproduced.

Interestingly, the model cannot reproduce the contoured images (III) and (IV), since there is no information in it regarding the orientation selectivity of neurons in V1. Recently, a much more detailed model of the functional and anatomical structure of V1 has been developed that treats the cortex as a continuum of interacting hypercolumns, each of which has the internal structure of the ring

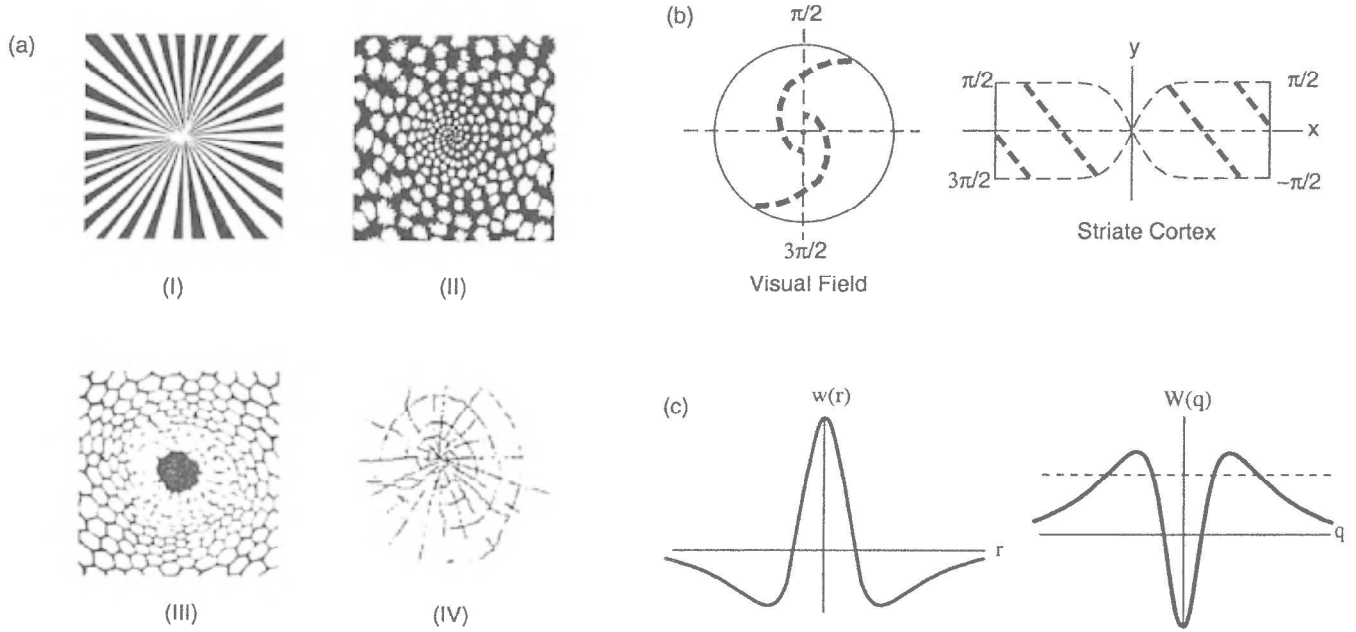


Figure 2. A, Hallucination form constants: (I) funnel and (II) spiral images seen following ingestion of LSD, (III) Honeycomb generated by marijuana, (IV) cobweb petroglyph. B, Retinocortical map showing how a spiral im-

age in visual field is mapped to a stripe of activity in cortex. C, Mexican hat interaction function $w(r)$ showing short-range excitation and long-range inhibition together with its Fourier transform $W(q)$.

model for orientation tuning (Bressloff et al., 2001). In this new model, both contoured and noncontoured hallucinatory images can be generated depending on whether each isolated hypercolumn undergoes a local Turing or bulk instability with respect to orientation (see Figure 1A).

We now consider the theory of cortical pattern formation in more detail. For simplicity, we describe the earlier version due to Ermentrout and Cowan. Let $a_E(\mathbf{r}, t)$ be the activity of excitatory neurons in a given volume element of a slab of neural tissue located at $\mathbf{r} \in \mathbb{R}^2$, and let $a_I(\mathbf{r}, t)$ be the corresponding activity of inhibitory neurons; a_E and a_I can be interpreted as local spatiotemporal averages of the membrane potentials or voltages of the relevant neural populations. When neuron activation rates are low, they can be shown to satisfy nonlinear evolution equations of a similar form to Equation 1:

$$\tau \frac{\partial a_i}{\partial t} = -a_i + \sum_{m=E,I} w_{im} \cdot \sigma[a_m] + h_i \quad (10)$$

where $w \cdot \sigma$ now signifies the convolution

$$(w \cdot \sigma[a])(\mathbf{r}) = \int_{\mathbb{R}^2} w(|\mathbf{r} - \mathbf{r}'|) \sigma[a(\mathbf{r}', t)] d\mathbf{r}' \quad (11)$$

with $w_{im}(|\mathbf{r} - \mathbf{r}'|)$ giving the weight per unit volume of all synapses to the i th population from neurons of the m th population a distance $|\mathbf{r} - \mathbf{r}'|$ away. Note that $w_{IE} > 0$ and $w_{II} < 0$ and the external input h_i is assumed to be constant. An important property of the weight distributions w_{im} is that they are invariant under the action of the planar Euclidean group—the group of rigid motions in the plane consisting of translations, rotations, and reflections. This symmetry plays a crucial role in determining the types of pattern that emerge through a Turing instability.

For a sigmoid firing rate function σ , it can be shown that there exists at least one fixed-point solution \bar{a}_i of Equation 10:

$$\bar{a}_i = \sum_{m=E,I} W_{im} \cdot \sigma[\bar{a}_m], \quad W_{im} = \int_{\mathbb{R}^2} w_{im}(\mathbf{r}) d\mathbf{r} \quad (12)$$

If the external input h_i is sufficiently small relative to the threshold for firing, then this fixed point is unique and stable. There are thus two ways to increase the excitability of the network and thus destabilize the fixed point: either by increasing the external input or reducing the threshold. The latter can occur through the action of drugs on certain brainstem nuclei, which therefore provides a mechanism for generating geometric visual hallucinations. The local stability of the fixed point is found by linearization. Setting $a_i(\mathbf{r}, t) = \bar{a}_i + u_i(\mathbf{r})e^{\lambda t}$ leads to the eigenvalue equation

$$(\lambda + 1)u_i(\mathbf{r}) = \sum_{m=E,I} \mu_m(w_{im} \cdot u_m)(\mathbf{r}) \quad (13)$$

where $\mu_i = \sigma'(\bar{a}_i)$. This can be diagonalized by introducing Fourier transforms $W_{im}(\mathbf{k})$ and $U_m(\mathbf{k})$ and using the convolution theorem. The result is a matrix dispersion relation for λ as a function of $q = |\mathbf{k}|$ given by solutions of the characteristic equation $\det[(\lambda + 1)\mathbf{I} - \Lambda(q)] = 0$, where $\Lambda_{im}(q) = \mu_m W_{im}(|\mathbf{k}|)$ and \mathbf{I} is the unit matrix. One can simplify the formulation by assuming that $w_{EE} = w_{IE}$ and $w_{II} = w_{EI}$ so that the dispersion relation reduces to $\lambda(q) = -1 + \mu W(q)$, where $W(q)$ is the Fourier transform of $w(\mathbf{r}) = [\mu_E w_{EE}(\mathbf{r}) + \mu_I w_{II}(\mathbf{r})]/\mu$.

It is then relatively straightforward to set up the conditions under which the homogeneous state undergoes a Turing instability, namely, that $W(q)$ be *bandpass*. This can be achieved with the “Mexican hat” function shown in Figure 2C, representing short-range excitation and long-range inhibition. It is simple to establish that λ then passes through zero at the critical value $\mu_c = 1/W(q_c)$, signaling the growth of spatially periodic patterns with wave number q_c , where $W(q_c) = \max_q \{W(q)\}$. Close to the bifurcation point, these patterns can be represented as linear combinations of plane waves:

$$u(\mathbf{r}) = \sum_{n=1}^N [z_n e^{i\mathbf{k}_n \cdot \mathbf{r}} + z_n^* e^{-i\mathbf{k}_n \cdot \mathbf{r}}] \quad (14)$$

where the sum is over all wave vectors with $|\mathbf{k}_n| = q_c$. Rotation symmetry implies that the space of such modes is infinite dimen-

sional. That is, all plane waves with wave vectors on the critical circle $|\mathbf{k}_n| = q_c$ are allowed (see Figure 3A). However, translation symmetry means that we can restrict the space of solutions to that of doubly periodic functions corresponding to regular tilings of the plane. The symmetry group is then reduced to that of certain crystal lattices: square, rhomboid, and hexagonal lattices (see Figure 3B). The sum over n in Equation 14 is now finite with $N = 2$ (square, rhomboid) or $N = 3$ (hexagonal), and depending on the boundary conditions, various patterns of stripes or spots can be obtained as solutions. Amplitude equations for the coefficients z_n can then be obtained by using the perturbation approach described in the discussion of orientation tuning. Here the rotation and translation symmetries introduced above restrict the structure of the amplitude equations. In the case of a square or rhombic lattice, we can take $\mathbf{k}_1 = q_c(1, 0)$ and $\mathbf{k}_2 = q_c(\cos \varphi, \sin \varphi)$ such that

$$\frac{dz_n}{dt} = z_n \left[\mu - \mu_c - \gamma_0 |z_n|^2 - 2\gamma_\varphi \sum_{m \neq n} |z_m|^2 \right] \quad (15)$$

for $n = 1, 2$, where γ_φ depends on the angle φ . In the case of a hexagonal lattice, we can take $\mathbf{k}_n = q_c(\cos \varphi_n, \sin \varphi_n)$ with $\varphi_1 = 0$, $\varphi_2 = 2\pi/3$, and $\varphi_3 = 4\pi/3$ such that

$$\begin{aligned} \frac{dz_n}{dt} = z_n & \left[\mu - \mu_c - \gamma_0 |z_n|^2 - \eta z_{n-1}^* z_{n+1}^* \right. \\ & \left. - 2\gamma_{\varphi_2} z_n (|z_{n-1}|^2 + |z_{n+1}|^2) \right] \end{aligned} \quad (16)$$

where $n = 1, 2, 3 \pmod{3}$.

These ordinary differential equations can then be analyzed to determine which particular types of pattern are selected and to calculate their stability. The results can be summarized in a bifurcation diagram as illustrated in Figure 3C for the hexagonal lattice with $\eta > 0$ and $2\gamma_{\varphi_2} > \gamma_0$. (Note that such patterns have also been observed in fluids in the form of convection rolls and honeycombs as well as in animal coat markings in the form of stripes and spots. This indicates that although the physics may be very different, the interactions in all these phenomena are such that they can all be represented within the framework of the Turing mechanism.)

Cortical Development

Essentially the same analysis can be applied to a variety of problems concerning the neural development of feature maps and connectivity patterns (see also OCULAR DOMINANCE AND ORIENTATION COLUMNS). Consider, for example, the development of topographic maps from eye to brain (von der Malsburg and Willshaw, 1977; DEVELOPMENT OF RETINOTECTAL MAPS). Such maps develop by a process that involves both innate and activity-dependent factors. The actual growth and decay of connections are activity dependent, involving synaptic plasticity. However, the final solution is constrained by innate factors in the form of gene products acting as *morphogens* (see PATTERN FORMATION, BIOLOGICAL), which act like boundary conditions. The key insight was provided by von der Malsburg (1973), who showed that pattern formation can occur in a developing neural network whose synaptic connectivity or weight matrix is activity dependent and modifiable, provided that some form of *competition* is present. Thus, Häußler and von der Malsburg (1983) formulated the topographic mapping problem (in the case of a one-dimensional cortex) as follows: let w_{rs} be the weight of connections from the retinal point r to the cortical point s , and let \mathbf{w} be the associated weight matrix. An evolution equation for \mathbf{w} embodying synaptic plasticity and competition can then be written as

$$\frac{d\mathbf{w}}{dt} = \alpha \mathbf{J} + \beta \mathbf{w} \cdot C(\mathbf{w}) - \mathbf{w} \cdot B[\alpha \mathbf{J} + \beta \mathbf{w} \cdot C(\mathbf{w})] \quad (17)$$

where \mathbf{J} is a matrix with all elements equal to unity, $C_{rs}(\mathbf{x}) = \sum_{r's'} c(r - r', s - s') x_{r's'}$, and

$$B(\mathbf{x}) = \frac{1}{2} \left[\frac{1}{N} \sum_{r'} x_{r's} + \frac{1}{N} \sum_{s'} x_{rs'} \right] \quad (18)$$

One can easily show that $\mathbf{w} = \mathbf{J}$ is an unstable fixed point of Equation 17. Linearizing about this fixed point leads to an equation that can be written as

$$\tau \frac{dv}{dt} = -v(r, t) + \tau(I - B)[(I + C)(v)] \quad (19)$$

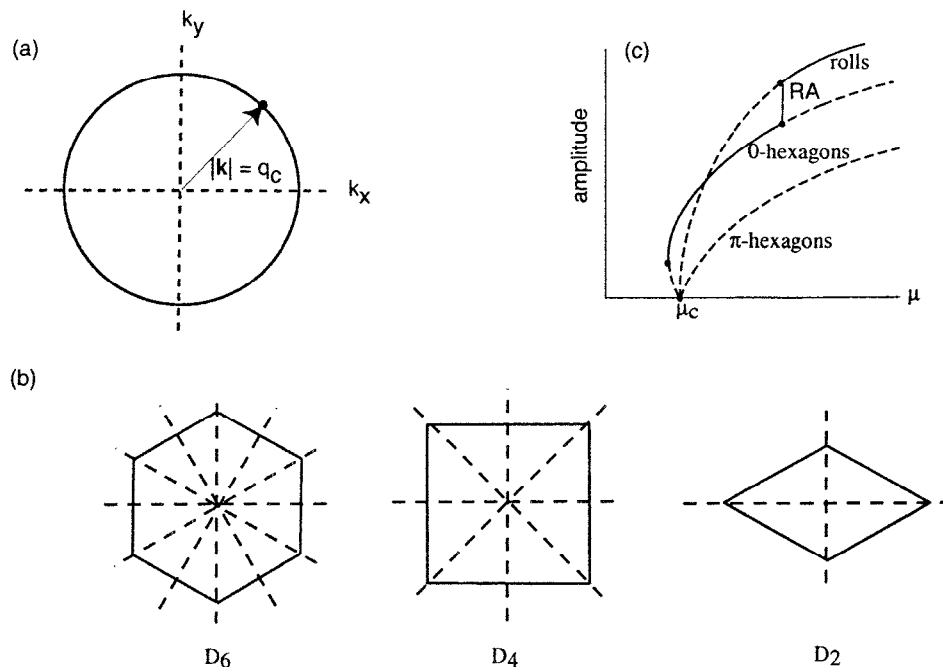


Figure 3. A, Critical circle for Turing instability. B, Crystal lattice groups: hexagonal (D_6), square (D_4), and rhomboid (D_2). C, Bifurcation diagram showing the variation in amplitude C with parameter μ for patterns on a hexagonal lattice. Solid and dashed curves indicate stable and unstable solutions, respectively. The different patterns are distinguished by the coefficients $\mathbf{z} = (z_1, z_2, z_3)$ with $\mathbf{z} = (1, 0, 0)$ for roll or stripe patterns, $\mathbf{z} = (1, 1, 1)$ for 0-hexagons, and $\mathbf{z} = (1, 1, -1)$ for π -hexagons. It is also possible for additional patterns to form through secondary bifurcations (such as rectangular (RA) patterns). However, higher-order contributions to the amplitude equation (Equation 16) are needed to determine such bifurcations.

where $\tau = (1 - \alpha)^{-1}$. It is not too difficult to see that the term $(I - B)[(I + C)(\mathbf{v})]$ is equivalent to the action of an effective convolution kernel of the form $w(\mathbf{r}) = w_+(\mathbf{r}) - w_-(\mathbf{r})$, so Equation 19 can be rewritten in the familiar form:

$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v}(\mathbf{r}, t) + \tau \int_{\mathbb{R}^2} w(\mathbf{r} - \mathbf{r}') \mathbf{v}(\mathbf{r}', t) d\mathbf{r}' \quad (20)$$

where in this case $\mathbf{r} = \{r, s\}$ and \mathbf{v} is a matrix. Once again there is a dispersion relation of the form $\lambda = -1 + \mu W(\mathbf{k}) \equiv \lambda(\mathbf{k})$, where $k = \{k, l\}$, and given appropriate boundary conditions, it is the Fourier transform $W(\mathbf{k})$ of $w(\mathbf{r})$ that determines which of the eigenmodes

$$\sum_{kl} c_{kl} \exp \left[i \frac{2\pi}{N} (kr + ls) \right]$$

emerges at the critical wave number $\mathbf{k}_C = \{k_C, l_C\}$. It can be shown that in the rs plane, $w(\mathbf{r})$ looks like a circular Mexican hat except that the inhibitory surround is in the form of a cross. This forces the eigenmodes emerging from the Turing instability to be diagonal in the rs plane. If \mathbf{k}_C is selected so that only one wave is present, and if the initial conditions or some morphogen favor the NW \rightarrow SE diagonal rather than the NE \rightarrow SW one, then this corresponds to an ordered and correctly oriented retinocortical map. Figure 4 shows details of the emergence of such an eigenmode.

A second example involves the development of ocular dominance maps (Swindale, 1980). Let $n_R(\mathbf{r}, t)$ and $n_L(\mathbf{r}, t)$ be the normalized right and left eye densities of synaptic connections to the visual cortex modeled as a two-dimensional sheet of neurons. Such densities are assumed to evolve according to the equation

$$\frac{\partial u_m(\mathbf{r}, t)}{\partial t} = \sum_{m=R,L} \int_{\mathbb{R}^2} w_{lm}(|\mathbf{r} - \mathbf{r}'|) \sigma[u_m(\mathbf{r}', t)] d\mathbf{r}' \quad (21)$$

where $u_m = \log n_m - \log(1 - n_m)$ so that $\sigma[u_m] = n_m$, and the coupling matrix \mathbf{w} is given by

$$\mathbf{w}(|\mathbf{r}|) = w(r) \begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}.$$

With the additional constraint $n_R + n_L = 1$, Equation 21 reduces to a one-variable form in n_R :

$$\frac{\partial n_R(\mathbf{r}, t)}{\partial t} = \left[2 \int_{\mathbb{R}^2} w(|\mathbf{r} - \mathbf{r}'|) n_R(\mathbf{r}', t) d\mathbf{r}' - \int_{\mathbb{R}^2} w(|\mathbf{r}'|) d\mathbf{r}' \right] \times n_R(\mathbf{r}, t) (1 - n_R(\mathbf{r}, t)) \quad (22)$$

The fixed points of this equation are easily seen to be 0, 1, and 0.5. The first two are stable; however, the third is unstable to small perturbations. Linearizing about this fixed point generates the dispersion relation $\lambda = 0.5W(|\mathbf{k}|)$. Once again, the Fourier transform of the interaction kernel $w(|\mathbf{r}|)$ controls the emergence of the usual eigenmodes, in this case plane waves of the form $\exp(i\mathbf{k} \cdot \mathbf{r})$ in the

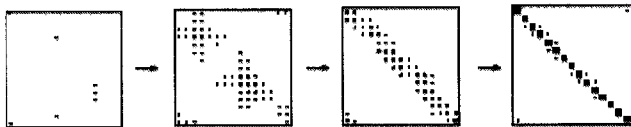


Figure 4. Stages in the development of an ordered and correctly oriented retinotopic map. A single stripe develops in the rs -plane. (Redrawn from Haussler and von der Malsburg, 1983.)

cortical plane. Note that the fixed point $n_R = n_L = 0.5$ corresponds to $u_R = u_L = 0$ and is a point of reflection symmetry of the function $\sigma[u]$. It is this additional symmetry that generates stripes rather than spots or blobs when the fixed point destabilizes.

Discussion

It will be seen that many examples of both spontaneous and stimulus-driven neural pattern formation can be formulated and analyzed within the framework of the Turing instability. Many other examples exist of the role of this instability in visual neuroscience, such as stereopsis (Dev, 1975; Marr and Poggio, 1976; see STEREO CORRESPONDENCE) and the development of iso-orientation patches (Swindale, 1982; see OCULAR DOMINANCE AND ORIENTATION COLUMNS). All such models contain the same basic mechanism of competition between excitation and inhibition, and most have some underlying symmetry that plays a crucial role in the selection and stability of the ensuing patterns. It is an interesting question as to how universal this mechanism is for neural pattern formation.

Road Map: Dynamic Systems

Related Reading: Amplification, Attenuation, and Integration; Cooperative Phenomena; Dynamics and Bifurcation in Neural Nets; Pattern Formation, Biological

References

- Bressloff, P. C., and Cowan, J. D., 2002, Spontaneous pattern formation in primary visual cortex, in *Nonlinear Dynamics: Where Do We Go from Here?* (A. Champneys and S. J. Hogan, Eds.), Bristol, Engl.: IOP. ♦
- Bressloff, P. C., Cowan, J. D., Golubitsky, M., Thomas, P. J., and Wiener, M. C., 2001, Geometric visual hallucinations, Euclidean symmetry, and the functional architecture of striate cortex, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 356:299–330. ♦
- Dev, P., 1975, Perception of depth surfaces in random-dot stereograms: A neural model, *Int. J. Man-Machine Studies*, 7:511–528.
- Ermentrout, G. B., and Cowan, J. D., 1979, A mathematical theory of visual hallucination patterns, *Biol. Cybernetics*, 34:137–150.
- Haussler, A., and von der Malsburg, C., 1983, Development of retinotopic projections: An analytical treatment, *J. Theoret. Neurobiol.*, 2:47–73.
- Issa, N. P., Trepel, C., and Stryker, M. P., 2000, Spatial frequency maps in cat visual cortex, *J. Neurosci.*, 20:8504–8514.
- Marr, D., and Poggio, T., 1976, Cooperative computation of stereo disparity, *Science*, 194:283–287.
- Schwartz, E., 1977, Spatial mapping in the primate sensory projection: Analytic structure and relevance to projection, *Biol. Cybernetics*, 25:181–194.
- Sharon, D., and Grinvald, A., 2002, Dynamics and constancy in cortical spatiotemporal patterns of orientation processing, *Science*, 295:512–515.
- Somers, D. C., Nelson, S., and Sur, M., 1995, An emergent model of orientation selectivity in cat visual cortex simple cells, *J. Neurosci.*, 15:5448–5465.
- Swindale, N. V., 1980, A model for the formation of ocular dominance stripes, *Proc. R. Soc. Lond. B Biol. Sci.*, 208:243–264.
- Swindale, N. V., 1982, A model for the formation of orientation columns, *Proc. R. Soc. Lond. B Biol. Sci.*, 215:211–230.
- von der Malsburg, C., 1973, Self-organization of orientation-selective cells in striate cortex, *Kybernetik*, 14:85–100.
- von der Malsburg, C., and Willshaw, D., 1977, How to label nerve cells so that they can interconnect in an ordered fashion, *Proc. Natl. Acad. Sci. USA*, 74:5176–5178.
- Wilson, H. R., and Cowan, J. D., 1973, A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue, *Kybernetik*, 13:55–80. ♦

Pattern Recognition

Yann LeCun and Yoshua Bengio

Introduction

Pattern recognition (PR) addresses the problem of classifying objects, often represented as vectors or as strings of symbols, into categories. The difficulty is to synthesize, and then to efficiently compute, the *classification function* that maps objects to categories, given that objects in a category can have widely varying input representations. In most instances, the task is known to the designer through a set of example patterns whose categories are known, and through general, a priori knowledge about the task, such as “the category of an object is not changed when the object is slightly translated or rotated in space.”

Historically, the field of PR started with the early efforts in neural networks (perceptrons, adalines, etc.; see PERCEPTRONS, ADALINES, AND BACKPROPAGATION). Whereas in the past, neural networks (NNs) sometimes played the role of an outsider in PR, recent progress in learning algorithms and the availability of powerful hardware have made them the method of choice for many PR applications.

Because most PR problems are too complex to be solved entirely by handcrafted algorithms, machine learning has always played a central role in PR. Learning automatically synthesizes a classification function from a set of labeled examples. Unfortunately, no learning algorithm can be expected to succeed unless it is guided by prior knowledge. The traditional way of incorporating knowledge about the task is to divide the recognizer into a feature extractor and a classifier. Since most learning algorithms work better in low-dimensional spaces with easily separable patterns, the role of the feature extractor is to transform the input patterns so that they can be represented by low-dimensional vectors, or short strings of symbols, that (1) can be easily compared or matched, and (2) are relatively invariant to transformations that do not change the nature of the input objects. The feature extractor contains most of the prior knowledge and is rather specific to the task. It also requires most of the design effort because it is often handcrafted, although unsupervised learning methods, such as PRINCIPAL COMPONENT ANALYSIS (q.v.), can sometimes be used. The classifier, on the other hand, is often general purpose and trainable. One of the main problems with this approach is that the recognition accuracy is largely determined by the ability of the designer to come up with an appropriate set of features. This turns out to be a daunting task which, unfortunately, must be redone for each new problem.

One of the main contributions of NNs to PR has been to provide an alternative to this design: properly designed multilayer networks can learn complex mappings in high-dimensional spaces without requiring complicated handcrafted feature extractors. Networks containing hundreds of inputs and tens of thousands of parameters can be trained on databases containing several hundreds of thousands examples. This allows designers to rely more on learning and less on detailed engineering of feature extractors. Crucial to success is the ability to tailor the network architecture to the task, which allows incorporating prior knowledge and therefore learning complex tasks without requiring excessively large networks and training sets.

The success of multilayer networks relies on one surprising fact: gradient-based minimization techniques can be used to learn very complex nonlinear mappings. Generalizations of the concept of gradient-based learning have allowed one to view many PR techniques, neural and nonneural, in a unified way, including not only traditional multilayer feedforward nets with sigmoid units and dot

products but also many other structures such as radial basis functions, hidden Markov models (HMMs), vector quantizers, etc. Many recent efforts have been directed toward combining adaptive modules of different types into a single system and training them cooperatively by propagating gradients through them, particularly for recognizing composite objects such as handwritten or spoken words (see LeCun et al., 1998, for a review and applications).

Learning and Generalization

Owing to the presence of noise, the high dimension of the input, and the complexity of the mapping to be learned, PR applications create some of the most challenging problems in machine learning. Most learning methods are trained by minimizing a *cost function* computed over a set of training examples. The cost function is generally of the form

$$C(W) = \sum_X Q(X, F(X, W)) + H(W) \quad (1)$$

where X is a training example, $F(X, W)$ is the recognizer output for pattern X and “parameters” W , $Q(X, F(X, W))$ is a cost function (the training error), and $H(W)$ is a measure of “capacity” of the recognizer (the *regularizer*). Such cost functions attempt to model the real measure of performance, i.e., the *testing error* (error rate on a test set disjoint from the training set) (see LEARNING AND GENERALIZATION: THEORETICAL BOUNDS).

System designers have to strike the right balance between learning the training set (by using powerful learning architectures) and minimizing the difference between the training error and the test error (by limiting the capacity of the machine). Large machines can learn the training set but may perform poorly if the training set is not large enough, a problem known as overparameterization, or overfitting. On the other hand, too little capacity yields underfitting—i.e., large error on both training and test sets.

Most adaptive recognizers stand between two extremes of a continuous spectrum. At one end, parameter-based methods, in which a set of learned parameters determines the input-output relation, put the emphasis on minimizing the first term in Equation 1 with a fixed H (e.g., multilayer neural networks). At the other end, memory-based methods, which rely on matching or comparing the incoming pattern with a set of learned or stored prototypes, keep the first term close to zero, and attempt to minimize the regularizer (e.g., nearest-neighbor algorithms).

Although in principle, any appropriate functional form for F , Q , and H can be used, the choice is largely determined by (1) the belief that it is well suited to the task and (2) the efficiency of the available minimization algorithms. There is a strong incentive to choose smooth and well-behaved functions whose gradient can be computed easily, so that gradient-based minimization algorithms can be used, as opposed to inefficient combinatorial search methods. Preferably, F will be a smooth real-valued function (e.g., layers of sigmoid units) rather than a discrete function (e.g., layers of threshold units); Q is often chosen to be the mean square error between the actual output and a target, rather than the number of misclassified patterns, which would be more relevant but is practically impossible to minimize.

A Few Basic Classification Methods

Linear and Polynomial Classifiers

A linear classifier is essentially a single neuron. An elementary two-class discrimination is performed by comparing the output to

a threshold (multiple classes use multiple neurons). Training algorithms for linear classifiers are well studied (see PERCEPTRONS, ADALINES, AND BACKPROPAGATION). Their limitations are well known: the likelihood that a partition of P vectors of dimension N is computable by a linear classifier decreases very quickly as P increases beyond N (Duda and Hart, 1973). One method to ensure separability is to represent the patterns by high-dimensional vectors (large N). If necessary, the dimension of original input vectors can be enlarged using a set of basis functions ϕ_i :

$$F(X, W) = \sum_i w_i \phi_i(X) \quad (2)$$

A simple example is when the basis functions are cross-products of K or fewer coordinates of the input vector X (F is a polynomial of degree K). Such polynomial classifiers have been studied since the early 1960s and have been "renamed" in the context of NNs as sigma-pi units or high-order nets. Unfortunately, polynomial classifiers are often impractical because the number of features scales like N^K . Nevertheless, *feature selection* methods can be used to reduce the number of product terms, or to reduce the number of original input variables.

Local Basis Functions

Another popular kind of space expansion (Equation 2) uses *local* basis functions, which are activated within a small area of the input space. A popular family consists of the radial basis functions (RBFs; see RADIAL BASIS FUNCTION NETWORKS): $\phi_i(X) = e^{-(X - P_i)^2}$, where the P_i are a set of appropriately chosen "prototypes." Methods based on such expansions can cover the full spectrum between parameter-based and purely memory-based methods by varying the number of prototypes, the way they are computed, and the classifier that follows the expansion (which can be more complex than a simple weighted sum). At one extreme, each training sample is used as a prototype, to which the sample's label is attached. In the K -nearest neighbors algorithms, the K nearest prototypes to an unknown pattern vote for its label. In the Parzen windows method, the normalized sum of all the $\phi_i(X)$ associated with a particular class is interpreted as the conditional probability that X belongs to that class (Duda and Hart, 1973). In the RBF method the output is a (learned) linear combination of the outputs of the basis functions. Associating a prototype with each training sample can be very inefficient and increases the complexity term. Therefore, several methods have been proposed for *learning* the prototypes. One way is to use unsupervised clustering techniques such as K -means to put prototypes in regions of high sample density, but supervised methods can also be used (see RADIAL BASIS FUNCTION NETWORKS). An important one is LVQ2, in which prototypes that are near a training sample are moved away from it if its assigned class differs from the sample's, and moved toward it if its class is equal to the sample's (see LEARNING VECTOR QUANTIZATION). Another important supervised method for RBF networks is simply gradient descent. The partial derivatives of the cost function with respect to the parameters of the basis functions (the prototype vectors) can be computed using a form of backpropagation: in the same way that gradients can be backpropagated through sigmoids and dot products, they can be backpropagated through exponentials and Euclidean distances. The parameters can then be adjusted using the gradient. It has been argued that the local property leads to faster learning than standard multilayer nets, and to good rejection properties (Lee, 1991). Several authors enhance the power of prototype-based systems by using distance measures that are more complex than just Euclidean distance between the prototypes and the input patterns (such as general bilinear forms with learned coefficients). Methods that add prototypes as needed have also been proposed, notably the RCE algorithm.

Support Vector Machines

A recently proposed and elegant way of avoiding the curse of dimensionality in polynomial and local classifiers rests on the fact that, if the w_i in Equation 2 are computed to maximize the *margin* (the minimum distance between training points and the classification surface), the W obtained after training can be written as a linear combination of a small subset of the expanded training examples (Boser, Guyon, and Vapnik, 1992). Points in this subset are called *support vectors*, hence the name support vector machines (SVMs). This leads to a surprisingly simple way of evaluating high-degree polynomials in high-dimensional spaces without having to explicitly compute all the terms of the polynomial. For example, maximum-margin polynomials of degree K can be computed using

$$F(X) = \sum_{j \in S} \alpha_j (X \cdot P_j + 1)^K \quad (3)$$

where the P_j are the support points (subset of the training set) and the α_j are coefficients that uniquely determine the weights W . Learning the α_j amounts to solving a quadratic programming problem with linear inequality constraints. Besides polynomial kernels, SUPPORT VECTOR MACHINES (q.v.) can be built for a variety of kernels, such as RBFs and neuron-like kernels. SVMs have also been extended beyond classification problems to regression and density estimation. Excellent results for the classification of handwritten digit images have been obtained with a fourth-degree polynomial computed using this method (Bottou et al., 1994). The number of multiply-adds per recognition was a few hundred thousands, much less than the $O(400^4)$ multiply-adds required to directly evaluate the polynomial.

Complex Distance Measures

Although many memory-based methods use simple distance measures (Euclidean distance) and large collections of prototypes, some applications can take advantage of more complex, problem-dependent, distance measures and use fewer prototypes. Ideal distance measures should be invariant with respect to transformations of the patterns that do not change their nature (e.g., translations and distortions for characters, time or pitch distortion for speech). With invariant distances, a single prototype can potentially represent many possible instances of a category, reducing the number of necessary prototypes. An important family of invariant distance measures entails *elastic matching*. Elastic matching comes down to finding the point closest to the input pattern on the surface of all possible deformations of the prototype. Naturally, the exhaustive search approach is prohibitively expensive in general. However, if the surface is smooth, better search techniques can be used, such as gradient descent or conjugate gradient. If the deformations are along one dimension (as in speech), dynamic programming can find the best solution efficiently. Simard, LeCun, and Denker (1993) approximated the surface of a deformed prototype by its tangent plane at the prototype. The matching problem reduces to finding the minimum distance between a point and a plane, which can be done efficiently. This has been applied to handwritten character recognition with great success.

Multilayer Networks and Gradient-Based Learning

The vast majority of applications of NNs to PR are based on multilayer feedforward networks trained with backpropagation. At first it seems almost magical that an algorithm as simple as gradient descent works at all to learn complex nonlinear mappings (non-convex, ill-conditioned error surfaces). Minsky and Selfridge's warning about the limitations of "hill-climbing" methods for machine learning in the late 1950s is an indication of the general belief

that it could not work. Surprisingly, experiments show that local minima are rarely a problem with large networks. As evidence of the success of backpropagation, all but two of the entries in the last NIST character recognition competition used some form of backpropagation network.

PR problems are often characterized by large and redundant training sets with high-dimensional inputs, which translates into large networks and long learning times. Much effort has been devoted to speeding up training using refined nonlinear optimization methods (conjugate gradient, quasi-Newton methods, and so on). These are essentially batch methods (the weights are updated after a complete pass through the training set) and can rarely compete with “carefully tuned” stochastic (on-line) gradient descent (where the weights are updated after each pattern presentation). This is due to the presence of redundancy in large natural training sets. On typical large-scale image or speech recognition tasks, stochastic gradient descent converges in one to a few dozen epochs. To avoid overlearning, a validation set should be set aside, and training should be stopped when the error rate on the validation set stops decreasing. An important limitation to the popularity of NN techniques for PR is that certain simple tricks must be used and many common pitfalls must be avoided that are part of the “oral culture” rather than scientific facts.

Once backpropagation with feedforward networks of sigmoid units and dot products established the value of gradient-based learning, it seemed natural to extend the idea to other structures. Minimizing a cost function through gradient-based learning can be seen as the unifying principle behind many methods: RBFs or mixtures of Gaussians, learning vector quantization, HMMs, and many prototype-based methods that use various distance measures. Experiments have shown the advantage of using different types of modules in different parts of a learning system. In particular, sigmoids and dot products seem better for processing large amounts of high-dimensional and low-level information (early feature extraction), while RBFs or other more local modules seem better suited for final classification, a more memory-intensive task. With the gradient-based learning framework, modules of different types can be connected in any configuration and trained cooperatively by backpropagating gradients through them. To achieve this, one needs only to be able to compute the partial derivatives of each output of a module with respect to each input and each parameter of the module (see MODULAR AND HIERARCHICAL LEARNING SYSTEMS). In addition, many cost functions can be considered as just another module (with a scalar output) through which gradients can be backpropagated. Examples include the mean squared error, modified LVQ cost functions, maximum likelihood, maximum mutual information, cross-entropy, classification figure of merit, and several types of statistical or graphical postprocessors (LeCun et al., 1998).

Local/Global and Modular Methods

It has recently been suggested that good PR systems should behave differently in different parts of the input space. For example, parts of the input space may be very sparsely populated, requiring a low-capacity learner, while denser areas may require a more complex one. A simple idea is to use a collection of modules, each of which is activated when the input lies in a particular region. A separate module called a *gater* decides which module should be activated. When the gater is differentiable, the whole system (modules plus gater) can be trained cooperatively (see MODULAR AND HIERARCHICAL LEARNING SYSTEMS). In such multimodular systems, parameters are relatively decoupled across modules, which is believed to allow faster training (or better scaling of training time). In another interesting “semilocal” method, a simple network (e.g., single layer) is trained each time a new test pattern is presented, using

training patterns in the neighborhood of this test pattern; training is done “on demand” during recognition (Bottou and Vapnik, 1992).

In general, local methods learn fast, but they are expensive at runtime in terms of memory and, often, of computation. In addition, they may not be appropriate for problems with high-dimensional inputs. Global methods, such as multilayer networks, take longer to train, but they are quite compact, and they execute quickly. They can handle high-dimensional inputs, particularly when specialized architectures are used.

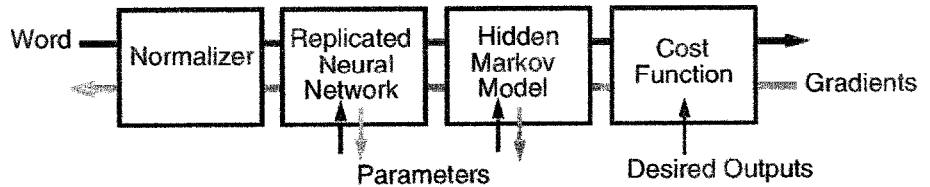
Specialized Architectures, Convolutional Networks

The great hope that multilayer networks brought with them was the possibility of eliminating the need for a separate handcrafted feature extractor, relying on the first layers to automatically learn the right set of features. Although fully connected networks fed with “raw” character images (or speech spectra) have very large numbers of free parameters, they have been applied with some success (Martin and Pittman, 1991). This can be explained as follows. With small initial weights, a multilayer network is almost equivalent to a single-layer network (each layer is quasilinear). As incremental learning proceeds, the weights gradually increase, thereby progressively increasing the effective capacity of the system (to the authors’ knowledge, this explanation was first suggested by Léon Bottou in 1988).

On the other hand, using a specialized network architecture instead of a fully connected net can reduce the number of free parameters and facilitate the learning of invariances. In certain applications, the need for a separate handcrafted feature extractor can be eliminated by wiring the first few layers of the network in a way that forces it to learn relevant features and eliminate irrelevant variability. Convolutional networks, including time-delay neural networks (TDNNs), are an important class of specialized architectures well suited for dealing with one- or two-dimensional signals such as time series, images, or speech (see CONVOLUTIONAL NETWORKS FOR IMAGES, SPEECH, AND TIME SERIES). Convolutional networks use the techniques of local receptive fields, shared weights, and subsampling (loosely based on the architecture of the visual cortex) to ensure that the first few layers extract and combine local features in a distortion-invariant way. Although the wiring of the convolutional layers is designed by hand, the values of all the coefficients are learned with a variant of the backpropagation algorithm. The main advantage of this approach is that the feature extractor is totally integrated into the classifier and is produced by the learning process rather than by the hand of the designer (LeCun et al., 1990). Because of the weight-sharing technique, the number of free parameters in a convolutional network is much less than in a fully connected network of comparable power, which has the effect of reducing the complexity term in Equation 1 and improving the generalization. The success of convolutional nets of various types has had a major impact on several application domains, among them speech recognition, character recognition, and object spotting. On handwriting recognition tasks they compare favorably with other techniques (Bottou et al., 1994) in terms of accuracy, speed, and memory requirements. Character recognizers using convolutional nets have been deployed in commercial applications. A very promising feature of convolutional nets is that they can be efficiently replicated, or scanned, over large input fields, resulting in the so-called *space displacement neural net* (SDNN) architecture (see below, and CONVOLUTIONAL NETWORKS FOR IMAGES, SPEECH, AND TIME SERIES).

Networks with recurrent connections can be used to map input sequences to output sequences, while taking long-term context into account. The main advantage of recurrent networks over TDNNs for analyzing sequences is that the span of the temporal context

Figure 1. A multimodule architecture combining a convolutional NN with a postprocessor such as an HMM.



that the network can take into account is not hard-wired within a fixed temporal window by the architectural choices but can be learned by the network. However, theoretical and practical hurdles (Bengio, Simard, and Frasconi, 1994) limit the span of long-term dependencies that can be learned efficiently.

Recognition of Composite Objects

In many real applications, the difficulty is not only to recognize individual objects but also to separate them from context or background. For example, one approach to handwritten word recognition is to *segment* the characters out of their surrounding and *recognize* them in isolation. A typical handwritten word recognizer uses heuristics to form multiple, possibly overlapping character candidates by cutting the word or by joining nearby strokes. Then the recognizer must either classify each candidate as a character or reject it as a noncharacter. In many applications, such as cursive handwriting or continuous speech, it is difficult or even impossible to devise robust segmentation heuristics. One approach to avoid explicit segmentation is simply to scan the recognizer over all possible locations on the input (character string or spoken sentence) and collect the sequence of corresponding recognizer outputs. Although this is very computationally expensive in general, replicated convolutional networks (SDNNs or TDNNs) can be used to do that very efficiently. In the case of handwriting recognition, an SDNN output will contain a well-identified label when centered on a character. Between characters, the output should indicate a reject. However, combinations of off-center characters may cause ambiguous outputs (e.g., *cl* labeled as *d*). Since both methods, explicit segmentation and scanning, generate many extraneous candidates, a postprocessor is required to resolve ambiguities and pull out the most consistent interpretation, retaining genuine characters and rejecting erroneous stroke combinations, possibly taking linguistic constraints into account (a lexicon or grammar) Figure 1. For this to succeed, the recognizer must be trained not only to classify characters but also to reject noncharacters. The search for the best interpretation is easily done within the framework of graph transducers (LeCun et al., 1998), which generalize HMMs. A graph is built in which each path corresponds to a possible interpretation of the input and in which each node is given probabilities of matching recognizer outputs. Dynamic programming can be used to find the path of highest probability, which yields the most likely interpretation. Furthermore, it is possible to backpropagate errors through this graph in order to train the system to maximize the a posteriori probability of the correct sequence of labels.

Multimodule Architectures and Cooperative Training

Such combinations of neural networks and HMMs (or other graph-based postprocessors) have been proposed by several authors, mostly for speech recognition (see SPEECH RECOGNITION TECHNOLOGY), but also for handwriting recognition (see LeCun et al., 1998, and CONVOLUTIONAL NETWORKS FOR IMAGES, SPEECH, AND TIME SERIES).

The main technical difficulty is in training such hybrid systems. Training the recognizer exclusively on presegmented characters is

neither sufficient nor always possible, since (1) the recognizer must be trained to reject noncharacters and (2) in many cases, such as cursive handwriting, segmented characters are not available, only whole words are. The solution is to simultaneously train the recognizer and the postprocessor to minimize an error measure at the *word level*. This means being able to backpropagate gradients through the postprocessor, down to the recognizer, or to generate desired outputs for the recognizer using the best path in the graph (see Franzini, Lee, and Waibel, 1990, and SPEECH RECOGNITION TECHNOLOGY). Simultaneous training of such hybrids has been reported to yield large reductions in error rates over independent training of the modules in speech recognition (for TDNN/dynamic time warping, see Driancourt, Bottou, and Gallinari, 1991, and Haffner, Franzini, and Waibel, 1991; for TDNN/HMM, see Bengio et al., 1992), and on-line handwriting recognition (for SDNN/HMM, see Bengio, LeCun, and Henderson, 1994). See LeCun et al. (1998) for a review and applications to document analysis.

Discussion

Neural networks, particularly multilayer backpropagation NNs, provide simple yet powerful and general methods for synthesizing classifiers with minimal effort. However, most practical systems combine NNs with other techniques for pre- and postprocessing. On isolated character recognition tasks, multilayer nets trained with variants of backpropagation have approached human accuracy, at speeds of about 1,000 characters per second using NN hardware. NNs have allowed workers to minimize the role of detailed engineering and maximize the role of learning. Despite the recent advances in multimodule architectures and gradient-based learning, several key questions are still unanswered, and many problems are still out of reach. How much has to be built into the system, and how much can be learned? How can one achieve true transformation-invariant perception with NNs? Convolutional nets are a step in the right direction, but new concepts will be required for a complete solution (see DYNAMIC LINK ARCHITECTURE). How to recognize compound objects in their context? The accuracy of the best NN/HMM hybrids for written or spoken sentences cannot even be compared with human performance. Topics such as the recognition of three-dimensional objects in complex scenes are totally out of reach. Human-like accuracy on complex PR tasks such as handwriting and speech recognition may not be achieved without a drastic increase in the available computing power. Several important questions may simply resolve themselves with the availability of more powerful hardware, allowing the use of brute-force methods and very large networks.

Road Map: Learning in Artificial Networks

Related Reading: Concept Learning; Feature Analysis; Perceptrons, Adalines, and Backpropagation; Statistical Mechanics of On-line Learning and Generalization

References

- Bengio, Y., Simard, P., and Frasconi, P., 1994, Learning long-term dependencies with gradient descent is difficult, in *Recurrent Neural Networks* (special issue), *IEEE Trans. Neural Netw.*, 5:157–166.

- Bengio, Y., LeCun, Y., and Henderson, D., 1994, Globally trained handwritten word recognizer using spatial representation, space displacement neural networks and hidden Markov models, in *Advances in Neural Information Processing Systems 6* (J. Cowan, G. Tesauro, and J. Alspector, Eds.), San Mateo, CA: Morgan Kaufmann, pp. 937–944.
- Bengio, Y., Mori, R. D., Flammia, G., and Kompe, R., 1992, Global optimization of a neural network-hidden Markov model hybrid, *IEEE Trans. Neural Netw.*, 3:252–259.
- Boser, B., Guyon, I., and Vapnik, V., 1992, An algorithm for optimal margin classifiers, in *Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, pp. 144–152.
- Bottou, L., 1998, Online algorithms and stochastic approximations, in *Online Learning in Neural Networks* (D. Saad, Ed.), Cambridge, U.K.: Cambridge University Press. ♦
- Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L., LeCun, Y., Muller, U., Sackinger, E., Simard, P., and Vapnik, V., 1994, Comparison of classifier methods: A case study in handwritten digit recognition, in *Proceedings of an International Conference on Pattern Recognition*, Los Alamitos, CA: IEEE Computer Society Press.
- Bottou, L., and Vapnik, V., 1992, Local learning algorithms, *Neural Computat.*, 4:888–900. ♦
- Driancourt, X., Bottou, L., and Gallinari, P., 1991, Learning vector quantization, multi-layer perceptron and dynamic programming: Comparison and cooperation, in *Proceedings of an International Joint Conference on Neural Networks*, vol. 2, Piscataway, NJ: IEEE Press, pp. 815–819.
- Duda, R., and Hart, P., 1973, *Pattern Classification and Scene Analysis*, New York: Wiley. ♦
- Franzini, M., Lee, K., and Waibel, A., 1990, Connectionist Viterbi training: A new hybrid method for continuous speech recognition, in *Proceedings of an International Conference on Acoustics, Speech and Signal Processing*, Piscataway, NJ: IEEE Press, pp. 425–428.
- Haffner, P., Franzini, M., and Waibel, A., 1991, Integrating time alignment and neural networks for high performance continuous speech recognition, in *Proceedings of an International Conference on Acoustics, Speech and Signal Processing*, Piscataway, NJ: IEEE Press, pp. 105–108.
- LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L., 1990, Handwritten digit recognition with a back-propagation network, in *Advances in Neural Information Processing Systems 2* (D. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 396–404.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., 1998, Gradient-based learning applied to document recognition, *Proc. IEEE*, 86(11):2278–2324.
- Lee, Y., 1991, Handwritten digit recognition using K nearest neighbor, radial-basis function, and backpropagation neural network, *Neural Computat.*, 3:441–449.
- Martin, G., and Pittman, J., 1991, Recognizing hand-printed letters and digits using back-propagation learning, *Neural Computat.*, 3:258–267.
- Minsky, M., and Selfridge, O. G., Learning in random nets, in *Information Theory* (C. Cherry, Ed.), London: Butterworths, pp. 335–347.
- Simard, P., LeCun, Y., and Denker, J., 1993, Efficient pattern recognition using a new transformation distance, in *Advances in Neural Information Processing Systems 5* (S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds.), San Mateo, CA: Morgan Kaufmann, pp. 50–58.

Perception of Three-Dimensional Structure

James T. Todd

Introduction

Human observers have a remarkable ability to perceive the three-dimensional (3D) layout of the environment from patterns of light that project onto the retina. Were it not for the facts of our day-to-day experiences, it would be tempting to conclude that the perception of 3D form is a mathematical impossibility, because the properties of optical stimulation appear to have so little in common with the properties of real objects encountered in nature. Whereas real objects exist in 3D space and are composed of tangible materials, an optical image of an object is confined to a two-dimensional (2D) projection surface and consists of nothing more than flickering patterns of light. Nevertheless, for many animals, including humans, these seemingly uninterpretable patterns of light are the primary source of sensory information about the layout of surfaces in the surrounding environment.

There are many different aspects of optical stimulation that are known to provide perceptually salient information about 3D form. Some of these properties—the so-called pictorial depth cues—are available within individual static images. These include texture gradients, contour configurations, and patterns of shading. Others are defined by systematic transformations among multiple images, including the disparity between each eye's view in binocular vision and the optical deformations that occur when objects are observed in motion.

There are two important issues that need to be considered in evaluating any computational model of 3D form perception. The first of these issues involves how 3D structure is perceptually represented. After all, in order to compute an object's shape from visual information, one must first define precisely what shape is. There are numerous attributes of 3D structure that could potentially be represented by the visual system (e.g., curvature, relative depth, local orientation), and the relative computational complexity of an-

alyzing these different attributes can vary dramatically. It is much more difficult, for example, to determine the precise euclidean distance between a pair of visible points than to merely assess which point is closer in depth.

A second related issue concerns ambiguities that are inherent in the nature of visual information. The primary goal of all 3D vision problems is to invert (or partially invert) a function of the following form: $\Lambda = f(\phi)$, where ϕ is the space of environmental properties that can influence patterns of ambient light and Λ is the space of measurable image properties at a point of observation. What can make these problems so difficult is that this is a many-to-one mapping: for any give pattern of optical structure, there is often an infinity of possible 3D structures that could potentially have produced it.

For many problems in 3D vision, it can be useful to separate the properties of environmental structure into two distinct categories. There are some aspects of environmental structure that are invariant over the transformation $\Lambda = f(\phi)$. These properties can be determined with relative ease by measuring the appropriate relationships within available image data. Other properties that are not invariant over this transformation are much more difficult to estimate. Because these properties are inherently ambiguous, a computational analysis must restrict the set of possible interpretations by assuming the existence of environmental constraints. Unfortunately, many of the constraints that have been employed for this purpose seem to have been adopted more for their mathematical convenience than for their ecological validity. The problem with this approach is that the resulting analyses of 3D form may only function effectively within narrowly defined contexts, which have a small probability of occurrence in the natural environments of real biological organisms.

This article reviews various computational models that have been proposed in the literature for analyzing an object's 3D struc-

ture from different types of optical information, both individually and in combination. It also examines how the performance of these models compares with the capabilities and limitations of actual human observers in judging different aspects of 3D structure under varying viewing conditions. The goal is to identify the specific representations and computational mechanisms by which 3D form is perceptually analyzed within the human visual system.

Shape from Shading

The most basic type of information available to any visual system is the light that stimulates different regions of the retina from illuminated surfaces in the environment. Smooth gradations in surface luminance are often referred to as shading, and they are one of the primary cues used by artists for the pictorial representation of 3D form. The analysis of image shading is made especially difficult because the luminance of any visible surface region can depend on several factors, including the positions and spectral composition of its sources of illumination, the local reflectance and orientation of the surface, and the position of the observer. In order to compute shape from shading it is necessary to somehow decompose these different factors. Most existing algorithms for analyzing 3D shape from shading achieve this decomposition by making several strong assumptions to constrain the structure of an observed scene (Horn and Brooks, 1989). It is typically assumed, for example, that a scene is composed of smoothly curved surfaces that have a known uniform reflectance function, with no specular components, and that there is a uniform pattern of illumination with a known direction and spectral composition.

An important limitation on theoretical analyses of 3D shape from shading is that an image of a surface with homogeneous lambertian reflectance has an infinity of possible 3D interpretations that are all related by an affine transformation (Belhumeur, Kriegman, and Yuille, 1999). Thus, a pattern of image shading provides sufficient information to specify the affine properties of an object, such as the parallelism of local surface patches or relative distance intervals in parallel directions, but it does not allow a unique determination of metric properties involving relative distance intervals in different directions. A similar ambiguity is also evident in judgments of 3D shape from shading by human observers. Recent psychophysical experiments have shown that these judgments are often idiosyncratic and task dependent (Koenderink et al., 2001), such that the correlations between observers or response tasks can in some cases be close to zero. Despite these variations in judged metric structure, however, the affine properties of perceived shape from shading have a high degree of reliability.

A fundamental assumption of almost all existing models of the perception of 3D shape from shading is that illumination and reflectance remain constant throughout a scene, such that all variations in shading can be attributed to the geometry of an observed surface. The problem with this approach, however, is that these assumptions are seldom satisfied in an unconstrained natural environment. Other common factors that can produce variations of image shading include the attenuation of light with distance, inter-reflections among different surfaces, variations in surface reflectance, cast shadows, specular highlights, and transparency. Existing computational models of 3D shape from shading cannot cope with any of these phenomena, yet human observers seem to have little difficulty in correctly identifying them.

Shape from Surface Contours and Texture

It has long been recognized that a convincing pictorial representation of an object can sometimes be achieved by drawing just a few critical lines, and there have been numerous attempts in both human and machine vision to analyze how line drawings of 3D

scenes might be perceptually interpreted. The earliest models to address this issue were developed for interpreting line drawings of simple plane-faced polyhedra whose vertices are all formed by the junction of three faces. The different types of vertices that can arise in line drawings of these objects were exhaustively catalogued, and then used to label which lines in a drawing correspond to convex, concave, or occluding edges. Similar procedures were later developed to label line drawings of arbitrary polyhedral scenes (Malik, 1987), and to deal with other types of lines corresponding to shadows or cracks.

Another type of image feature that provides useful information for the perception of 3D shape is the contour that separates the visible and occluded parts of a smoothly curved surface. Indeed, an occlusion contour presented in isolation can often provide sufficient information to recognize an object, and to reliably segment it into distinct parts. Koenderink (1984) has shown that the relative sign of curvature at each point on a smooth occlusion contour uniquely specifies the sign of Gaussian curvature of visible surface regions in its immediate local neighborhood. More recent research has combined this analysis with earlier work on polyhedral objects to perform edge labeling on complex surfaces with both smoothly curved and faceted regions (Malik, 1987).

Still another important source of information for the perception of 3D shape comes from patterns of reflectance on smoothly curved surfaces, which are often referred to as texture. Some popular textures that are used in optical art for creating an appearance of 3D shape include random patterns of polka dots or networks of roughly parallel contours (see Todd and Oomes, 2002). Several potential models have been proposed for the computational analysis of these patterns. Some of these models are based on the assumption that all texture elements are approximately circular. Others assume that the depicted surface is singly curved, and that the contours lie along lines of curvature or surface geodesics (e.g., Knill, 2001). The empirical evidence suggests, however, that these assumptions are too restrictive to account for the perceptions of human observers (Todd and Oomes, 2002). A more general approach to this problem has recently been developed by Malik and Rosenholtz (1997). This approach computes local surface structure by measuring the affine distortions between texture patches in neighboring image regions, based on a more ecologically reliable assumption about texture homogeneity.

Shape from Binocular Disparity

Some of the most powerful analyses for estimating 3D shape from visual information are designed to exploit the systematic transformations of optical structure that occur when an object is viewed over multiple vantage points. For example, human observers have two eyes with overlapping visual fields, such that each eye receives a slightly different view of the same scene. It is especially interesting to note that binocular overlap reduces the size of the combined visual field relative to what would otherwise be possible if the two eyes faced in opposite directions, as is the case with many other animals. For the ecology of human observers, however, this cost is apparently outweighed by the useful information that is provided by the disparities between each eye's view in the region of overlap.

In order to compute 3D shape from binocular disparity, it is first necessary to determine the correspondence relations between the patterns of stimulation in each eye. The difficulty of this problem is demonstrated most clearly by the ability of observers to perceive 3D structure from random dot stereograms, in which each stereoscopic half-image contains a dense configuration of small dots. For any given dot presented to one eye, the visual system must somehow determine a single corresponding dot with which it should be matched among the many possible targets presented to the other.

Numerous computational models have been developed for solving this stereo correspondence problem, many of which are designed to simulate the physiological properties of the primate visual system (see Anderson and Julesz, 1995, for a recent review).

Like other sources of visual information, the horizontal disparity between each eye's view is inherently ambiguous with respect to the metric structure of an observed scene. The ambiguity in this case arises from the fact that the disparity produced by a given depth interval varies with viewing distance. In principle, there are a variety of ways that disparity could be scaled based on other sources of information, such as knowledge of the convergence angle or an analysis of vertical disparities, but there is considerable evidence to indicate that human observers are incapable of doing so with any reasonable degree of accuracy. Although observers can make accurate judgments about some aspects of 3D shape from stereoscopic vision, their judgments of metric structure typically exhibit large systematic distortions, even when viewing real 3D scenes in a fully illuminated natural environment (see Hecht, van Doorn, and Koenderink, 1999).

A fundamental assumption for most existing models of binocular stereopsis is that the corresponding features in each eye's view are projectively related to the same physical points in 3D space. This assumption is generally valid for certain types of image structures, such as those that arise from discontinuities of surface orientation or surface reflectance, but there other types of optical phenomena for which this assumption can be violated. There are two important cases that need to be considered in this regard. One is the occurrence of features that are occluded in one eye but not the other, which are sometimes referred to as half-occlusions. These would be treated as noise by most existing models of binocular stereopsis, but there is a growing body of evidence to indicate that they provide an important source of information for human perception (Anderson and Julesz, 1995). A second important case to consider is the occurrence of optical structures such as smooth occlusion contours or specular highlights, whose location on a surface varies with viewing direction. Because each eye sees these structures at a different surface location, their binocular disparities should therefore provide misleading information about 3D shape. Recent evidence suggests, however, that highlights and smooth occlusions provide perceptually useful information that enhances the appearance of stereoscopic depth (Todd et al., 1997).

Shape from Motion

Another relevant source of information for the perceptual analysis of 3D shape includes the systematic transformations of optical structure that occur when an object is observed in motion. The analysis of structure from motion is similar in some respects to the analysis of shape from binocular disparity in that it generally requires two distinct stages of processing. The first of these stages is to compute an optical flow field from changing patterns of light on the retina, and numerous models have been proposed to describe how this process is accomplished within the primate visual system (see Watanabe, 1998, chaps. 4–6).

The next stage of the problem is to incorporate these motion measures to estimate the 3D structure of an observed scene. When an object rotates in depth under appropriate conditions, its pattern of projected motion over three or more views provides sufficient information to determine its complete metric structure. However, there is a growing body of evidence that human observers have low sensitivity to this information and that the perception of 3D structure from motion is primarily determined by first-order relations between pairs of views (Watanabe, 1998, chap. 12). First-order motion measures under weak perspective are similar to shading in that they allow an infinity of possible 3D interpretations that are all related by an affine transformation (Koenderink and van

Doorn, 1991). Thus, they can uniquely specify the affine structure of an object, while being inherently ambiguous with respect to metric structure. A similar distinction between these structural attributes is also characteristic of human perception. Observers are often quite accurate at judging structural properties that are uniquely specified by first-order motion measures, whereas judgments of metric structure are inaccurate and unreliable.

Another similarity between models for computing 3D shape from motion or binocular disparity is their dependence on a limited subset of the possible optical structures that can occur under natural conditions. Most shape-from-motion algorithms are based on a strong assumption that moving features within a visual image remain projectively attached to fixed locations on an object's surface, but a wide variety of common optical phenomena violate this assumption. These phenomena include the deformations of smooth occlusion contours and specular highlights, which do not remain fixed on an object's surface when it is observed in motion. Similarly, when an object moves relative to its sources of illumination, then the optical deformations of shadows and lambertian shading will violate this assumption as well. Although these aspects of optical motion are degenerate for most current models, they are easily interpretable for human observers (Norman and Todd, 1994).

Shape from Multiple Sources

All of the models considered thus far are designed to be used with a particular source of information presented in isolation. Under natural viewing conditions, of course, it is likely that these different sources of information would occur in combination with one another, thus providing a certain degree of redundancy for the perceptual specification of 3D shape. The presence of these redundancies could be potentially quite useful for resolving ambiguities that are often inherent in visual information. If two sources of information allow different families of possible interpretations, then their simultaneous occurrence could be used to mutually constrain those interpretations in order to obtain a more accurate estimate of 3D metric structure. There is some disagreement in the literature about the extent to which this strategy can be exploited in human perception. A majority of investigators have found, however, that observers' judgments of 3D metric structure are inaccurate and unreliable even when multiple sources of information are available simultaneously (e.g., Tittle et al., 1995).

Discussion

A fundamental problem for the computational analysis of 3D shape from various aspects of optical stimulation is that most known sources of visual information are inherently ambiguous with respect to the precise metric structure of an observed scene. There are two general strategies for dealing with problems. One is to incorporate prior assumptions about environmental constraints to limit the set of possible interpretations. The primary weakness of this approach is that it will produce large systematic errors when objects are observed in an unconstrained natural environment, in which these assumptions may frequently be violated. An alternative strategy that seems to be more characteristic of human perception is to limit the analysis of shape to those structural properties that can be estimated with a higher degree of confidence. Depending on the particular source of information being analyzed, this could involve a perceptual representation of affine, ordinal, or topological relations. The evidence suggests that biological visual systems prefer robustness over precision.

Another important problem for the computational analysis of 3D shape from visual information is that changes in image intensity can be caused by a wide variety of environmental phenomena, including surface occlusions, specular highlights, transparency, var-

iations in surface reflectance, variations in the pattern of illumination, and smooth or abrupt changes in surface orientation. Because all current models of 3D shape perception are appropriate for just a limited subset of these phenomena, their successful application in an unconstrained environment would seem to require an early-level process for labeling image intensity changes. It is interesting to note that contour labeling is one of the oldest problems in computational vision, but the successes in this area have been largely limited to line drawings (see Malik, 1987). In order to develop more robust models of 3D shape estimation, it is important that this work be extended to include natural scenes with shading and texture.

Road Map: Vision

Related Reading: Object Structure, Visual Processing; Stereo Correspondence; Tensor Voting and Visual Segmentation

References

- Anderson, B. L., and Julesz, B., 1995, A theoretical analysis of illusory contour formation in stereopsis, *Psychol. Rev.*, 102:705–743.
- Belhumeur, P. N., Kriegman, D. J., and Yuille, A. L., 1999, The bas-relief ambiguity, *Int. J. Comput. Vision*, 35:33–44.
- Hecht, H., van Doorn, A., and Koenderink, J. J., 1999, Compression of visual space in natural scenes and in their photographic counterparts, *Percept. Psychophys.*, 61:1269–1286.
- Horn, B. K. P., and Brooks, M. J., 1989, *Shape from Shading*, Cambridge, MA: MIT Press.
- Knill, D. C., 2001, Contour into texture: Information content of surface contours and texture flow, *J. Opt. Soc. Am. A*, 18:12–35.
- Koenderink, J. J., 1984, What does the occluding contour tell us about solid shape? *Perception*, 13:321–330.
- Koenderink, J. J., and van Doorn, A. J., 1991, Affine structure from motion, *J. Opt. Soc. Am. A*, 8:377–385.
- Koenderink, J. J., van Doorn, A. J., Kappers, A. M. L., and Todd, J. T., 2001, Ambiguity and the “mental eye” in pictorial relief, *Perception*, 30:431–448.
- Malik, J., 1987, Interpreting line drawings of curved objects, *Int. J. Comput. Vision*, 1:73–103.
- Malik, J., and Rosenholtz, R., 1997, Computing local surface orientation and shape from texture for curved surfaces, *Int. J. Comput. Vision*, 23:149–168.
- Norman, J. F., and Todd, J. T., 1994, The perception of rigid motion in depth from the optical deformations of shadows and occlusion boundaries, *J. Exp. Psychol. Hum. Percept. Perform.*, 20:343–356.
- Tittle, J. S., Todd, J. T., Perotti, V. J., and Norman, J. F., 1995, The systematic distortion of perceived 3D structure from motion and binocular stereopsis, *J. Exp. Psychol. Hum. Percept. Perform.*, 21:663–678.
- Todd, J. T., Norman, J. F., Koenderink, J. J., and Kappers, A. M. L., 1997, Effects of texture, illumination and surface reflectance on stereoscopic shape perception, *Perception*, 26:806–822.
- Todd, J. T., and Oomes, A. H. J., 2002, Generic and nongeneric conditions for the perception of surface shape from texture, *Vision Res.*, 42:837–850.
- Watanabe, T., 1998, *High-Level Motion Processing: Computational, Neurobiological, and Psychophysical Perspectives*, Cambridge, MA: MIT Press.

Perceptrons, Adalines, and Backpropagation

Bernard Widrow and Michael A. Lehr

Introduction

The field of neural networks has enjoyed major advances since 1960, a year which saw the introduction of two of the earliest feedforward neural network algorithms: the perceptron rule (Rosenblatt, 1962) and the LMS algorithm (Widrow and Hoff, 1960). Around 1961, Widrow and his students devised Madaline Rule I (MRI), the earliest learning rule for feedforward networks with multiple adaptive elements. The major extension of the feedforward neural network beyond Madaline I took place in 1971, when Paul Werbos developed a backpropagation algorithm for training multilayer neural networks. He first published his findings in 1974 in his doctoral dissertation (see BACKPROPAGATION: GENERAL PRINCIPLES). Werbos's work remained almost unknown in the scientific community until 1986, when Rumelhart, Hinton, and Williams (1986) rediscovered the technique and, within a clear framework, succeeded in making the method widely known.

The development of backpropagation has made it possible to attack problems requiring neural networks with high degrees of nonlinearity and precision (Widrow and Lehr, 1990; Widrow, Rumelhart, and Lehr, 1994). Backpropagation networks with fewer than 150 neural elements have been successfully applied to vehicular control simulations, speech generation, and undersea mine detection. Small networks have also been used successfully in airport explosive detection, expert systems, and scores of other applications. Furthermore, efforts to develop parallel neural network hardware are advancing rapidly, and these systems are now becoming available for attacking more difficult problems such as continuous speech recognition.

The networks used to solve the above applications varied widely in size and topology. A basic component of nearly all neural networks, however, is the adaptive linear combiner.

The Adaptive Linear Combiner

The adaptive linear combiner has as output a linear combination of its inputs. In a digital implementation, this element receives at time k an input signal vector or input pattern vector $\mathbf{X}_k = [x_0, x_{1k}, x_{2k}, \dots, x_{n_k}]^T$ and a desired response d_k , a special input used to effect learning. The components of the input vector are weighted by a set of coefficients, the weight vector $\mathbf{W}_k = [w_{0k}, w_{1k}, w_{2k}, \dots, w_{n_k}]^T$. The sum of the weighted inputs is then computed, producing a linear output, the inner product $s_k = \mathbf{X}_k^T \mathbf{W}_k$. The components of \mathbf{X}_k may be either continuous analog values or binary values. The weights are essentially continuously variable and can take on negative as well as positive values.

During the training process, input patterns and corresponding desired responses are presented to the linear combiner. An adaptation algorithm automatically adjusts the weights so the output responses to the input patterns will be as close as possible to their respective desired responses. In signal processing applications, the most popular method for adapting the weights is the simple LMS (least mean square) algorithm (Widrow and Hoff, 1960), often called the Widrow-Hoff Delta Rule (Rumelhart et al., 1986). This algorithm minimizes the sum of squares of the linear errors over the training set. The linear error ε_k is defined to be the difference between the desired response d_k and the linear output s_k during presentation k . Having this error signal is necessary for adapting the weights. Both the LMS rule and Rosenblatt's perceptron rule will be detailed in later sections.

An important element used in many neural networks is the ADaptive LInear NEuron, or *adaline* (Widrow and Hoff, 1960). In the neural network literature, such elements are often referred to as *adaptive neurons*. The adaline is an adaptive threshold logic

element. It consists of an adaptive linear combiner cascaded with a hard-limiting quantizer that is used to produce a binary ± 1 output, $y_k = \text{sgn}(s_k)$. A bias weight, *threshold*, w_{0k} , which is connected to a constant input, $x_0 = +1$, effectively controls the threshold level of the quantizer. Such an element may be seen as a McCulloch-Pitts neuron augmented with a learning rule for adjusting its weights.

In single-element neural networks, the weights are often trained to classify binary patterns using binary desired responses. Once training is complete, the responses of the trained element can be tested by applying various input patterns. If the adaline responds correctly with high probability to input patterns that were not included in the training set, it is said that generalization has taken place. Learning and generalization are among the most useful attributes of adalines and neural networks.

With n binary inputs and one binary output, a single adaline is capable of implementing certain logic functions. There are 2^n possible input patterns. A general logic implementation would be capable of classifying each pattern as either $+1$ or -1 , in accordance with the desired response. Thus, there are 2^{2^n} possible logic functions connecting n inputs to a single binary output. A single adaline is capable of realizing only a small subset of these functions, known as the linearly separable logic functions or threshold logic functions. These are the set of logic functions that can be obtained with all possible weight variations. With two inputs, a single adaline can realize 14 of the 16 possible binary logic functions. The two it cannot learn are exclusive OR and exclusive NOR functions. With many inputs, however, only a small fraction of all possible logic functions are realizable, i.e., linearly separable. Combinations of elements or networks of elements can be used to realize functions that are not linearly separable.

Nonlinear Neural Networks

One of the earliest trainable layered neural networks with multiple adaptive elements was the *Madaline I* structure of Widrow and Hoff. In the early 1960s, a 1,000-weight Madaline I was built out of hardware and used in pattern recognition research (Widrow and Lehr, 1990). The weights in this machine were memistors—

electrically variable resistors, developed by Widrow and Hoff, that are adjusted by electroplating a resistive link in a sealed cell containing copper sulfate and sulfuric acid.

Madaline I was configured in the following way. Retinal inputs were connected to a layer of adaptive adaline elements, the outputs of which were connected to a fixed logic device that generated the system output. Methods for adapting such systems were developed at that time. An example of this kind of network is shown in Figure 1. Two adalines are connected to an AND logic device to provide an output. With weights suitably chosen, the separating boundary in pattern space for the system can implement any of the 16 two-input binary logic functions, including the exclusive OR and exclusive NOR functions.

Madalines were constructed with many more inputs, with many more adaline elements in the first layer, and with various fixed logic devices such as AND, OR, and majority vote-taker elements in the second layer. Those three functions are all threshold logic functions.

Multilayer Networks

The madaline networks of the 1960s had an adaptive first layer and a fixed threshold function in the second (output) layer (Widrow and Lehr, 1990). The feedforward neural networks of today often have many layers, all of which are usually adaptive. The backpropagation networks of Rumelhart et al. (1986) are perhaps the best-known examples of multilayer networks. A three-layer feedforward adaptive network is illustrated in Figure 2. It is “fully connected” in the sense that each adaline receives inputs from every output in the preceding layer.

During training, the responses of the output elements in the network are compared with a corresponding set of desired responses. Error signals associated with the elements of the output layer are thus readily computed, so adaptation of the output layer is straightforward. The fundamental difficulty associated with adapting a layered network lies in obtaining *error signals* for hidden-layer adalines, that is, for adalines in layers other than the output layer. The backpropagation algorithm provides a method for establishing these error signals.

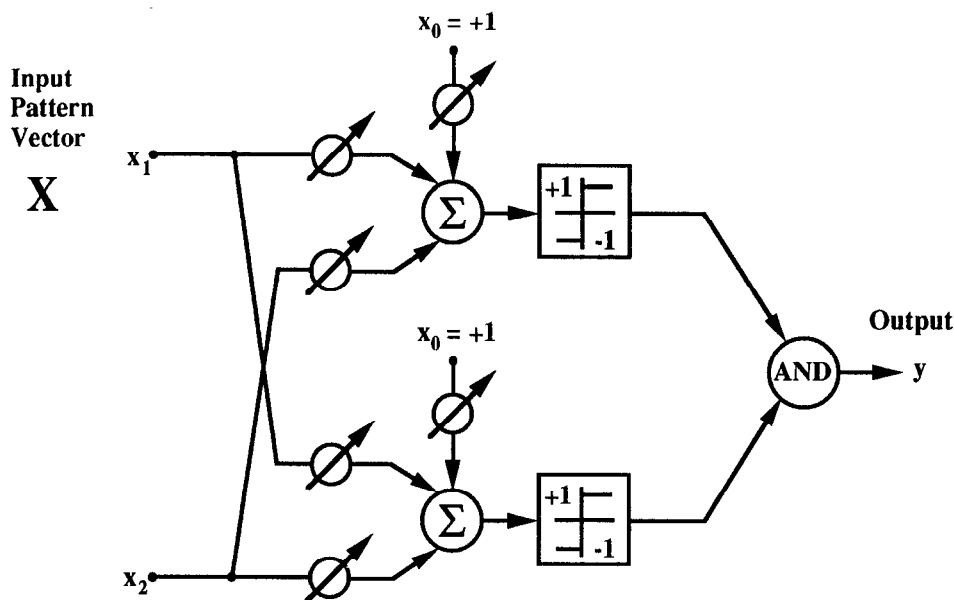
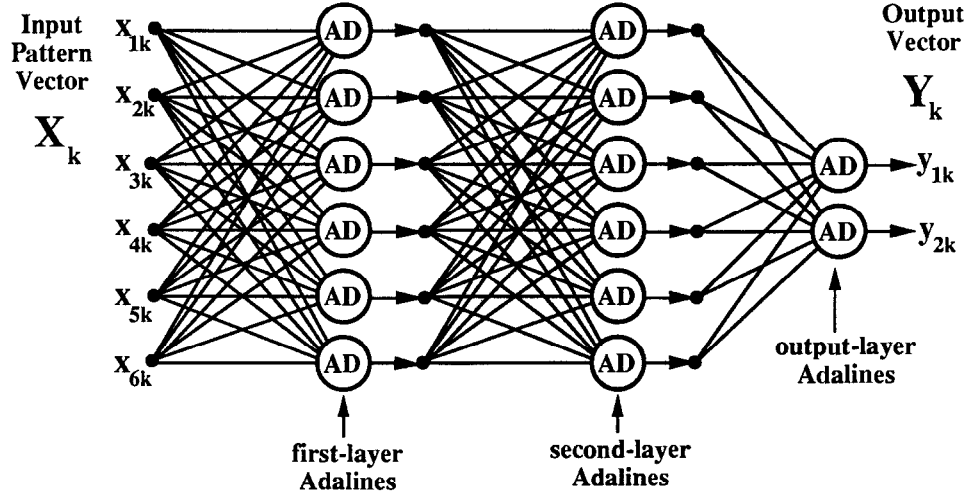


Figure 1. A two-adaline form of madaline.

Figure 2. A three-layer adaptive neural network.



Learning Algorithms

The iterative algorithms described here are all designed in accord with the *Principle of Minimal Disturbance: Adapt to reduce the output error for the current training pattern, with minimal disturbance to responses already learned*. Unless this principle is practiced, it is difficult to simultaneously store the required pattern responses. The minimal disturbance principle is intuitive. It was the motivating idea that led to the discovery of the LMS algorithm and the madaline rules. In fact, the LMS algorithm had existed for several months as an error reduction rule before it was discovered that the algorithm uses an instantaneous gradient to follow the path of steepest descent and minimizes the mean square error of the training set. It was then given the name LMS (least mean square) algorithm.

The LMS Algorithm

The objective of adaptation for a feedforward neural network is usually to reduce the error between the desired response and the network's actual response. The most common error function is the mean square error (MSE), averaged over the training set. The most popular approaches to mean-square-error reduction in both single-element and multielement networks are based on the method of gradient descent.

Adaptation of a network by gradient descent starts with an arbitrary initial value \mathbf{W}_0 for the system's weight vector. The gradient of the mean-square-error function is measured and the weight vector is altered in the direction opposite to the measured gradient. This procedure is repeated, causing the MSE to be successively reduced on average and causing the weight vector to approach a locally optimal value.

The method of gradient descent can be described by the relation

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(-\nabla_k) \quad (1)$$

where μ is a parameter that controls stability and rate of convergence and ∇_k is the value of the gradient at a point on the MSE surface corresponding to $\mathbf{W} = \mathbf{W}_k$.

The LMS algorithm works by performing approximate steepest descent on the mean-square-error surface in weight space. This surface is a quadratic function of the weights and is therefore convex and has a unique (global) minimum. An instantaneous gradient based on the square of the instantaneous error is

$$\hat{\nabla}_k = \frac{\partial \varepsilon_k^2}{\partial \mathbf{W}_k} = \begin{Bmatrix} \frac{\partial \varepsilon_k^2}{\partial w_{0k}} \\ \vdots \\ \frac{\partial \varepsilon_k^2}{\partial w_{nk}} \end{Bmatrix} \quad (2)$$

LMS works by using this crude gradient estimate in place of the true gradient ∇_k . Making this replacement into Equation 1 yields

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(-\hat{\nabla}_k) = \mathbf{W}_k - \mu \frac{\partial \varepsilon_k^2}{\partial \mathbf{W}_k} \quad (3)$$

The instantaneous gradient is used because (1) it is an unbiased estimate of the true gradient (Widrow and Stearns, 1985) and (2) it is easily computed from single data samples. The true gradient is generally difficult to obtain. Computing it would involve averaging the instantaneous gradients associated with all patterns in the training set. This is usually impractical and almost always inefficient.

The present error or *linear error* ε_k is defined to be the difference between the desired response d_k and the linear output $s_k = \mathbf{W}_k^T \mathbf{X}_k$ before adaptation:

$$\varepsilon_k \triangleq d_k - \mathbf{W}_k^T \mathbf{X}_k \quad (4)$$

Performing the differentiation in Equation 3 and replacing the linear error by the definition in Equation 4 gives

$$\mathbf{W}_{k+1} = \mathbf{W}_k - 2\mu \varepsilon_k \frac{\partial (d_k - \mathbf{W}_k^T \mathbf{X}_k)}{\partial \mathbf{W}_k} \quad (5)$$

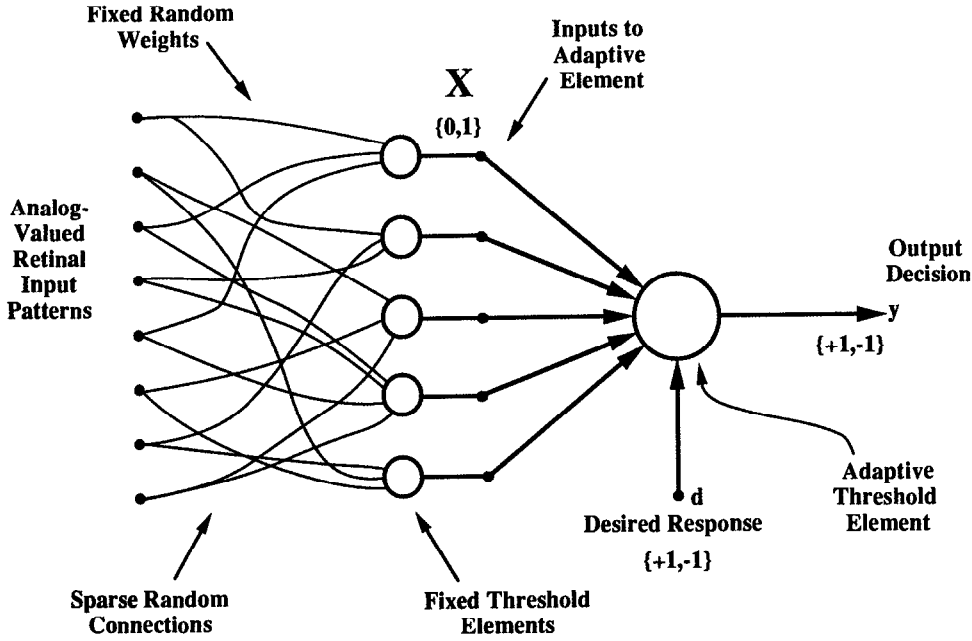
Noting that d_k and \mathbf{X}_k are independent of \mathbf{W}_k yields

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu \varepsilon_k \mathbf{X}_k \quad (6)$$

This is the LMS algorithm. The learning constant μ determines stability and convergence rate (Widrow and Stearns, 1985).

The Perceptron Learning Rule

The Rosenblatt α -perceptron (Rosenblatt, 1962), diagrammed in Figure 3, processed input patterns with a first layer of sparse, randomly connected, fixed-logic devices. The outputs of the fixed first layer fed a second layer, which consisted of a single adaptive linear threshold element. Other than the convention that its input signals and its output signal were $\{1,0\}$ binary and that no bias weight was included, this element was equivalent to the adaline element. The learning rule for the α -perceptron was very similar to LMS, but its behavior was in fact quite different.

Figure 3. Rosenblatt's α -perceptron.

Adapting with the perceptron rule makes use of the *quantizer* error $\tilde{\epsilon}_k$, defined to be the difference between the desired response and the output of the quantizer

$$\tilde{\epsilon}_k \triangleq d_k - y_k \quad (7)$$

The perceptron rule, sometimes called the *perceptron convergence procedure*, does not adapt the weights if the output decision y_k is correct, i.e., if $\tilde{\epsilon}_k = 0$. If the output decision disagrees with the binary desired response d_k , however, adaptation is effected by adding the input vector to the weight vector when the error $\tilde{\epsilon}_k$ is positive, or subtracting the input vector from the weight vector when the error $\tilde{\epsilon}_k$ is negative. Note that the quantizer error $\tilde{\epsilon}_k$ is always equal to either 1, -1, or 0. Thus, the product of the input vector and the quantizer error $\tilde{\epsilon}_k$ is added to the weight vector. The perceptron rule is identical to the LMS algorithm, except that with the perceptron rule, one-half of the quantizer error, $\tilde{\epsilon}_k/4$, is used in place of the linear error ϵ_k of the LMS rule. The perceptron rule is nonlinear, in contrast to the LMS rule, which is linear. Nonetheless, it can be written in a form which is very similar to the LMS rule of Equation 6:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu \frac{\tilde{\epsilon}_k}{2} \mathbf{X}_k \quad (8)$$

Rosenblatt normally set μ to 1. In contrast to LMS, the choice of k does not affect the stability of the perceptron algorithm, and it affects convergence time only if the initial weight vector is non-zero. Also, while LMS can be used with either analog or binary desired responses, Rosenblatt's rule can be used only with binary desired responses.

The perceptron rule stops adapting when the training patterns are correctly separated. There is no restraining force controlling the magnitude of the weights, however. The direction of the weight vector, not its magnitude, determines the decision function. The perceptron rule has been proved capable of separating any linearly separable set of training patterns (Rosenblatt, 1962; Nilsson, 1965). If the training patterns are not linearly separable, the perceptron algorithm goes on forever, and in most cases the weight vector gravitates toward zero. As a result, on problems that are not linearly

separable, the perceptron often does not yield a low-error solution, even if one exists.

This behavior is very different from that of the LMS algorithm. Continued use of LMS does not lead to an unreasonable weight solution if the pattern set is not linearly separable. Nor, however, is this algorithm guaranteed to separate any linearly separable pattern set. LMS typically comes close to achieving such separation, but its objective is different, i.e., error reduction at the linear output of the adaptive element.

"Backpropagation" for the Sigmoid Adaline

A *sigmoid adaline* element incorporates a sigmoidal nonlinearity. The input-output relation of the sigmoid can be denoted by $y_k = \text{sgm}(s_k)$. A typical sigmoid function is the hyperbolic tangent

$$y_k = \tanh(s_k) = \left(\frac{1 - e^{-2s_k}}{1 + e^{-2s_k}} \right) \quad (9)$$

We shall adapt this adaline with the objective of minimizing the mean square of the *sigmoid error* $\tilde{\epsilon}_k$, defined as

$$\tilde{\epsilon}_k \triangleq d_k - y_k = d_k - \text{sgm}(s_k) \quad (10)$$

The method of gradient descent is used to adapt the weight vector. By following the same line of reasoning used to develop LMS, the instantaneous gradient estimate obtained during presentation of the k th input vector \mathbf{X}_k can be found to be

$$\hat{\nabla}_k = \frac{\partial(\tilde{\epsilon}_k)^2}{\partial \mathbf{W}_k} = 2\tilde{\epsilon}_k \frac{\partial \tilde{\epsilon}_k}{\partial \mathbf{W}_k} = -2\tilde{\epsilon}_k \text{sgm}'(s_k) \mathbf{X}_k \quad (11)$$

Using this gradient estimate with the method of gradient descent provides a means for minimizing the mean square error even after the summed signal s_k goes through the nonlinear sigmoid. The algorithm is

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(-\hat{\nabla}_k) = \mathbf{W}_k + 2\mu\tilde{\epsilon}_k \mathbf{X}_k \quad (12)$$

where δ_k denotes $\tilde{\epsilon}_k \text{sgm}'(s_k)$. The algorithm of Equation 12 is the *backpropagation* algorithm for the single adaline element, although

the backpropagation name makes sense only when the algorithm is utilized in a layered network, which will be studied later.

If the sigmoid is chosen to be the hyperbolic tangent function (Equation 9), then the derivative $\text{sgm}'(s_k)$ is given by

$$\begin{aligned}\text{sgm}'(s_k) &= \frac{\partial(\tanh(s_k))}{\partial s_k} \\ &= 1 - (\tanh(s_k))^2 = 1 - y_k^2\end{aligned}\quad (13)$$

Accordingly, Equation 12 becomes

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu\tilde{\epsilon}_k(1 - y_k^2)\mathbf{X}_k \quad (14)$$

The single sigmoid adaline trained by backpropagation shares some advantages with both the adaline trained by LMS and the perceptron trained by Rosenblatt's perceptron rule. If a pattern set is linearly separable, the objective function of the sigmoid adaline, the mean square error, is minimized only when the pattern set is separated. This is because, as the weights of the sigmoid adaline grow large, its response approximates that of a perceptron with weights in the same direction. The sigmoid adaline trained by backpropagation, however, also shares the advantage of the adaline trained by LMS: it tends to give reasonable results even if the training set is not separable.

Backpropagation training of the sigmoid adaline does have one drawback, however. Unlike the linear error of the adaline, the output error of the sigmoid adaline is a nonlinear function of the weights. As a result, its mean square error surface is not quadratic, and may have local minima in addition to the optimal solution. Thus, unlike the perceptron rule, it cannot be guaranteed that backpropagation training of the sigmoid adaline will successfully separate a linearly separable training set. Nonetheless, the single sigmoid adaline performs admirably in many filtering and pattern classification applications. Its most important role, however, occurs in multilayer networks, to which we now turn.

Backpropagation for Networks

The backpropagation technique is a substantial generalization of the single sigmoid adaline case discussed in the previous section. When applied to multilayer feedforward networks, the backpropagation technique adjusts the weights in the direction opposite to the instantaneous gradient of the sum square error in weight space. Derivations of the algorithm are widely available in the literature (Rumelhart et al., 1986; Widrow and Lehr, 1990). Here we provide only a brief summary of the result.

The instantaneous sum square error ϵ_k^2 is the sum of the squares of the errors at each of the N_y outputs of the network. Thus

$$\epsilon_k^2 = \sum_{i=1}^{N_y} \epsilon_{ik}^2 \quad (15)$$

In its simplest form, backpropagation training begins by presenting an input pattern vector \mathbf{X} to the network, sweeping forward through the system to generate an output response vector \mathbf{Y} , and computing the errors at each output. We continue by sweeping the effects of the errors backward through the network to associate a *square error derivative* δ with each adaline, computing a gradient from each δ , and finally updating the weights of each adaline based on the corresponding gradient. A new pattern is then presented and the process is repeated. The initial weight values are normally set to small random values. The algorithm will not work properly with multilayer networks if the initial weights are either zero or poorly chosen non-zero values.

The δ s in the output layer are computed just as they are for the sigmoid adaline element. For a given output adaline,

$$\delta = \tilde{\epsilon} \text{sgm}'(s) \quad (16)$$

where $\tilde{\epsilon}$ is the error at the output of the adaline and s is the summing junction output of the same unit.

Hidden-layer calculations, however, are more complicated. The procedure for finding the value of $\delta^{(l)}$ the value of δ associated with a given adaline in hidden layer l , involves respectively multiplying each derivative $\delta^{(l+1)}$ associated with each element in the layer immediately downstream from the given adaline by the weight connecting it to the given adaline. These weighted square error derivatives are then added together, producing an error term $\epsilon^{(l)}$, which in turn is multiplied by $\text{sgm}'(s^{(l)})$, the derivative of the given adaline's sigmoid function at its current operating point. Thus, the δ corresponding to adaline j in hidden layer l is given by

$$\delta_j^{(l)} = \text{sgm}'(s_j^{(l)}) \sum_{i \in N^{(l+1)}} \delta_i^{(l+1)} w_{ij}^{(l+1)} \quad (17)$$

where $N^{(l+1)}$ is a set containing the indices of all adalines in layer $l+1$ and $w_{ij}^{(l+1)}$ is the weight connecting adaline i in layer $l+1$ to the output of adaline j in layer l .

Updating the weights of the adaline element using the method of gradient descent with the instantaneous gradient is a process represented by

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(-\hat{\nabla}_k) = \mathbf{W}_k + 2\mu\delta_k\mathbf{X}_k \quad (18)$$

where \mathbf{W} is the adaline's weight vector and \mathbf{X} is the vector of inputs to the adaline. Thus, after backpropagating all square error derivatives, we complete a backpropagation iteration by adding to each weight vector the corresponding input vector scaled by the associated square error derivative. Equations 16, 17, and 18 comprise the general weight update rule of the back propagation algorithm for layered neural networks.

Many useful techniques based on the backpropagation algorithm have been developed. One popular method, called *backpropagation through time*, allows dynamical recurrent networks to be trained. Essentially, this is accomplished by running the recurrent neural network for several time steps and then "unrolling" the network in time. This results in a virtual network with a number of layers equal to the product of the original number of layers and the number of time steps. The ordinary backpropagation algorithm is then applied to this virtual network and the result is used to update the weights of the original network. This approach was used by Nguyen and Widrow (1989) to enable a neural network to learn without a teacher how to back up a computer-simulated trailer truck to a loading dock (Figure 4). This is a complicated and highly nonlinear steering task. Nevertheless, with just six inputs providing information about the current position of the truck, a two-layer neural network with only 26 sigmoid adalines was able to learn of its own accord to solve this problem. Once trained, the network could successfully back up the truck from any initial position and orientation in front of the loading dock.

Discussion

Although this article has focused on pattern classification issues, nonlinear neural networks are equally useful for such tasks as interpolation, system modeling, state estimation, adaptive filtering, and nonlinear control. Unlike their linear counterparts, which have a long track record of success, nonlinear neural networks have only recently begun proving themselves in commercial applications. The capabilities of multielement neural networks have improved markedly since the introduction of Madaline Rule I. This has resulted largely from development of the backpropagation algorithm, easily the most useful and popular neural network training algorithm currently available. As we have seen, backpropagation is a generalization of LMS that allows complex networks of sigmoid adalines to be efficiently adapted. Backpropagation and related algorithms

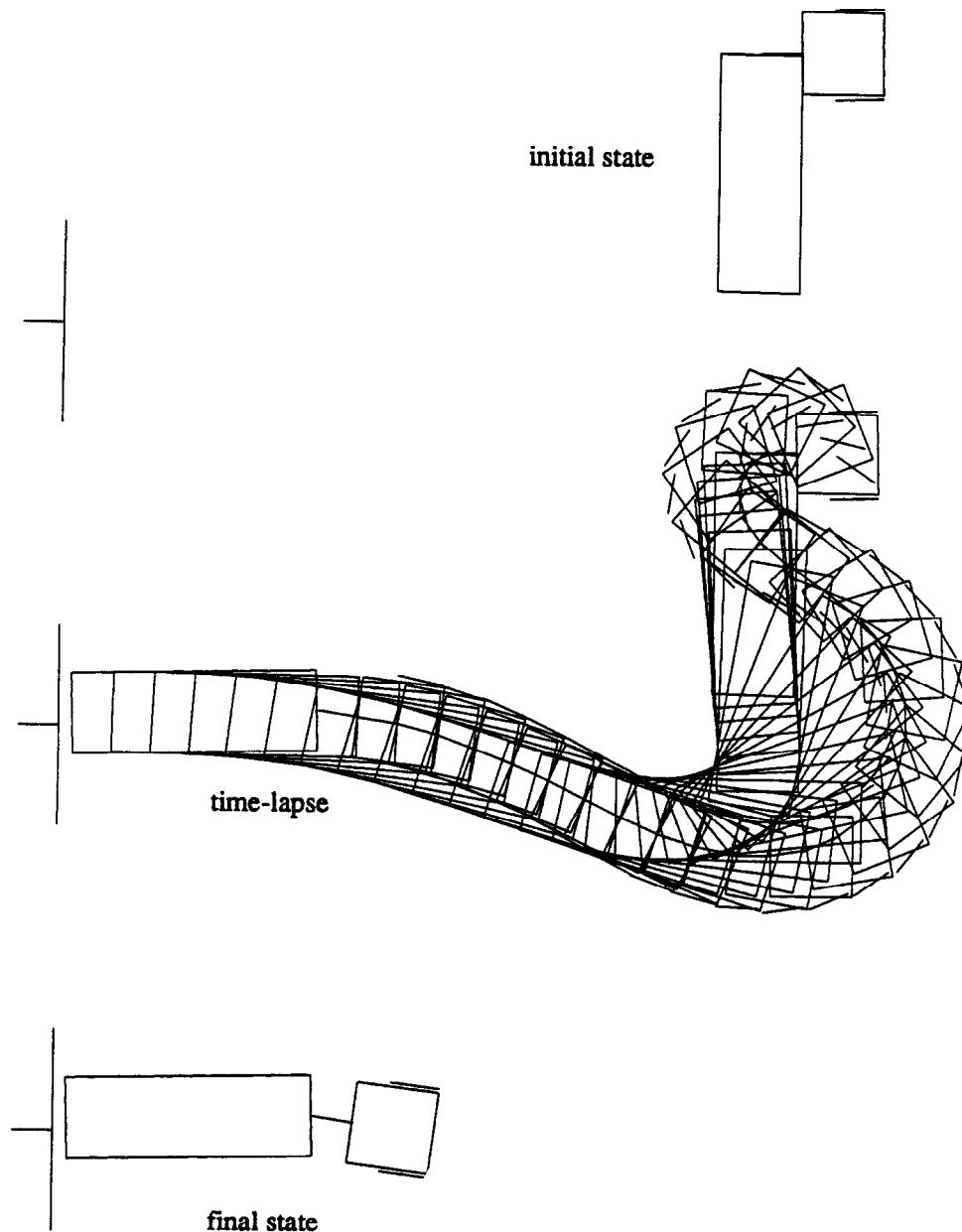


Figure 4. Example of a truck backup sequence.

are in large part responsible for the dramatic growth the field of neural networks is currently experiencing.

The timing of the current boom in the field of neural networks is also due to the rapid advance in computer and microprocessor performance, which continues to improve the feasibility and cost-effectiveness of computationally expensive techniques in relation to classical approaches of engineering and statistics. Although single-element linear adaptive filters are still used more extensively than nonlinear multielement neural networks, the latter are potentially applicable to a much wider range of problems. Furthermore, the applications for which multielement neural networks are best suited often involve complicated nonlinear relationships for which classical solutions are either ineffective or unavailable. The continued advancement of neural network algorithms and techniques, in conjunction with improvements in the special and general purpose computer hardware used to implement them, sets the stage for a

future in which neural networks will play an increasing role in commercial and industrial applications.

[Reprinted from the First Edition]

Road Maps: Grounding Models of Networks; Learning in Artificial Networks

Background: Dynamics and Bifurcation in Neural Nets

Related Reading: Identification and Control; Filtering, Adaptive

References

- Nilsson, N., 1965, *Learning Machines*, New York: McGraw-Hill. ♦
 Nguyen, D., and Widrow, B., 1989, The truck backer-upper: An example of self-learning in neural networks, in *Proceedings of the International*

- Joint Conference on Neural Networks*, vol. 2, New York: IEEE, pp. 357–363.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, Learning internal representations by error propagation, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, *Foundations*, (D. E. Rumelhart, J. L. McClelland, and PDP Research Group, Eds.), Cambridge, MA: MIT Press, chap. 8. ♦
- Rosenblatt, F., 1962, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Washington, DC: Spartan.
- Widrow, B., and Hoff, M. E., Jr., 1960, Adaptive switching circuits, in *1960 IRE WESCON Convention Record*, Part 4, New York: IRE, pp. 96–104.
- Widrow, B., and Lehr, M. A., 1990, 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation, *Proc. IEEE*, 78:1415–1442. ♦
- Widrow, B., Rumelhart, D., and Lehr, M. A., 1994, Neural networks: Applications in industry, business, and science, *Commun. ACM*, 37(3):93–105.
- Widrow, B., and Stearns, S. D., 1985, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall. ♦

Perspective on Neuron Model Complexity

Wilfrid Rall

Introduction

There is a wide range of choice in model complexity, from very simple to rather complex neuron models. Which model to choose depends, in each case, on the context. How much information do we already have about the neurons under consideration? What questions do we wish to explore?

Sometimes we wish to model a particular biological neuron whose anatomy and physiology are known in considerable experimental detail. In such cases, we may choose to specify a model that includes at least some of the dendritic branching of the neuron, because synapses from one source may be distributed preferentially to either a distal or a proximal dendritic location, while synapses from another source may end mainly at the soma, or on a different dendritic tree of the same neuron. Also, there may be a functionally significant nonuniformity in the distribution of channel densities of several ion channel types over the surface of the soma and dendrites. How much detail is needed depends on the biological experiments to be simulated and the questions asked.

Conversely, many network modelers are not constrained by anatomical or physiological data. For some network modeling, this is partly justified by a paucity of available data. However, more often, network modelers are constrained by their mathematical methods, which lead them to focus on abstract networks composed of extremely simple units. The simplest units are two-state, binary units, analogous to atomic spin, previously studied for condensed matter physics (see, e.g., OPTIMIZATION, NEURAL). Such binary units do not resemble neurons, but they do have a strong appeal for nerve-net modelers, who have generated an extensive literature. That literature lies outside the scope of the current article.

When simple binary units are compared with a dendritic neuron model (especially with nonuniform distributions of synapses and ion channels), it becomes apparent that one dendritic model neuron can perform tasks that would require a network of many simple units to duplicate. For the purpose of machine design, it may seem quite appropriate to consider the trade-offs in cost and flexibility (between the one realistic model and the many binary units), but for functional insights and understanding of biological nervous systems, I freely state my bias for the more realistic neuron models. I do not choose the most complex, in the sense of including all known anatomical and physiological details; I favor an intermediate level of complexity that preserves the most significant distinctions between regions (soma, proximal dendritic, distal dendritic, different trees), especially when further justified by nonuniform distributions of synapses and ion channels (see also Segev, 1992).

The claim is sometimes made that network properties depend primarily on the connectivity between the units, and not on the

properties of the units. Although this may be true for some gross network properties, I do not believe it to be true for many of the actual biological networks that perform important, complicated tasks. I regard it as a worthwhile challenge for like-minded neural modelers to provide interesting demonstrations in support of this belief. The challenge is to demonstrate a useful computation or discrimination that can be accomplished with a dendritic neuron model, or a network composed of such models, and then show that this useful capacity is lost when all of the dendritic membrane is lumped with the soma, and all of the inputs to each neuron are now delivered to that lumped membrane. There are valuable examples that already meet this challenge, several of which are presented in three later sections of this article. Other examples can be found in a review by Borst and Egelhaaf (1994; see also VISUAL COURSE CONTROL IN FLIES).

Brief Historical Notes

Neurons are biological cells, and their electrical properties depend on ions and the cell membrane, in a manner brilliantly elucidated by Hodgkin, Huxley, and Katz during the period 1948–1952. It is a fascinating historical coincidence that two seeds of their important insights can be found in a single 1902 volume of *Pfluegers Archiv*, in pioneering articles by Bernstein and by Overton. Following the earlier theoretical insights of Nernst and Planck, Bernstein recognized the importance of the potassium ion concentration difference across the membrane in determining a non-zero resting potential; he regarded excitation as a brief breakdown of the membrane, a concept that prevailed until 1948, when Hodgkin and Katz showed that the key is a sudden overwhelming increase in membrane permeability to sodium ions. Overton's 1902 paper had correctly emphasized the importance of the external sodium ion concentration to the excitability properties of nerve, but no one put these ideas together in 1902. Between 1900 and 1914, several investigators, including Hermann, Lucas, and Lapique, recognized the importance of membrane capacitance; the concept of nerve membrane as a leaky integrator, with a threshold for an action potential, was used to understand the strength-duration curve for a threshold stimulus. During the 1930s, several investigators, including Rashevsky, Hill, and Monnier, developed mathematical models of excitation and inhibition; Rashevsky's textbook *Mathematical Biophysics* (1948) includes many examples of network modeling by himself; by Householder, Landahl, and others; and by McCulloch and Pitts, whose famous 1943 paper arose in the context of Rashevsky's research seminars at the University of Chicago (see also the historical notes in Schwartz, 1990). Ever since that time, many neuron modelers have been content with the leaky integrator neuron model, which reduces a neuron to a single node that

integrates synaptic excitation (+) and synaptic inhibition (−) delivered to it by other neurons. Several errors caused by these oversimplified assumptions were demonstrated by compartmental computations in 1962; see Rall's chapter in Reiss (1964) or in Segev, Rinzel, and Shepherd (1995). Other chapters in Reiss (1964) also provide several interesting early perspectives on neural modeling. The mathematical modeling of nonlinear membrane properties has been presented and discussed in an outstanding early review by FitzHugh (1969), and in a chapter by Rinzel and Ermentrout that appears in Koch and Segev (1989).

The concept of a nerve axon as an extended core conductor (i.e., membrane cylinder with ionic conducting media inside and outside) goes back to the 1870s, when it was treated mathematically by Hermann and Weber; both the concept of passive electrotonus in membrane cylinders and the mathematics (of passive cable theory) were explored over the years, culminating in classic papers by Hodgkin and Rushton and by Davis and Lorente de Nó, both around 1946–1947; see references in Rall (1977). Before 1900, neuroanatomical studies by Ramón y Cajal demonstrated the extensiveness of dendritic branching for most neuron types; this was confirmed by many anatomists, and later (in the 1950s), use of the electron microscope made it possible to verify the existence of very many synapses on the dendritic branches and on the dendritic spines of neurons. These anatomical facts, together with the introduction of intracellular microelectrode recording from single dendritic neurons (in the 1950s), made it urgent to extend cable theory to the dendrites of individual neurons. This was begun in the late 1950s and carried forward into the 1960s and 1970s; for a review, see Jack, Noble, and Tsien (1975) or Rall (1977); see also Koch and Segev (1989), McKenna, Davis, and Zornetzer (1992), Rall et al. (1992), Segev et al. (1995), and DENDRITIC PROCESSING.

Dendritic Neuron Model Complexity: Geometric Versus Membrane Complexity

The concept of complexity in dendritic neuron models can be explored quite efficiently by making a two-dimensional chart. One dimension would be membrane complexity, ranging from the simple case of a passive linear membrane to that of postsynaptic membrane models with time-varying ion permeability (or conductance), and then to excitable membrane models with voltage-dependent ion conductances as described by Hodgkin and Huxley (see AXONAL MODELING), or as now described with increasing detail in terms of many different species of ion channels whose voltage and time dependence are currently being characterized (see ION CHANNELS: KEYS TO NEURONAL SPECIALIZATION). The other dimension would be geometric complexity, ranging from the simple case of an isopotential region of membrane (a soma, or a space-clamped section of a cylinder) to that of a uniform membrane cylinder with two sealed ends (or with one end voltage clamped, or current clamped), and then to several dendritic trees attached to a soma (with or without an axon), where the soma may be shunted and the branching of the trees may be specified to varying degrees of arbitrariness. The most complicated geometric case, with arbitrary branching and shunted soma, has recently been solved analytically (for transients, assuming passive membrane) in a mathematical tour de force by Major, Evans, and Jack (1993); see also Holmes, Segev, and Rall (1992). The less complicated, but illuminating, case of idealized branching with a point soma was solved earlier by Rall and Rinzel; see the 1973 and 1974 papers reprinted in Segev et al. (1995). However, these analytical methods do depend on the assumption of linear membrane properties. When nonlinear membrane complexity is combined with geometric complexity, the transient solutions can be obtained computationally by using compartmental models; see 1964 and 1968 papers reprinted in Se-

gev et al. (1995); see also DENDRITIC PROCESSING and several chapters in Koch and Segev (1989) and in McKenna et al. (1992).

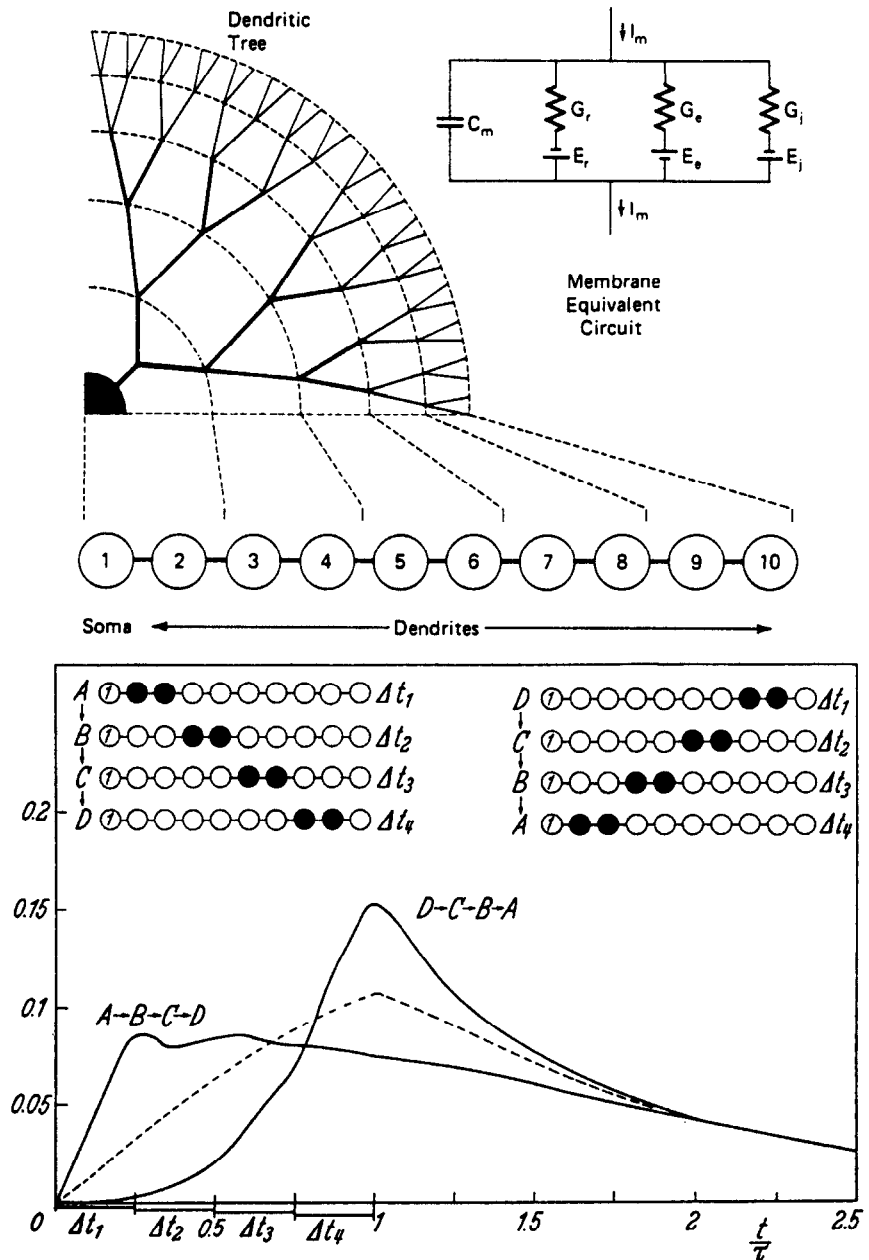
Dendritic Model Can Provide Spatiotemporal Discrimination

Figure 1 summarizes a demonstration of how a dendritic neuron model could perform a discrimination between two contrasting spatiotemporal patterns of synaptic input (i.e., possible movement detection); this discrimination is lost if the compartments and inputs are lumped together. A neuron is represented by a chain of ten compartments; compartment 1 represents the soma, while compartments 2 to 10 represent dendritic membrane of the same neuron, with increasing cable distance from the soma. One spatiotemporal input sequence, A-B-C-D, has the proximal dendritic input first, followed in time by progressively more distal dendritic input locations. The other input pattern, D-C-B-A, is opposite in having the most distal input first, followed in time by progressively more proximal input locations. Comparison of the resulting computed voltage transients (EPSP at the soma), shown in Figure 1, reveals that input sequence D-C-B-A yields a significantly larger voltage amplitude than does input sequence A-B-C-D. Intuitive understanding of this computed result is obtained by noting that the delayed proximal input builds on membrane depolarization that has spread to the soma (with delay) from the distal dendrites (which were activated earlier). If the voltage threshold for spiking at the soma were tuned between these two peak amplitudes, a spike would be produced by sequence D-C-B-A, but a spike would not be produced by sequence A-B-C-D; this would constitute a discrimination between these two sequences. The dashed curve in the figure shows the computed result when the compartments are lumped together; either sequence of input synapse activation then produces the same intermediate result, and no discrimination would be possible.

Models for Mitral and Granule Cell Populations in Olfactory Bulb

A rather different example is provided by the neuron models used for the mitral cell and granule cell populations in simulating experiments on the OLFACTORY BULB (q.v.) of rabbit; see the 1968 paper of Rall and Shepherd in Segev et al. (1995); or see figures 2.11 and 2.12 in Koch and Segev (1989). Here, the task was to model and compute extracellular field potentials that matched those observed experimentally in olfactory bulb when the mitral cell population was activated in near synchrony by means of an antidromic volley. Compartmental models were used; a nine-compartment model (three axonal, one somatic, and five dendritic) was used to simulate antidromic activation of a mitral cell, while a ten-compartment model was used to simulate nonspiking activity in the dendrites of an axonless granule cell. The dendritic compartments were absolutely essential for the computation of electric current flow between different dendritic regions of each granule cell and between the dendrites and soma of each mitral cell; without these currents, it would have been impossible to compute the field potentials generated by the synchronously activated neuron populations. Also, this modeling led to a critically important distinction in the depth distribution of the two fields: the larger, longer-lasting field potentials generated by the very large population of granule cells extended to significantly greater depth in the olfactory bulb than did the earlier, smaller, briefer field potentials generated by the mitral cell population. The difference between these two fields was such that neither population could have generated the other field. This provided the key to our prediction of (and the functional interpretation of subsequent electron microscopic evidence for) dendrodendritic synaptic interactions between the mitral secondary dendrites and the distal dendrites of the granule cells, which are

Figure 1. Effect of spatiotemporal dendritic pattern of synaptic input on the computed EPSP at the soma, for a ten-compartment model. Upper diagram indicates the mapping of a soma and dendritic tree into a chain of ten equal compartments. Compartment 1 represents the soma membrane, while compartments 2 to 10 represent dendritic membrane, from proximal to distal locations. The middle diagram (at left) shows the synaptic input sequence A-B-C-D, meaning proximal dendritic input location active first, followed by successive activation at increasingly more distal input locations; this input pattern produced the soma voltage transient (computed composite EPSP) labeled A-B-C-D at lower left. The middle diagram (at right) shows the opposite synaptic input sequence. D-C-B-A, meaning distal dendritic input location first, followed by successively more proximal input locations; this input pattern produced a significantly different soma voltage transient (computed composite EPSP), having a delayed rise to a larger peak amplitude, labeled D-C-B-A. In both cases, each input compartment (shown in black) received a synaptic excitatory conductance pulse ($G_e = G_r$, for a duration 0.25τ) during one of the four labeled periods. The same total amount of synaptic input produced the dashed curve when the spatiotemporal pattern was eliminated by smearing the synaptic conductance in space and time ($G_e = 0.25 G_r$ in eight compartments (compartments 2 to 9) for the full time duration from $t = 0$ to $t = \tau$). The membrane equivalent circuit (upper right) holds for each compartment. Further details can be found in the 1964 chapter by Rall in Reiss (1964), reprinted in Segev et al. (1995).



intermingled in the external plexiform layer of the bulb. If these cells had been modeled as lumped somas, without dendrites, neither the successful simulation of the experimental field potentials nor the exciting new insights about a dendrodendritic pathway for recurrent inhibition would have been possible.

Similarly, for the earlier simulations and insights obtained for motor neurons of cat spinal cord, we found that observations made at the soma seemed paradoxical until they were understood in terms of synaptic events that occur in distal dendrites (see the 1967 paper in Segev et al., 1995); these results and insights would not have been possible without dendritic compartments in the neuron field.

Comment on Functional Aspect of Dendrodendritic Interactions

To highlight an important functional difference, note first that motor neurons do exhibit the classical functional polarity envisaged

by Ramón y Cajal and Sherrington (as well as most modelers). The dendrites receive inputs from many sources (their effects converge on the soma); the output (when spike threshold is exceeded) is an all-or-nothing action potential propagated by the axon to muscle units that may be quite distant; i.e., classically, input is received by the dendrites and output is delivered by the axon. In contrast, the dendrites of both the mitral cells and the granule cells are functionally different, because they both send as well as receive synaptic information, locally. The mitral secondary dendrites, which are smooth and spineless, send nonsynaptic excitatory output, which is received as input by the spines (see DENDRITIC SPINES) of the adjacent granule cells. The granule cells have no axons and perhaps no action potentials; their spines receive graded synaptic excitatory input and then send graded synaptic output that is inhibitory to the adjacent mitral cell dendrites. It is important to emphasize that this is not a rare anomaly found only in the olfactory

bulb; evidence for dendrodendritic synapses and for graded local synaptic interactions is now found in many parts of the brain (e.g., retina and inferior olive). In 1965, when we (Rall et al.; see 1966 and 1968 papers reprinted in Segev et al., 1995) first presented our interpretations of dendrites that send as well as receive, some critics resisted this concept as heretical; however, our functional interpretation of these dendrodendritic synapses is now widely accepted by physiologists and anatomists. This kind of graded two-way synaptic interaction is very different from the classical functional polarity just described for motor neurons; it provides graded functional coupling between neurons (without axonal impulses). The implications have so far hardly been explored in theoretical networks. Such exploration will require explicit modeling of dendritic compartments; a point neuron model would be useless for this. Note also that computational exploration of localized plastic changes at synapses and at dendritic spines depends on neuron models that include dendritic compartments.

Network Rhythmogenesis Using the Traub Model and a Reduced Model

A 19-compartment cable model for the pyramidal cells of the CA3 region of guinea pig hippocampus was developed by Traub et al. (1991; see also the chapter by Traub and Miles in McKenna et al., 1992). Based on experimental measurements, parameters were chosen for each compartment, using up to six active ionic conductances, and controlled by ten-channel gating variables. They succeeded in finding a set of physiologically reasonable parameters for which the network of model neurons could simulate several important aspects of the experimental repertoire of the slightly disinhibited hippocampal slice preparation. Although Traub et al. recognized that their successful simulations of network behavior depended on specifying significantly different ion channel densities for the soma and for the dendrites, the critical importance of this difference was made starkly clear by the modeling of Pinsky and Rinzel (1994); they obtained essentially the same behavioral repertoire by using a network composed of a severely reduced neuron model consisting of only two compartments per pyramidal cell. One compartment represented the soma and proximal dendrites, while the other compartment represented the distal dendrites. To be more specific, the ion channels for fast-spiking currents (inward sodium, and delayed rectifier) were restricted to the soma-like compartment, and the ionic channels for the slower calcium currents (calcium-inward and calcium-modulated currents) were restricted to the dendrite-like compartment. I hasten to add that these results also show that at least two compartments are needed for simulations of this behavior; a single lumped compartment, with all of the ion channels in parallel, could not produce the same behavior, especially the rhythm, which basically involves an alternating flow of current between the two coupled compartments. A special advantage of the reduced neuron model is that much simpler computations can explore how much the interesting behavior depends on the values of key parameters, especially the parameter that defines the tightness of coupling between the two compartments. Also, the behavior of very large networks can be explored more efficiently using such a reduced neuron model. Further study may show that the two-compartment model cannot match the fuller model in certain important tests, but, in any case, these findings so far represent a very satisfying example that illustrates the thesis of this article.

Discussion

In an earlier essay offering perspective on neural modeling (a chapter in Binder and Mendell, 1990), I provided a completely different set of examples. One of these provided a detailed consideration of the number of degrees of freedom to be found in a neuron model

composed of a thousand compartments. Such models exist today because of tremendous improvements in anatomical methods and in computation facilities now available to experimental investigators. Because they have the morphological data and a computer, why not put everything into the model? The answer is that you can if you wish to, but you should be aware of the huge number of degrees of freedom implied by the large number of parameters that must be specified; as someone once pointed out, given enough free parameters, he could fit an elephant. Is the membrane uniform, or do we know the density of every channel species in every membrane compartment? How are the inputs distributed to the many compartments? Today, the data needed for such detailed specifications are largely missing; however, such data are beginning to become at least partly available for some neurons. Where the data are not available, the modeler must make reasonable guesses. If it seems reasonable to assign the same parameter values to many neighboring compartments, one should consider lumping those compartments together to produce a simpler model with fewer compartments. Nevertheless, one important merit of the larger model is that it can be used to test whether it can perform some interesting task that cannot be performed by the reduced model.

As stated earlier, my preference is for intermediate levels of complexity; I vote for the smallest number of compartments that can preserve what one judges to be the functionally important differences between dendritic regions with regard to ion channel densities and to distributions of synapses from different sources. If a five-compartment model can provide a good approximation of the interesting properties of a 1,000-compartment model, I would prefer the smaller model, for two important reasons: (1) it helps sharpen our intuitive understanding about what is essential to obtaining the behavior of interest, and (2) it can greatly facilitate computations with networks composed of such neuron models. I expect modeling of this kind will continue to be particularly fruitful in the near future (see also the discussion by Segev, 1992).

Concluding Comment

As when drawing, painting, sculpting, or composing music, so too, when deeply engaged in neural modeling, I believe that much of the fun and satisfaction comes from interactions between my conscious mind and my subconscious sources of creativity. It seems that preliminary sketching serves to plant seeds in the subconscious, where they can grow, if nurtured. Conscious pursuit of the problem can then stimulate differentiation and development in the subconscious and may produce fruits that can reach conscious awareness (popping up like mushrooms produced by an underground mycelium). Such fruits may provide exciting new insights for the conscious mind. Indeed, the pleasure of such creative discovery can become almost addictive for those fortunate enough to have both the interest and the opportunity for creative activity. I hasten to add that a lot of hard work is usually required to test and polish before one can produce a finished product. Pioneering in dendritic neuron modeling provided me with such an opportunity; now [at the time of the First Edition], with retirement upon me, I hope to persist by sculpting, painting, and by designing a house for a natural mountain setting.

[Reprinted from the First Edition]

Road Maps: Biological Neurons and Synapses; Grounding Models of Neurons

Background: I.1. Introducing the Neuron

Related Reading: Dendritic Processing

References

Binder, M. D., and Mendell, L. M., Eds., 1990, *The Segmental Motor System*, Oxford: Oxford University Press.

- Borst, A., and Egelhaaf, M., 1994, Dendritic processing of synaptic information by sensory interneurons, *Trends Neurosci.*, 17:257–263.
- FitzHugh, R., 1969, Mathematical models of excitation and propagation in nerve, in *Biological Engineering* (H. P. Schwann, Ed.), New York: McGraw-Hill. ♦
- Holmes, W. R., Segev, I., and Rall, W., 1992, Interpretation of time constant and electrotonic length estimates in multi-cylinder or branched neuronal structures, *J. Neurophysiol.*, 68:1401–1420.
- Jack, J. J. B., Noble, D., and Tsien, R. W., 1975, *Electric Current Flow in Excitable Cells*, Oxford: Oxford University Press. ♦
- Koch, C., and Segev, I., 1989, *Methods in Neuronal Modeling: From Synapses to Networks*, Cambridge, MA: MIT Press. ♦
- Major, G., Evans, J. D., and Jack, J. J. B., 1993, Solutions for transients in arbitrarily branching cables: I. Voltage recording with a somatic shunt, *Biophys. J.*, 65:423–449.
- McKenna, T., Davis, J., and Zornetzer, S. F., 1992, *Single Neuron Computation*, San Diego, CA: Academic Press. ♦
- Pinsky, P. F., and Rinzel, J., 1994, Intrinsic and network rhythmogenesis in a reduced Traub model for CA3 neurons, *J. Computat. Neurosci.*, 1:39–60.
- Rall, W., 1977, Core conductor theory and cable properties of neurons, in *Handbook of Physiology: The Nervous System: Cellular Biology of Neurons*, sect. 1, vol. I, part 1, chap. 3, Bethesda, MD: American Physiological Society, pp. 39–97. ♦
- Rall, W., Burke, R. E., Holmes, W. R., Jack, J. J. B., Redman, S. J., and Segev, I., 1992, Matching dendritic neuron models to experimental data, *Physiol. Rev.*, 72:S159–S186. ♦
- Rashevsky, N., 1948, *Mathematical Biophysics*, Chicago: University of Chicago Press; reissued 1960, New York: Dover.
- Reiss, R., Ed., 1964, *Neural Theory and Modeling*, Stanford, CA: Stanford University Press.
- Schwartz, E. L., 1990, *Computational Neuroscience*, Cambridge, MA: MIT Press. ♦
- Segev, I., 1992, Single neurone models: Oversimple, complex and reduced, *Trends Neurosci.*, 15:414–421. ♦
- Segev, I., Rinzel, J., and Shepherd, G. M., Eds., 1995, *The Theoretical Foundation of Dendritic Function: Selected Papers of Wilfrid Rall with Commentaries*, Cambridge, MA: MIT Press. ♦
- Traub, R., Wong, R., Miles, R., and Michelson, H., 1991, A model of a CA3 hippocampal pyramidal neuron incorporating voltage-clamp data on intrinsic conductances, *J. Neurophysiol.*, 66:635–649.

Phase-Plane Analysis of Neural Nets

Bard Ermentrout

Introduction

Models of neural networks often involve the solutions to differential equations that describe the time evolution of these complex systems. The dynamical behavior of these networks ranges from the convergence to an equilibrium (generally desired in connectionist applications) to oscillatory behavior (in models of central pattern generators and bursting) through possibly chaotic behavior. There are many ways to analyze these models; the most commonly used techniques entail simulation. In this article I will give an overview of an alternative technique for studying the *qualitative* behavior of small systems of interacting neural networks. One form that the models take is (Ellias and Grossberg, 1975; Hopfield, 1984; Wilson and Cowan, 1972):

$$\tau_i \frac{dx_i}{dt} = -x_i + f_i \left(\sum_{j=1}^n w_{ij} x_j + s_i \right) \quad i = 1, \dots, n \quad (1)$$

where x_i represents the activity or firing rate of the i th neuron, τ_i is the time constant, w_{ij} are the connection weights, s_i are inputs, and f_i are typically saturating nonlinear functions that have the form shown in Figure 1. That is, the nonlinear functions are increasing and bounded. Some typical examples are:

$$f(x) = \tanh(x) \quad (2)$$

$$f(x) = \tan^{-1}(x) \quad (3)$$

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (4)$$

Often, a slightly different form of (1) is chosen where the nonlinearities are inside the sums. The transformation from one to the other is elementary and all of the following holds for either type of model.

A complete analysis of networks of the form in Equation 1 is obviously impossible. However, if $n \leq 2$, then a fairly complete description of Equation 1 can be given. Thus, the goal of this article is to introduce the reader to the qualitative theory of differential equations in the plane. In particular, I will analyze two neuron

networks that consist of (1) two excitatory cells, (2) two inhibitory cells, and (3) an excitatory and an inhibitory cell. The advantages of restricting the analysis to these small networks are the special topology of the plane, the completeness of the analysis possible, and finally the ease of exposition. Indeed, an overview of nonlinear dynamics can be obtained through these simple examples. Beer (1995) has attempted to exhaustively study the dynamics in the case $n = 2$ and gives a nearly complete overview of the possible types of behavior that can be expected. However, he does miss several interesting examples (Ermentrout, 1998, pp. 371–373). Another more general approach for the analysis of large numbers of coupled systems is to use bifurcation methods that enable one to *reduce* the dimensionality of the resulting equations and then apply techniques such as those used here. While planar systems may seem to be a rather extreme simplification, there is some justification for it. For example, in some local cortical circuits, there is no structure in the connectivity and there are essentially two types of neurons, excitatory and inhibitory. Thus, we can view the simple planar system as representing a population of coupled excitatory and inhibitory neurons. This approach was used successfully to study cortical processing in the rodent somatosensory system (Pinto et al., 1996) and to explain the effects of altering inhibitory interneurons in the hippocampus (Tsodyks et al., 1997).

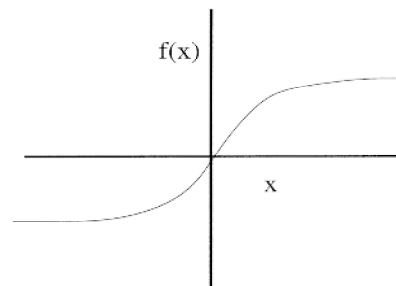


Figure 1. Typical nonlinear input-output function of a single model neuron.

The approach of this article is not restricted to neural networks and can be applied to a variety of other systems such as positive-feedback biochemical models (Segel, 1984), activator-inhibitor systems (Edelstein-Keshet, 1988), population and disease models (Murray, 1989), and membrane models of the action potential (Rinzel and Ermentrout, 1998). The techniques are powerful and provide insights into the behavior of these systems that would otherwise only be accessible through simulation. Computational methods are a very powerful adjunct to this type of analysis and, together with the qualitative analysis of this article, enable the researcher to understand his or her system.

In the next section, I will describe a pair of neurons coupled with mutual inhibition and mutual excitation. The penultimate section is devoted to a summary of the rich behavior of an excitatory-inhibitory pair. Finally, some comments on numerical methods and software close the review. In DYNAMICS AND BIFURCATION IN NEURAL NETS (q.v.), a systematic analysis of a particular case is given in order to illustrate alternative techniques.

Two Coupled Cells of the Same Type

In this section, we analyze the behavior of two cells that act via mutual inhibition or mutual excitation. I will use phase-plane analysis to draw a complete picture of the phase space.

General Considerations

Before analyzing the two-component neural network, I will first give a brief description of phase-plane techniques in general. Consider a planar differential equation:

$$x' = f(x, y) \quad (5)$$

$$y' = g(x, y) \quad (6)$$

At each point (x_0, y_0) there is a solution $(x(t), y(t))$ such that $(x(0), y(0)) = (x_0, y_0)$ and such that the tangent to the trajectory is $(f(x(t), y(t)), g(x(t), y(t)))$. Thus, at any point in the plane, one can easily determine the direction of the trajectory by simply evaluating f and g at that point. These directions enable one to paint a qualitative picture of the dynamics of Equations 5 and 6; i.e., I can determine where x and y are increasing and decreasing with time. The most crucial points are those values of x and y at which the direction of the trajectory changes. Thus, setting $f(x, y) = 0$ defines a curve in the plane where x does not change and breaks the plane into regions where x is either increasing or decreasing. This curve is called the *x-nullcline*. The curve $g(x, y) = 0$ defines the *y-nullcline*. The two curves together usually break the plane into regions of four distinct types: (1) x and y are increasing, (2) x and y are decreasing, (3) x increases and y decreases, and (4) x decreases and y increases. The intersection of the two nullclines occurs at points where both x and y are not changing, that is, at *equilibria* or *rest states* of Equations 5 and 6.

The behavior of trajectories away from equilibria is straightforward and is found by simply looking at the signs of f and g . Near the equilibria, one can look at the linearization of (f, g) about the equilibrium. This results in a two-by-two matrix called the *Jacobian*:

$$A = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} \equiv \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (7)$$

where the partial derivatives are evaluated at the equilibrium. The eigenvalues of A determine the behavior of the equilibria. If they both have negative real parts, the equilibrium is stable, and if any have positive real parts, the equilibrium is unstable. If both are real

and of the same sign, the point is called a *node*. Nodes consist of infinitely many trajectories emanating from (unstable) or entering (stable) the equilibrium. If both eigenvalues are complex, the rest state is a *vortex*, and trajectories spiral into (stable) or out of (unstable) the rest point. If the eigenvalues have opposite signs, the rest state is a *saddle point*. Then a single pair of trajectories that define the *stable manifold* or *set* enter the rest point, and a single pair of trajectories, defining the *unstable manifold*, leave the equilibrium. When the determinant of A (i.e., $ad - bc$) is negative, the rest point is a saddle; if it is positive and the trace of A ($a + d$) is non-zero, the equilibrium is a node or vortex. Cases for which the real part is zero do not persist for small changes in the parameters and often indicate the appearance of new equilibria or periodic solutions. A simple necessary and sufficient criterion for linear stability is that the trace $a + d$ be negative and the determinant of A , $ad - bc$, be positive. A complete description of phase-plane methods can be found in Edelstein-Keshet (1988) and in most texts on ordinary differential equations.

Crossed Excitatory and Inhibitory Networks

The first result I want to establish in systems that have mutual coupling of the same sign is that periodic solutions are impossible. Once this is established, then a complete characterization can be made simply by studying the intersections of the nullclines.

Theorem 1. Suppose that $w_{21}w_{12} \geq 0$. Then there are no periodic solutions to

$$\tau_1 x'_1 = -x_1 + f(w_{11}x_1 + w_{12}x_2 + s_1) \quad (8)$$

$$\tau_2 x'_2 = -x_2 + f(w_{21}x_1 + w_{22}x_2 + s_2) \quad (9)$$

As the proof of this theorem was given in the previous edition of the *Handbook* (p. 733, Theorem 1), I will not prove it again here.

All solutions to Equations 8 and 9 are bounded, and Theorem 1 implies that trajectories are monotone, so this means that all solutions tend to equilibria. This in turn means that the time constants can be set to 1 without loss of generality, as the dynamics is completely trivial. The intersections of the nullclines and some observations on the signs of the coefficients in the linearized matrix based on the nullclines allow one to completely determine the number and stability type of the equilibria.

Recall that f is increasing and bounded. Without loss of generality, one can assume that the minimum of f is 0 and the maximum is 1. f is invertible, and the inverse is also monotone, with asymptotes at 0 and 1. The formula for the x_1 -nullcline is

$$x_2 = (-w_{11}x_1 - s_1 + f^{-1}(x_1))/w_{12} \quad (10)$$

The x_2 -nullcline satisfies:

$$x_1 = (-w_{22}x_2 - s_2 + f^{-1}(x_2))/w_{21} \quad (11)$$

If $h(x) = (-w_s x - s + f^{-1}(x))/w_c$, then h is monotone if w_s is either positive or small and negative. However, if w_s is large enough, h develops a kink and is "cubic"-shaped (Figure 2A). If w_c is positive (mutual excitation) then $h \rightarrow -\infty$ as $x \rightarrow 0$ and $h \rightarrow \infty$ as $x \rightarrow 1$ (Figure 2A). When w_c is negative (mutual inhibition), the asymptotes are switched (Figure 2B). Finally, the stimulus parameter s merely translates the nullclines up and down in the case of the x_1 -nullcline and left-right for the x_2 -nullcline. The phase plane is easy to construct with these observations.

In both cases, there can be up to nine different equilibria, and there is always at least one. Figure 3 shows some typical configurations for mutually inhibitory interactions. To assess the stability of the equilibria, one need only look at the positions of the nullclines at the equilibria. Referring to Equation 7, I will use the nullclines to determine the signs and relative magnitudes of the entries

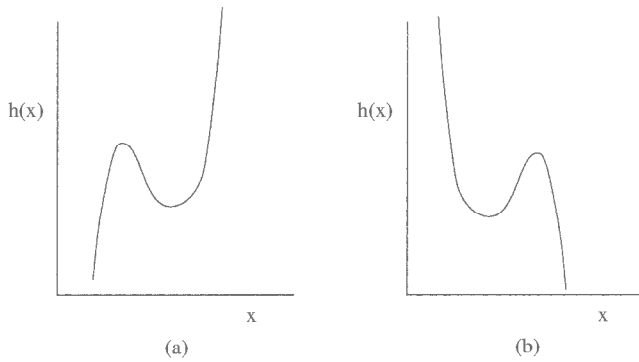


Figure 2. Nullcline shape for (A) mutual excitation and (B) mutual inhibition.

in this matrix. For mutually inhibiting cells, the following are necessary and sufficient for stability:

1. The slope of both nullclines is negative through an equilibrium.
2. The slope of the x_1 nullcline is steeper than the x_2 nullcline through the equilibrium.

If either of these is violated, the equilibrium is unstable.

For mutually excitatory cells, the conditions for stability are:

1. The slope of both nullclines is positive through an equilibrium.
2. The slope of the x_1 nullcline is steeper than the x_2 nullcline through the equilibrium.

Tangential intersections are saddle nodes and, as some parameter varies, will lead either to two new equilibria or to the disappearance of the pair. The matrix A has a zero eigenvalue when there are tangential crossings, for then the slopes of the nullclines are the same. That is, $-a/b = -c/d$, so $ad - bc = 0$. A bit of counting shows that when there are nine equilibria, four are stable nodes,

four are saddle points, and one is an unstable node. As the parameters vary, a pair of equilibria is lost, a saddle point and either a stable node or the unstable node, leaving seven equilibria. Further losses of equilibria (or gains, up to a maximum of nine) are obtained as the parameters vary, ending in the minimum of a single globally stable equilibrium.

When there are several stable equilibria, it is important to determine what initial conditions lead to which of the equilibria. The set of all initial data that tend to a particular equilibrium point is called the *domain of attraction* of the equilibrium point. For the present networks, this is very easy to determine geometrically. Figure 3 depicts a network of mutually inhibitory cells that has five equilibria (labeled a – e). The above discussion allows one to conclude that a , c , and e are stable nodes and b and d are saddle points. Each saddle point has associated with it one positive eigenvalue and one negative eigenvalue. Corresponding to this negative eigenvalue is the stable manifold for the saddle point. It consists of the set of all initial conditions that tend to the equilibrium point as $t \rightarrow \infty$. For two-dimensional neural nets, this is a one-dimensional set and is then often called a *separatrix*. I have drawn it for each of the two saddle points in Figure 3 as the dashed lines pointing into b and d . These curves divide the two-dimensional plane into three regions, which I have labeled A , C , and E . All initial data in A tend to equilibrium point a , and so on. Thus, although the saddle points are unstable, their stable manifolds provide the boundaries that determine the final states of the network given the initial state. From this description the reader should be able to construct complete qualitative pictures for other configurations of the nullclines for mutually excitatory or inhibitory nets.

Summarizing, I have used phase-plane analysis to show that for a pair of coupled neurons with mutual excitation or inhibition, the only stable solutions are equilibria. The stable manifolds of the saddle points divide the plane into domains of attraction for each of the stable equilibria. All equilibria are approached monotonically, and there can be up to four stable steady states.

A Pair of Excitatory and Inhibitory Cells

In many regions of cortex, and in fact throughout the central nervous system, many of the coupled excitatory and inhibitory cells

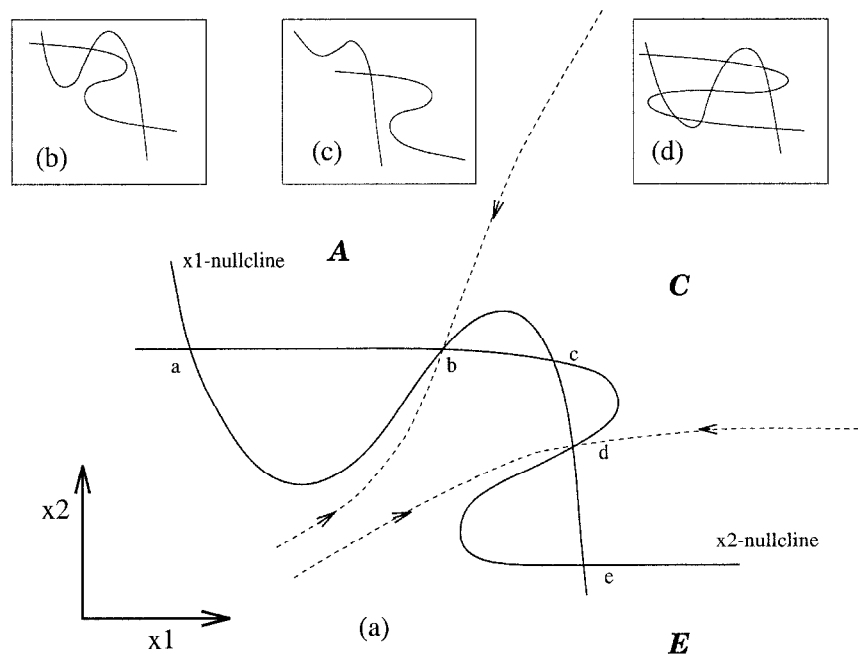


Figure 3. Phase plane for two mutually inhibitory neurons. Nullclines are solid lines and the stable manifolds of the saddle points b and d are shown dashed. a , c , and e are stable nodes with domains of attraction A , C , and E , respectively. Insets show other some other possible nullcline configurations.

constitute a local neural network. These networks have been the subject of numerous mathematical and computational investigations (Wilson and Cowan, 1972; Ellias and Grossberg, 1975; Ermentrout and Cowan, 1979; Beer, 1995; Pinto et al., 1996). One can view such systems either as two neurons acting in isolation (a difficult experiment to imagine) or, more reasonably, as a two-layer network with spatially homogeneous activity. Then each component is the activity of a *pool* of cells rather than the activity of a single cell.

I will consider a network of the form:

$$x_1' = -x_1 + f(w_{11}x_1 - w_{12}x_2 + s_1) \quad (12)$$

$$x_2' = (-x_2 + f(w_{21}x_1 - w_{22}x_2 + s_2))/\tau \quad (13)$$

where all of the weights are non-negative. I have introduced a time constant for the inhibitory neurons because one cannot expect them to have the same temporal behavior as the excitatory cells. The Jacobian matrix $A = [\partial x_i / \partial x_j]$ at an equilibrium point (\bar{x}_1, \bar{x}_2) has coefficients:

$$a = -1 + w_{11}f'(w_{11}\bar{x}_1 - w_{12}\bar{x}_2 + s_1) \quad (14)$$

$$b = -w_{12}f'(w_{11}\bar{x}_1 - w_{12}\bar{x}_2 + s_1) < 0 \quad (15)$$

$$c = (w_{21}f'(w_{21}\bar{x}_1 - w_{22}\bar{x}_2 + s_2))/\tau > 0 \quad (16)$$

$$d = (-1 - w_{22}f'(w_{21}\bar{x}_1 - w_{22}\bar{x}_2 + s_2))/\tau < 0 \quad (17)$$

It is clear that all of the coefficients except for a have a fixed sign independent of the parameters. If $w_{11}f' > 1$, then $a > 0$, and the system is called an *activator-inhibitor* system, since x_1 activates both itself and x_2 , while x_2 inhibits everything to which it connects. Activator-inhibitor models occur ubiquitously in biology, and their dynamics is rich and varied (see, e.g., PATTERN FORMATION, BIOLOGICAL). Oscillations, excitability, and multiple steady states are among the possible behaviors of these networks. Since a very complete analysis of these systems as applied to neural excitation is given in Rinzel and Ermentrout (1998), I only sketch some of the dynamic behavior possible for this network.

The qualitative behavior of any planar model can be understood by combining nullcline analysis with local stability analysis of the equilibria, which depends on the coefficients of the Jacobian A . The neural model studied in Rinzel and Ermentrout (1998) has exactly the same nullcline structure and has a Jacobian matrix with the same structure as the neural net model. Hence, I will only outline the dynamics of this system; details can be extracted from the aforementioned article.

It is instructive to first consider the effects of parameters on the shapes of the nullclines. A typical nullcline configuration is shown in Figure 4 for Equations 12 and 13. The x_2 -nullcline is always monotonically increasing; w_{21} sharpens it, while w_{22} makes it shallower and s_2 shifts it left and right. As described earlier, the effect of w_{11} is to kink the x_1 -nullcline, while w_{12} makes it less kinked; s_1 shifts it up and down. Finally, the parameter τ has no effect on the nullclines but dramatically alters the dynamics and stability of the equilibria. Changing τ has no effect on the determinant of A (so a saddle point cannot become a node), but it can switch the sign of the trace of A and so change a point from a stable node to an unstable node.

The positions of the nullclines make it clear that there can be up to five equilibria and at least one equilibrium point. Furthermore, all equilibria that occur on the "unkinked" part of the x_1 -nullcline are necessarily asymptotically stable, since then $a < 0$ in Equation 14. Thus, the trace, $a + d < 0$, and the determinant, $ad - bc > 0$. If w_{11} is sufficiently small so that the x_1 -nullcline is monotone, then there is only one equilibrium point, and it is globally stable. This statement follows from the facts that all solutions are bounded and from application of Bendixson's negative criterion (Edelstein-Keshet, 1988), which eliminates periodic orbits when $a + d < 0$. Any time the inhibitory nullcline has a lesser slope than the excit-

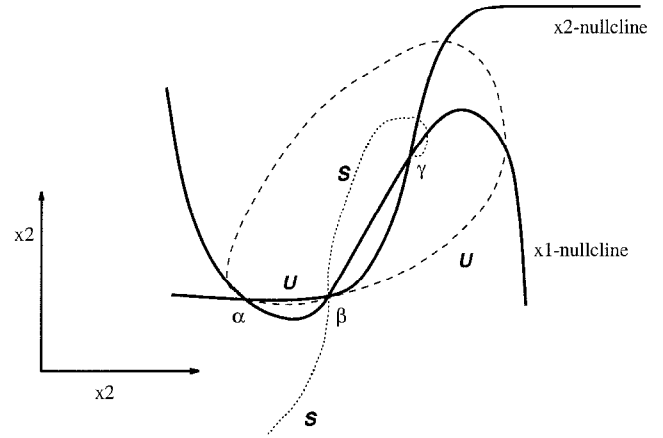


Figure 4. Typical phase plane for an excitatory-inhibitory pair. Nullclines and a typical trajectory are shown. There is a unique globally attracting equilibrium point.

atory nullcline, the equilibrium is a saddle point. These considerations, along with the preceding discussion, show how the parameters affect the local existence and stability of various rest states. The global dynamics is much more complicated since one cannot eliminate the possibility of limit cycle solutions.

Excitability

One important difference between networks consisting of one excitatory and one inhibitory layer and the networks described earlier in this article is the possibility of excitable dynamics. As was shown earlier, trajectories of the activity of cells are necessarily monotone. Thus, if, say, x_1 is increasing, then it can never decrease again. However, in mixed networks, no such restriction occurs, and it is possible for x_1 to initially increase before decreasing again. In particular, a network is said to be *excitable* if there is a *unique globally stable* rest state with the following property: Small perturbations from rest decay monotonically back to rest, but perturbations larger than some *threshold* continue to grow before decaying back to the stable rest state (Figure 5). There are at least two qualitatively different types of excitability for networks with the present structure. In type I excitability there are three equilibria, while in type II there is one. These two cases are described in Rinzel and Ermentrout (1998). In the context of neural networks, this type of behavior has been called an *active transient*. It can be viewed

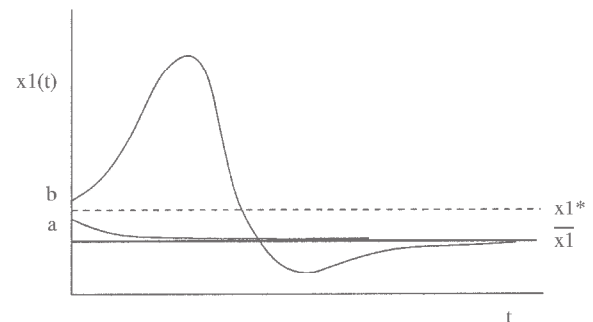


Figure 5. Excitable dynamics. \bar{x}_1 is the globally stable rest state and x_1^* is the threshold. Trajectory a is subthreshold and b is suprathreshold.

as a transient excitatory activity due to a stimulus that is eventually quelled by the inhibitory interneuronal feedback.

Periodic Solutions

Periodic solutions occur generally (although not strictly) when there is a single rest state on the middle branch and it is unstable. This point must necessarily be a node, and the boundedness of solutions thus implies that a limit cycle exists. If some parameter (say τ) is varied in such a way as to make the unique equilibrium go from a stable point to an unstable point (without introducing any new rest states), then a *Andronov-Hopf bifurcation* generically occurs, and this implies that a periodic solution exists near the rest state. For planar systems, easily checked necessary conditions for an Andronov-Hopf bifurcation are that the determinant of A remain positive and the trace change from negative to positive as the parameter is varied. If this new limit cycle is unstable, then there can be regimens in parameter space where there are two stable behaviors: (1) a stable rest state and (2) a stable *large-amplitude* periodic solution (Figure 6). This is known as *bistability*.

Other Behavior

In addition to excitability, multiple equilibria, oscillations, and bistability, other types of dynamic behavior can be found in these simple models. Infinite period oscillations and homoclinic trajectories can be obtained in some parameter regimens. (A *homoclinic* trajectory is one that leaves a saddle point from one side and enters it from another, and can occur as the period of a limit cycle tends to infinity.) Homoclinics are important because they separate qualitatively different types of behavior. Furthermore, when one periodically stimulates a system with homoclinics, it is possible to obtain a complex irregular behavior called *chaos* that cannot occur in planar systems without forcing (see Guckenheimer and Holmes, 1983.)

There are many other pictures possible with this simple model and I urge the reader to explore the phase-plane dynamics of this excitatory inhibitory net. Phase-plane methods provide a powerful analytic and qualitative tool for studying small neural networks. When combined with sophisticated numerical tools, a complete understanding of the global dynamics is possible.

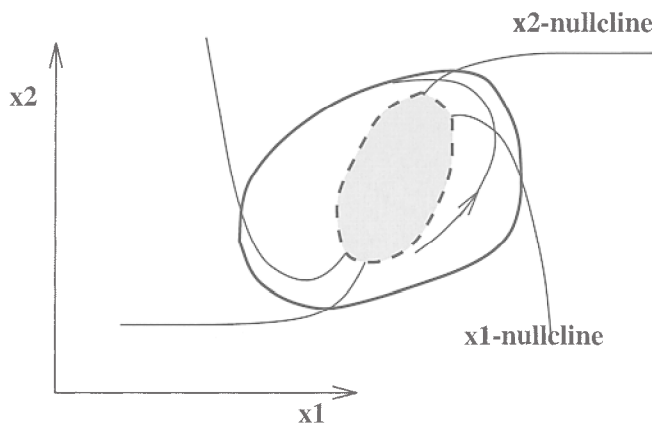


Figure 6. Phase plane for bistable regimen of parameters. Nullclines are shown, as is the stable periodic orbit (dark line), the unstable periodic orbit (dashed line), and representative trajectories (thin lines). The gray area denotes the domain of attraction for the fixed point. The rest of the plane is attracted to the stable periodic orbit.

In systems with more than two components, it is difficult to make any general comments on behavior. For symmetrically coupled networks with no self-connections, a complete analysis can be given (Hopfield, 1982). Weakly coupled systems of intrinsically oscillatory networks can be analyzed with the techniques described in *CHAINS OF OSCILLATORS IN MOTOR AND SENSORY SYSTEMS*. Bifurcation methods and averaging techniques can often be used to reduce higher-dimensional systems to a simpler set of equations that is in a much lower dimension (see Hoppensteadt and Izhikevich, 1997; see also *CANONICAL NEURAL MODELS*).

Numerical Methods

The computer is a valuable adjunct in the exploration of systems of differential equations. For this article, I have used a program called XPPAUT that is available for both Windows 95/NT/98 computers (Winpp) and Unix workstations. Both are available through <http://www.pitt.edu/~phase>. To get *global* pictures of the dynamics as one or two parameters are varied, a powerful numerical package written by Doedel et al. (1997) called AUTO can be used. A version is available at <http://indy.cs.concordia.ca/auto/>.

Road Map: Dynamic Systems

Related Reading: Canonical Neural Models; Cortical Population Dynamics and Psychophysics; Dynamics and Bifurcation in Neural Nets; Pattern Formation, Neural

References

- Beer, R. D., 1995, On the dynamics of small continuous time recurrent neural networks, *Adapt. Behav.*, 3:469–509. ♦
- Doedel, E., Champneys, A., Fairgrieve, T., Kuznetsov, Y., Sandstede, B., and Wang, X. J., 1997, AUTO97: Continuation and bifurcation software for ordinary differential equations (with HomCont), Montreal: Computer Science Department, Concordia University.
- Edelstein-Keshet, L., 1988, *Mathematical Models in Biology*, New York: Random House. ♦
- Ellias, S. A., and Grossberg, S., 1975, Pattern formation, contrast control, and oscillations in the short-term memory of shunting on-center off-surround networks, *Biol. Cybern.*, 20:69–98.
- Ermentrout, G. B., 1998, Neural networks as spatio-temporal pattern-forming systems, *Rep. Prog. Phys.*, 61:353–430. ♦
- Ermentrout, G. B., and Cowan, J. D., 1979, Temporal oscillations in neuronal nets, *J. Math. Biol.*, 7:265–280.
- Guckenheimer, J., and Holmes, P. J., 1983, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Heidelberg: Springer-Verlag.
- Hopfield, J. J., 1982, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, 79:2554–2558.
- Hopfield, J. J., 1984, Neurons with graded responses have collective computational properties like those of two-state neurons, *Proc. Natl. Acad. Sci. USA*, 81:3088–3092.
- Hoppensteadt, F., and Izhikevich, E., 1997, *Weakly Connected Neural Networks*, New York: Springer-Verlag. ♦
- Murray, J. D., 1989, *Mathematical Biology*, Heidelberg: Springer-Verlag.
- Pinto, D., Brumberg, J., Simons, D., and Ermentrout, B., 1996, A quantitative population model of whisker barrels: Re-examining the Wilson-Cowan equations, *J. Comput. Neurol.* 3:247–264.
- Rinzel, J., and Ermentrout, G. B., 1998, Analysis of neural excitability and oscillations, in *Methods of Neuronal Modelling: From Synapses to Networks*, 2nd ed. (C. Koch and I. Segev, Eds.), Cambridge, MA: MIT Press.
- Segel, L. A., 1984, *Modelling Dynamic Phenomena in Molecular and Cellular Biology*, New York: Cambridge University Press.
- Shepherd, G. M., 1990, *The Synaptic Organization of the Brain*, Oxford, Engl.: Oxford University Press.
- Tsodyks, M. V., Skaggs, W. E., Sejnowski, T. J., and McNaughton, B. L., 1997, Paradoxical effects of external modulation of inhibitory interneurons, *J. Neurosci.*, 17:4382–4388.
- Wilson, H. R., and Cowan, J. D., 1972, Excitatory and inhibitory interactions in localized populations of model neurons, *Biophys. J.*, 12:1–24.

Philosophical Issues in Brain Theory and Connectionism

Andy Clark and Chris Eliasmith

Introduction

In this article, we highlight three questions: (1) Does human cognition rely on structured internal representations? (2) How should theories, models, and data relate? (3) In what ways might embodiment, action, and dynamics matter for understanding the mind and the brain?

The first question concerns a fundamental assumption of most researchers who theorize about the brain. Do neural systems exploit classical compositional and systematic representations, distributed representations, or no representations at all? The question is not easily answered. Connectionism, for example, has been criticized for both holding and challenging representational views. The second question concerns the crucial methodological issue of how results emerging from the various brain sciences can help to constrain cognitive scientific models. Finally, the third question focuses attention on a major challenge to contemporary cognitive science: the challenge of understanding the mind as a controller of embodied and environmentally embedded action.

Does Cognition Need Representations?

This question is the most difficult and least well defined of our three questions. But it addresses one of the most philosophically interesting features of many connectionist models, especially those most closely related to brain theory. The intuition is that connectionism poses a challenge to the classical view of the brain as a syntax-sensitive engine. This idea involves depicting all or most of human cognition as involving something akin to logical operations applied to something akin to linguistic (sentential) structures. The prime philosophical exponent of the identification of genuine cognitive processes with such operations on quasi-sentential entities is Fodor (see Fodor, 1987; Fodor and Pylyshyn, 1988). Fodor argues in favor of an innate symbolic code (the *language of thought*) and mental processes involving operations defined over the syntactically structured strings of that code. The underlying image is of an inner economy in which symbol strings are operated on by procedures sensitive to the structure of the string.

In contrast, a typical trained-up network does not employ grammatical strings or, a fortiori, processing operations sensitive to the structure of such strings. Instead, we find prototypical complexes of properties represented in a high-dimensional space (see Churchland, 1995). The space is highly organized in the sense that data items that need to be treated in closely related ways become encoded in neighboring regions of the space. It is this *semantic metric* that allows the network to generalize and to respond well under conditions of noise, for example. But this organization of the encoded knowledge does not amount to the provision of a genuine syntax. One way to see this difference is to ask what rules of combination of represented elements apply, and in what systematic ways we can operate on complexes of represented items. As Fodor and Pylyshyn (1988) point out, there is no analog in distributed representations for the logical operations of detaching an element from one string (complex representation) and adding it to another.

Nevertheless, a variety of connectionist techniques have been developed to allow for structure-sensitive processing, but such techniques have been described (van Gelder, 1990) as providing *functional*, as opposed to *concatenative*, compositional structure. A complex representation has concatenative structure if it embeds the individual constitutive elements unaltered within it. It has functional compositional structure if such components are usable or

retrievable, but the complex expression does not itself embed unaltered tokens of these parts. Most connectionist schemes for dealing with compositional structure are functionally compositional (e.g., RAAM architectures, tensor product encodings, holographic reduced representations [HRRs]), although synchrony binding is concatenative. (For a review, see CONNECTIONIST AND SYMBOLIC REPRESENTATIONS.) Of these, HRRs are perhaps best suited to bridging the traditional gap between connectionist and symbolicist approaches to understanding language-like processing. HRRs are supremely structure sensitive, do not suffer from the dimensional increases of tensor products, and can be implemented in standard connectionist networks, yet they are not concatenative (Eliasmith and Thagard, 2001).

In our opinion, a major benefit of exploring the space of connectionist cognitive models is that it may help us expand our sense of the possible nature of internal representation and hence better understand what is truly essential to notions such as *structure*, *syntax*, and *complex representation*. Doing so, we may discover which aspects of our models are simply artifacts of our (over)familiarity with one representational format, viz., the format of atomic elements and grammar common to language and logic.

How Do Theories, Models, and Data Relate?

As a computational formalism, connectionism is quite powerful, allowing us to approximate nearly any function or performance profile that we desire. However, the mere fact that some input-output pattern *P* is found in human cognition and can be mimicked using some connectionist model is, in itself, of only marginal psychological interest. The demands of cognitive science, unlike those of artificial intelligence, require more. Ideally, we must provide models that are both consistent with neurological data and comprehensible. However, the relation between data and models is not unidirectional. Models are constrained by data, but they also help us determine what sorts of experiments to use in looking for more relevant details. The role of theory in unifying the brain sciences and connectionism is an important but (for the time being) mysterious one (but see Eliasmith and Anderson, 2002, chap. 1, for one possibility).

The data that constrain models come largely from two sources: higher levels (i.e., from the mind “down”) from work in disciplines such as psychology and psycholinguistics, and lower levels (i.e., from the neuron “up”), from work in neuroscience and brain theory with behavior as the meeting ground for these diverse approaches. From psychology and psycholinguistics we can extract vast bodies of constraining data that go far beyond the mere specification of a task-specific input-output mapping. Such data can concern, for example, the relative difficulty of parsing certain sentences or solving certain problems, the time course of problem solving, the developmental profile of skill acquisition, and the way in which new and old knowledge interact in the context of new learning (for detailed examples, see Karmiloff-Smith, 1992; see also DEVELOPMENTAL DISORDERS).

For current purposes, however, it is the lower-level constraints that we seek to highlight. The question here concerns the proper relation between connectionist-computational modeling and the detailed constraints emerging from the various brain sciences. Such sciences include neuroanatomy, neurochemistry, lesion studies, and research at the single-cell, circuit, and systems level. It seems clear that any acceptable model of human information processing must respect the results of such studies. To do so, some intelligible

relation must exist between the theories put forward by, for example, connectionist-computational modeling and the entities and lawful interactions studied by the brain sciences. It is a duty sadly neglected by both classical artificial intelligence and a great deal of connectionist work to make some effort to display the precise nature of such relations.

Such a task is complicated by the variety of levels of interest that may characterize the brain sciences. These include the levels of biochemical specification: single cells, circuits, subsystems, and networks of subsystems. Marr's suggestion that studies at each level can be independently pursued is highly dubious. Our top-level decomposition of a task into subtasks appropriate for computational modeling may be challenged once we become familiar with the distribution of information-processing resources in the brain. What we originally thought of as two distinct functions may actually share circuitry in the brain (see Arbib, 1989). Such a result will not be devoid of psychological significance, since it will figure in an explanation of the breakdown profile as revealed by, for example, lesion studies of the system.

How, then, should we conceive the bridge between idealized artificial intelligence models and brain theory? It is precisely the complex relations between implementation and function that have spawned a recent surge of interest in computational neuroscience. With the explicit goal of taking biological constraints as seriously as computational ones, computational neuroscience has begun to explore a vast range of realistic neural models. For example, Reike et al. (1997) provide an information-theoretic analysis of spike trains, allowing accurate stimulus signal reconstruction. The combination of such spike train analyses and, for example, Abbott's higher-level discussions of basis function representations can provide valuable insights into the functioning of populations of neurons (see Eliasmith and Anderson, 2002). Though preliminary, the tools developed by such research are promising candidates for generating biologically realistic connectionist models.

Such models should prove useful in providing constraints of their own. Insights from basis function analyses suggest new experiments for neurophysiologists. In particular, it seems that neurons may have higher-dimensional tuning profiles than previously imagined. Although neurological techniques for determining complex profiles have yet to be perfected, connectionist modeling suggests that such tuning properties are important to the everyday functioning of neurons. So, not only does biology inform the construction of computational models, but, ideally, those same models can help suggest important experiments for neuroscientists to perform. In this sense, models and data can be mutually beneficial. Of course, the benefits are highly constrained by assumptions of both the model and the experimental design.

Although no model can be expected to do justice to all aspects of its target, what justice it can do depends on the biological realism of the assumptions behind the model. Biological realism, of course, can be incorporated into a model in many ways, such as by including neurochemical diffusion, single neuron morphology, spike train statistics, neuroanatomical constraints, population dynamics, or system-level organization (see Eliasmith and Anderson, 2002, for examples). In any case, what we can and should expect from a modeler is a clear statement of what aspects of the target phenomenon are supposed to be explained, and (if it is a computational model) at what level the computational story is intended to capture real neurophysiological facts. Successful attempts to exploit the close relation between experiment and model are still something of a rarity. This is largely because theoreticians (typically mathematicians, physicists, and engineers) and experimentalists (typically neuroscientists and biologists) do not yet have many conceptual tools in common. In order to reap the benefits of mutual, interlevel constraint, this will likely have to be rectified.

In What Ways Might Embodiment Matter for Understanding the Mind and the Brain?

In recent years, an important challenge has been issued to cognitive science. It stems from the work of researchers espousing the *dynamicist hypothesis* (van Gelder, 1995). The dynamicist commitment to making time central to cognitive modeling is inspired by the broader realization that cognitive systems are real physical systems acting in the real world in real time. Given the finite, though vast, computational resources of the brain, it also seems that evolution has often off-loaded complex computational tasks to the body and to the environment. This double "situatedness" of cognitive systems needs to be reckoned with if we are to develop an accurate picture of precisely the kinds of computation neural systems perform. Connectionism and brain theory must conspire to explain this kind of representational and computational economy. Thus, while looking *inside* to the brain and the results of neuroscience, we can not afford to turn a blind eye to constraints and resources that come from the *outside*, from the gross body and environment of a cognitive system (Clark, 1997).

Consider vision. There is now a growing body of work devoted to so-called *animate vision* (Ballard, 1991). The key insight here is that the task of vision is not to build rich inner models of a surrounding three-dimensional reality, but rather to use visual information efficiently and cheaply in the service of real-world, real-time action. Animate vision thus rejects Marr's analysis, what Churchland et al. nicely dub the paradigm of "pure vision"—the idea (associated with work in classical AI and in the use of vision for planning) that vision is largely a means of creating a world model rich enough to let us "throw the world away," targeting reason and thought to the inner model instead. Real-world action, in these "pure vision" paradigms, functions merely as a means of implementing solutions arrived at by pure cognition.

The animate vision paradigm, by contrast, gives action a starring role. Computational economy and temporal efficiency are purchased by a variety of bodily actions and local environment-exploiting tricks and ploys, including

- the use of cheap, easy-to-detect (possibly idiosyncratic) environmental cues (e.g., searching for Kodak film in a drug store: Seek "Kodak yellow");
- the use of active sensing (e.g., use motor action, guided by rough perceptual analysis, to seek further inputs yielding *better* perceptual data—move head and eyes for better depth perception, etc.); and
- the use of repeated consultations of the world in place of rich, detailed inner models.

Ballard et al. (1997) have recently demonstrated that subjects do not bind color and location information in a block-copying task until it is absolutely required by current problem solving. As a result, changes made to the display (such as switching the color of blocks during a saccade) are very often undetected.

Vision, this body of work suggests, is a highly active and intelligent process. It is not the passive creation of a rich inner model so much as the active retrieval (typically by moving the high-resolution fovea in a saccade) of useful information *as it is needed* from the constantly present real-world scene. Ballard et al. speak of "just-in-time representation," while the roboticist Rodney Brooks coined the phrase, "The world is its own best model" (Brooks, 1991). The combined moral is clear: Vision makes the most of the persisting external scene, and gears its computational activity closely and sparingly to the task at hand.

The general thrust of the animate vision research program, however, is not to reject the ideas of internal models and representations so much as to reconfigure them in a sparser and more interactive

image. We thus read of inner databases that associate objects (such as car keys) and locations (on the kitchen table) of internal feature representations, of indexical representations, and so on. What is being rejected is not the notion of inner content-bearing states per se, but only the much stronger notion of rich, memory-intensive, all-purpose forms of internal representation.

The crucial distinction, it seems to us, is thus not between representational and nonrepresentational solutions so much as between rich and action-neutral forms of internal representation (which may increase flexibility but require additional computational work to specify a behavioral response) and sparse and action-oriented forms (which exploit the body and world and which begin to build the response into the representation itself).

Discussion

Our vision of basic biological reason is changing rapidly. There is a growing emphasis on the computational economies afforded by real-world action, and an increasing appreciation of the way larger structures (of agent and artifacts) both scaffold and transform the shape of individual reason. These twin forces converge on a rather more minimalist account of individual cognitive processing, an account that tends to eschew rich, all-purpose internal models and sentential forms of internal representations. Such minimalism, however, has its limits. Despite some ambitious arguments, there is currently no reason to doubt the guiding vision of individual agents as loci of internal representations and the individual agents as users of a variety of inner models. Rather than opposing representationalism against interactive dynamics, we should be embracing a broader vision of the inner representational resources themselves.

The sciences of the mind are thus in a state of productive flux, the product of multiple converging influences coming from real-world robotics, systems-level neuroscience, cognitive psychology, evolutionary theory, AI, and philosophical analysis. This flux has forced us to reconsider earlier accounts of the relation between theory, models, and data relevant to cognitive systems. More important, we can see a new vision of mind emerging. The point at which many of these influences currently converge is captured by seeing mind as in essence a controller of embodied and environmentally embedded action. Mind is an organ for orchestrating real-time responses to a real world.

One major player in these recent events has been the explosion of work on artificial neural networks (ANNs). Such networks amounted to an existence proof of the possibility of adaptive intelligent behavior without reliance on explicitly formulated rules or language-like data structures. Moreover, the networks integrated representation and action in a very direct manner: knowledge became encoded in a form dictated by its use in a particular type of problem solving. But the neural networks revolution was incomplete. It was incomplete because it was still burdened with much of the unnecessary baggage of the previous, disembodied, symbol-crunching approach to understanding cognition. Mind was still treated as an essentially timeless locus of abstract problem-solving capacities.

All this changed with the surge of interest, in the late 1980s and early 1990s, in what became known as autonomous agent research (see, e.g., essays in Beer, Ritzmann, and McKenna, 1993). This research aimed to model and understand the adaptive success of single, complete, embodied systems: insects that walk and seek food, the cockroach's amazingly sophisticated mechanisms for detecting and evading attackers, robots that learn to swing from branch to branch using real mechanical arms, and many other kinds

of systems. Many of these models exploit ANNs as control systems. But the constraints on success became very different.

Finally, the constraints on computation using ANNs are very different from the constraints on real biological computation. It is here that the relation among theory, model, and data again becomes pivotal. Interestingly, reconceptualizing mind in each of these previous cases has depended on rethinking the relevant constraints (i.e., linguistic versus nonlinguistic symbols, partial versus full-bodied systems). Introducing the complexities of natural neural computation is bound to have a similar impact on our concept of mind.

Many important questions remain open. Can work in ANNs come to grips with the real complexity of biological computation? What kinds of systems-level models can help make sense of the complex balance between specialization and cooperation that we find in real brains? Can a representation-sparse approach make headway with all aspects of human cognition, or is it limited to cases of perceptuomotor control and on-line reasoning? How does the command of public language impact and transform human thought and reason?

The cognitive science of the biological, embodied mind is still in its infancy, and the full power and scope of the new vision remain to be determined. But the issues raised will, we believe, shape the agenda of the next decade of research into mind and its place in nature.

Road Map: Psychology

Related Reading: Artificial Intelligence and Neural Networks; Consciousness, Neural Models of; Perspective on Neuron Model Complexity; Structured Connectionist Models

References

- Arbib, M. A., 1989, *The Metaphorical Brain 2: Neural Networks and Beyond*, New York: Wiley-Interscience.
- Ballard, D., 1991, Animate vision, *Artif. Intell.*, 48:57–86.
- Ballard, D., Hayhoe, M., Pook, P., and Rao, R., 1997, Deictic codes for the embodiment of cognition, *Behav. Brain Sci.*, 20:723–767.
- Beer, R., Ritzmann, R., and McKenna, T., Eds., 1993, *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, London: Academic Press. ♦
- Brooks, R., 1991, Intelligence without representation, *Artif. Intell.*, 47:139–159.
- Churchland, P. M., 1995, *The Engine of Reason, The Seat of the Soul*, Cambridge, MA: MIT Press. ♦
- Clark, A., 1997, *Being There: Putting Brain, Body, and World Together Again*, Cambridge, MA: MIT Press. ♦
- Eliasmith, C., and Anderson, C. H., 2002, *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*, Cambridge, MA: MIT Press. ♦
- Eliasmith, C., and Thagard, P., 2001, Integrating structure and meaning: A distributed model of analogical mapping, *Cognit. Sci.*, 25:245–286.
- Fodor, J., 1987, *Psychosemantics: The Problem of Meaning in the Philosophy of Mind*, Cambridge, MA: MIT Press. ♦
- Fodor, J., and Pylyshyn, Z., 1988, Connectionism and cognitive architecture: A critical analysis, *Cognition*, 28:3–71.
- Karmiloff-Smith, A., 1992, *Beyond Modularity: A Developmental Perspective on Cognitive Science*, Cambridge, MA: MIT Press.
- Reike, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W., 1997, *Spikes: Exploring the Neural Code*, Cambridge, MA: MIT Press.
- van Gelder, T., 1990, Compositionality: A connectionist variation on a classical theme, *Cognit. Sci.*, 14:355–384.
- van Gelder, T., 1995, What might cognition be, if not computation? *J. Philos.*, 91:345–381.

Photonic Implementations of Neurobiologically Inspired Networks

B. Keith Jenkins and Armand R. Tanguay, Jr.

Introduction

Technological implementations of neural networks in both software and hardware forms have been largely motivated by biological neural networks, but also include network topologies, synaptic interconnection rules, and neuron unit functionalities that combine to yield novel system-level properties. Software implementations of such networks are highly flexible in design and reconfiguration but are time-, power-, and computational-resource-consumptive for even modest numbers of neuron units and interconnections. Neural network implementations in both electronic and photonic hardware are designed to circumvent these limitations in applications that require compact systems, low latencies, and high computational throughputs.

In VLSI-based neural networks (Mead, 1989), the neuron units and weighted (synaptic) interconnections are incorporated on one or more planar microelectronic chips (see NEUROMORPHIC VLSI CIRCUITS AND SYSTEMS; ANALOG VLSI IMPLEMENTATIONS OF NEURAL NETWORKS; DIGITAL VLSI FOR NEURAL NETWORKS; SILICON NEURONS). An important advantage of this approach is the capability for near-term technology insertion, with leverage provided by a well-established technology base. An equally important limitation, however, is the difficulty of scaling up or interconnecting multiple neural chips to incorporate large numbers of neuron units in highly interconnected architectures without significantly increasing the computational time. This inherent trade-off derives from the limited pin counts, off-chip communication bandwidths, and on-chip interconnection densities available in both current generation and projected chip designs.

In this article we consider the photonic implementation of neurobiologically inspired networks, in which optical (free-space or through-substrate) techniques are utilized to enable an increase in the number of neuron units and the interconnection complexity by using the off-chip (third) dimension (Jenkins and Tanguay, 1992; Wagner and Psaltis, 1993; Jutamulia, 1994). This merging of optical and photonic devices with electronic circuitry provides additional features such as parallel weight implementation, adaptation, and modular scalability.

General Principles Extracted from Biological Systems

In this section, we discuss the key general principles that can be extracted from neurobiological systems and then crafted in hybrid electronic/photonic form, a base technology substrate that exhibits wide-ranging differences with respect to human wetware (Hubel, 1988). Although the discussion below is framed within the human visual system (retina through early visual cortex), these architectural principles apply throughout the mammalian brain to a large extent, and can potentially be used to advantage in biologically inspired neural systems.

Biological vision systems exhibit a number of common themes that have pertinence to hardware implementations, including (1) a propensity for layering of the processing architecture (Wandell, 1995), (2) the employment of massive parallelism with simple local processing units and minimal local storage within each processing unit, (3) the use of a multiplicity of neuron unit types and associated fan-out and fan-in patterns, (4) the incorporation of dense synaptic/

dendritic interconnections at all scales (from local to global, among multiple brain regions) with a high degree of fan-out and fan-in at each processing node, (5) adaptivity on multiple time scales as exhibited by both short- and long-term plasticity, (6) distributed storage of information, and (7) an associative memory organizational construct.

Although many of these themes can be individually incorporated into hardware implementations using existing technology, to our knowledge no implementation has included all of these themes together using a single technology base. Primate visual systems, for example, use several types of photoreceptors and neurons at the lowest levels of vision processing, with primarily local, fixed, and weighted interconnections among multiple layers within the retina (Dowling, 1992). The photoreceptors are densely packed, ranging from about $1\text{--}3 \times 10^7 \text{ cm}^{-2}$ in the fovea ($1\text{-}\mu\text{m}$ -diameter cones) to $4 \times 10^6 \text{ cm}^{-2}$ in the periphery, with a mixture of 4- to $10\text{-}\mu\text{m}$ -diameter cones separated by a much higher density of $1\text{-}\mu\text{m}$ -diameter rods (Wandell, 1995). This density can be instructively compared with the current pixel densities of solid-state imaging sensor arrays (including focal plane arrays in the visible, infrared, and ultraviolet frequencies; CCD arrays; and active pixel sensor (APS) arrays), which range from about $1 \times 10^6 \text{ cm}^{-2}$ to $4 \times 10^6 \text{ cm}^{-2}$. Current smart pixel arrays have not yet achieved even these densities because of the incorporation of local processing circuitry within each pixel.

The biological propensity for incorporation of dense interconnections is evident within the retina and the lateral geniculate nucleus, and extends into the lowest levels of the visual cortex. Throughout, the interconnection mappings tend to be local, highly regular (retinotopic), and only partially adaptive. Higher up the processing stream (within the primate visual cortex), interconnections tend to become gradually less local, less regular, and more adaptive, with a degree of interconnectivity (fan-out from and fan-in to a given neuron) that is typically 10^3 to 10^4 (Hubel, 1988; Dowling, 1992; Wandell, 1995).

The biological imperative for layering is also of considerable interest to examine further. In primate visual systems, layering accomplishes a number of computationally important functions. (1) Layering provides a convenient mechanism for the implementation of multiple concatenated operations comprising nonlinearities and weighted fan-out/fan-in functions. The latter operation can be viewed as the convolution of a 2D input function with a set of 2D kernels (weighting functions), and provides the basis for implementing both space-invariant and space-variant operations across multiple spatial scales. The separation of a given complex operation into several sequential steps of nonlinearity/convolution also allows access to intermediate scale results for both feedforward and feedback connections that project beyond intervening layers, as observed throughout biological vision systems. (2) Layering also provides for the implementation of higher-order complexity (hierarchical) operations that can be derived from simple primitives implemented over multiple spatial scales. (3) Layering naturally provides for the hierarchical buildup of the size of the receptive field, so that nonlocal operations such as contrast enhancement and color constancy can be implemented with invariance to object size. (4) Finally, layering carries with it the potential for increased algorithmic efficiency, in that certain operations (e.g., even certain linearly decomposable convolutions at a given kernel size) can be performed in multiple layers with less

cost in computational resources (e.g., fan-out from neurons via axons, synapses, and dendrites; total number of equivalent primitive operations; computational energy).

The Holographic Paradigm

With these general principles in mind, in this section we consider the potential implementation of neurobiologically inspired networks based on the holographic recording and readout of weighted interconnection patterns. In volume holography, the storage of a set of recorded holographic images in a volume holographic optical element by means of the coherent interference between a set of image-bearing signal beams and a corresponding set of reference beams (multiplexed in angle, wavelength, or position) can also be thought of as the storage of a set of weighted interconnection patterns; each input pixel (picture element) in a given reference beam is connected to a given output pixel in the reconstructed image with a weight (diffraction efficiency) that governs the brightness or intensity of the reconstructed pixel. Sets of weighted interconnection patterns can be read out in parallel to form multiple inputs to each output pixel, providing a dense interconnection network with a high degree of fan-out from each reference beam pixel and fan-in to each output image pixel.

The storage of such weighted interconnection patterns in volume holographic optical elements exhibits a number of characteristics that are similar to characteristics observed in the neurobiological processes of memory and learning as well as in many neural networks (Pribram, 1991). For example, the information (memory) associated with the set of holographically stored weighted interconnection patterns is distributed over a significant portion of the volume hologram; such distributed memory storage is also characteristic of neural networks. Holographically stored weighted interconnection patterns can implement dense interconnections at all scales from local to global, and can exhibit degrees of fan-out and fan-in from unity (a point-to-point connection) to upward of 10^3 or 10^4 . Recall of such stored patterns can be configured in many ways, including both retinotopic-like (highly structured) and associative-memory-like mappings, and can provide for both feedforward and feedback connections, depending on the specific network architecture.

If appropriate real-time recording and readout materials are employed, such as photorefractive crystals or polymers, the interconnection weights can be updated by means of holographic recording principles, so that weight updates analogous to those characteristic of adaptivity and learning in certain neural network models can be performed. These weights will typically decay with time (and also with multiple exposures), suggestive of short-term memory. In many such real-time materials, the capability exists to "fix" all or part of the recorded interconnection patterns, thereby greatly extending the storage time, suggestive in turn of long (or at least longer) term memory. In addition, the properties of real-time holographic recording lead naturally to the incorporation of a dependence for weight updates on temporal correlations between the signal and reference beams (as derived, for example, from two temporally correlated neuron units).

In the following sections, we describe several approaches for implementing photonic neurobiologically inspired networks. All such architectures incorporate dense fan-out/fan-in interconnections but range in the type and degree of connectivity from compact photonic multichip modules with local interconnections between layers to photonic neural networks that incorporate global holographic interconnections. Descriptions of the various photonic components that comprise such architectures can be found in the previous edition of the *Handbook* (Jenkins and Tanguay, 1995).

Compact Photonic Multichip Modules for Layered Networks

A conceptual diagram of one possible three-dimensional (3D) integrated electronic/photonic multichip module (PMCM) structure is shown in Figure 1 (Veldkamp, 1993; Tanguay et al., 2000; Tanguay and Jenkins, 2002). Multiple layers of pixellated silicon VLSI chips (chips that are divided into arrays of nearly identical functional regions) are densely interconnected by a combination of electronic, optical, and photonic devices to produce either a space-invariant or space-variant degree of fan-out and fan-in to each individual pixel (neuron unit, or processing node). These weighted fan-out/fan-in interconnections are suggestive of the axonal projections, synapses, and dendritic tree structures that characterize biological organisms. The use of optical and photonic devices in particular allows for the implementation of these interconnections *between* adjacent physical layers within the stack of chips. In addition, the use of silicon VLSI for neuron units or processing elements allows considerable flexibility in the neuron model or neuron-unit function implemented.

In one such implementation (Tanguay et al., 2000; Tanguay and Jenkins, 2002), shown schematically in Figure 2, two-dimensional (2D) arrays of bottom-emitting vertical-cavity surface-emitting lasers (VCSELs) fabricated on a gallium arsenide (GaAs) substrate provide optical outputs from a given integrated layer of the structure. These VCSEL arrays are flip-chip bonded on a pixel-by-pixel basis to the silicon VLSI chips, which typically incorporate local optical detectors (for optical inputs from the previous layer) and electronics comprising processing elements (either acting alone or in concert with electrical inputs from nearest and next-nearest neighbors within the plane), memory elements (in the analog or digital domain), and VCSEL drivers. Proximity-coupled diffractive optical element (DOE) arrays, designed to incorporate both focal power (lens) and weighted fan-out functions, are used to establish interconnections that are modulated (temporally varied) in intensity by each individual VCSEL element and its associated silicon driver circuit. Alternatively, separate concatenated DOE and microlens arrays can be used.

Other implementations of compact photonic multichip modules include a similar 3D stacked structure using a different materials system and an array of one-to-one (digital) photonic interconnections between adjacent chips (Bond et al., 1999); and a set of

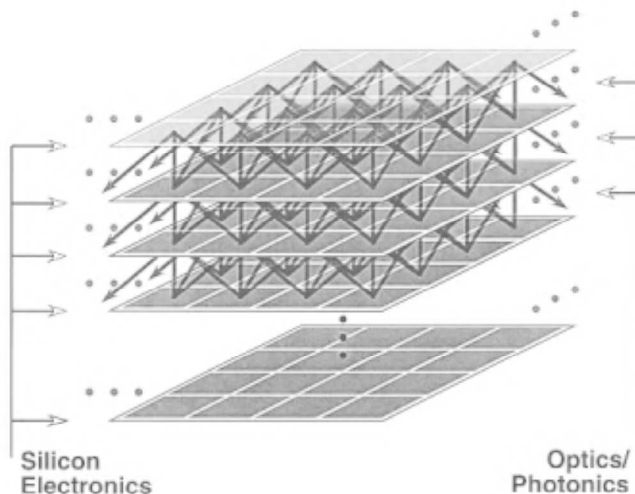


Figure 1. Three-dimensional photonic multichip module concept.

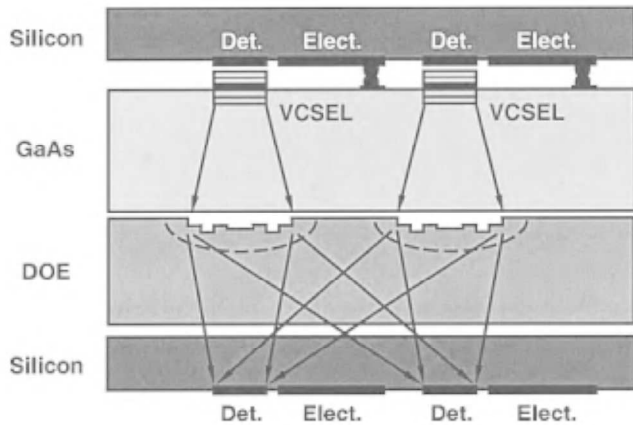


Figure 2. Multilayer compact photonic multichip module. Only two cells within an $N \times N$ array are shown, as well as only two (of M) silicon chip layers.

chips laid out on a planar substrate, with neural network interconnections implemented in the third dimension using photonics (Fey et al., 2000).

For the specific case of adaptive vision sensors, the design of the individual silicon VLSI chips, and in particular the use of spatiotemporal multiplexing techniques for network implementation and signal processing functions, is motivated by the recent development of several promising neurobiologically inspired vision algorithms that can potentially be mapped onto the emerging 3D PMCM platform (Tanguay et al., 2000, and references therein).

Extended Photonic Multichip Modules for Layered Networks

As the level of representation extends from low-level vision through mid-level vision to high-level vision operations, interconnections tend to become both more sparse and more global; in some cases, particularly between functionally partitioned vision processing modules, dense global interconnections may be required. Both local and global interconnection cases can potentially be accommodated within the PMCM architecture by incorporating novel stratified volume diffractive optical elements (SVDOEs; Tanguay et al., 2000, and references therein) as shown schematically in Figure 3. These SVDOEs consist of multiple layers of proximity-coupled and aligned DOEs that implement either space-variant or space-invariant interconnection patterns with properties characteristic of volume holograms. They also offer the advantages of planar fabrication methods compatible with VLSI design rules.

In Figure 3, several sets of compact PMCMs that implement primarily local interconnections, and perhaps also implement hierarchical functions, are interconnected across the faces of a cubic submodule by each SVDOE, offering the potential for more global interconnectivity. The mappings might be retinotopic in some cases, and columnar or fully space-variant in others.

Volume Holographic Systems for Large-Scale Adaptive Networks

Volume holographic optical elements are capable of global, dense interconnections that are adaptive at the hardware level. The potential exists for the implementation of large numbers of weighted

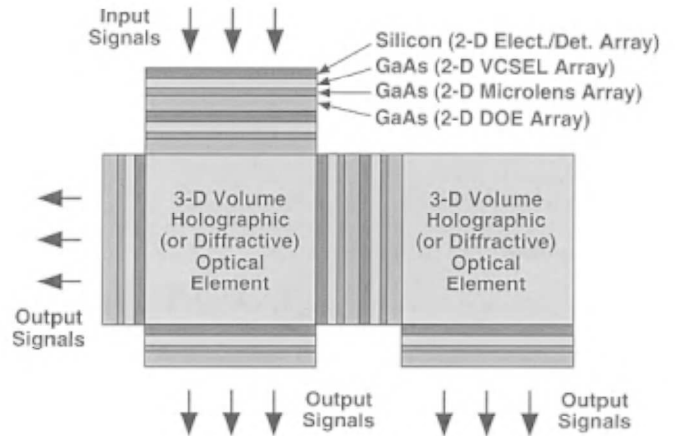


Figure 3. Densely interconnected extended photonic multichip module.

interconnections (e.g., 10^{10} in one module) that adapt by means of outer-product learning algorithms (e.g., Hebbian learning, or single- or multiple-layer least-mean squares [backpropagation], including weight decay). These learning algorithms exploit the physics of holographic (typically photorefractive) materials, the properties of which currently limit both adaptation and retention times. Such holographic interconnection systems for adaptive neural network implementations are not yet practical, but have been extensively researched (e.g., Jenkins and Tanguay, 1992; Wagner and Psaltis, 1993; Li et al., 1996; and references therein).

A conceptual diagram of one particular optical implementation of a large-scale artificial neural network with both local and global fan-out/fan-in adaptive interconnections is given in Figure 4. Many related implementations are possible, with similar overall characteristics. Two smart pixel spatial light modulators (SLMs, such as pixellated silicon VLSI chips mated to GaAs chips that comprise arrays of optical modulators rather than arrays of lasers; see also Jenkins and Tanguay, 1995, p. 679; Worchesky et al., 1996) are used to implement a 2D array of training term generators (SLM_1) and a 2D array of neuron units (SLM_2). A volume holographic

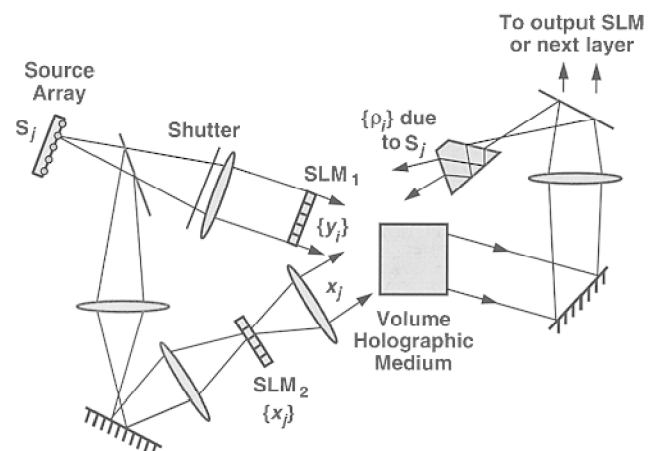


Figure 4. Photonic volume holographic system for large-scale adaptive networks; the Hebbian case is depicted. Beams from the source array on the left serve to read out the signals generated by both SLMs; the neuron-unit inputs to SLM_2 are not shown.

medium serves as the key interconnection element. The interconnection outputs occur in a plane that typically comprises the detector array of another SLM (for example, the output SLM). This SLM may serve as the input to subsequent signal processing stages, or as the training-term array (SLM₁) or input neuron-unit-array (SLM₂) in the case of a system with feedback.

In the computing phase (i.e., hologram readout), the beam from each input-neuron-unit pixel of SLM₂ (x_j) illuminates the holographic medium at a unique position, angle, and/or wavelength. The selective nature of volume holograms then allows only the appropriate interconnection weights w_{ij} to be multiplied by each input beam, yielding the set of products $w_{ij}x_j$. The output beam optics then sums over j of these terms at the location of each output term ρ_i in the interconnection output plane, yielding $\rho_i = \sum_j w_{ij}x_j$.

During the learning phase (i.e., hologram recording), each pixel of SLM₁ generates the appropriate training term, δ_i . An exposure is made of the interference pattern between beams emanating from the two SLMs in Figure 4. With appropriate choices of parameters, this exposure can increment the value of each interconnection weight w_{ij} by an amount that is approximately proportional to $\delta_i x_j$. In the Hebbian case, for example, $\delta_i = y_i$, in which y_i is the output-neuron-unit signal. It should be noted that when designing an architecture that will generate and record these weight increments, care must be taken to ensure that the appropriate interference terms are recorded with minimal crosstalk. One of the key differences among full-scale optical neural network architectures is the technique used to avoid such crosstalk (Wagner and Psaltis, 1993). In addition, the actual weight increment that physically occurs in holographic systems depends on the particular architecture employed (Wagner and Psaltis, 1993; Petrisor et al., 1996).

The interconnections within the particular implementation depicted in Figure 4 are based on a technique for multiplexed volume holography that uses double angular multiplexing and incoherent/coherent recording and readout (Jenkins and Tanguay, 1992). This interconnection technique exhibits an advantageous combination of total number of channels and interchannel crosstalk, in the case of high total optical throughput efficiency. One of its key features is the use of a source array that consists of an array of lasers, each of which generates light independently of the others. Because of this independence, a pair of beams from *different* lasers is not capable of interfering and recording a hologram, whereas a pair of beams generated from the *same* laser can interfere and thereby record a hologram in a holographic medium. From this source array, a set of coherent beam pairs is formed, one pair from each laser, that can record holograms pairwise.

In the computing phase of most volume hologram-based implementations of neural networks, the overall computing time is determined by the SLM response time. During learning, the holographic material sensitivity or response time typically limits the weight increment rate, which in turn puts an upper bound on the achievable learning gain constant; such holographic material response times depend on the material and vary over many orders of magnitude from one material to another. Similarly, the minimum decay rate (or maximum retention time) of holograms in read/write holographic materials puts a lower bound on the weight decay constant, and is similarly material dependent.

Large-scale optical *nonadaptive* networks are important as well. In this case the interconnection hologram need not be recorded in accordance with a specific learning algorithm. If the weights are known a priori, then any standard technique can be used to pre-record the hologram. In many cases, however, the weights may not be known. A common scenario may involve the training of a “master” network; once it has been trained, multiple copies of the network would be produced. If the network is large, and particularly if it utilizes volume holographic interconnections, then making direct copies of the volume hologram may be more practical than

probing the values of all of the weights and then loading those weight values into a recording system. Thus, the capability of rapidly copying a multiplexed volume interconnection hologram may be important (Jenkins and Tanguay, 1995, and references therein, especially to Piazzolla et al.).

Most optical and photonic neural network architectures (or “modules”) of this class can be generalized in the following additional ways. First, multiple modules can be cascaded with fully parallel communication. Second, multiple neural network layers can be implemented either by adding feedback capability to the single module described or by cascading multiple modules. And finally, smart pixel SLMs can be employed to implement various neuron-unit models.

Discussion

Research to date on optical and photonic neural network implementations has included the development and analysis of new architectures, analysis of scalability issues, development of the technology base for near-optimal individual components, and experimental proof-of-concept demonstrations. Additional research and development directions that are key for the eventual realization of physically small, high-performance, reasonable-cost photonic neural networks include increased focus on the manufacturability of the photonic and optical components, development of packaging and miniaturization techniques for hybrid electronic/photonic systems, and increased design automation and flexibility. Substantial progress has already been made along these lines. Significant leverage is also provided by rapid advances in the related areas of photonic digital interconnection systems, digital volume holographic memory systems, and components for early vision.

As the technology base evolves, photonic neural network implementations inspired by more sophisticated neuron, synapse, dendrite, axon, and network models should become feasible (e.g., models that include temporal correlations or dynamic synapses; see DYNAMIC LINK ARCHITECTURE; SELF-ORGANIZATION AND THE BRAIN; and SYNCHRONIZATION, BINDING AND EXPECTANCY). Furthermore, the eventual implementation of photonic neural networks will likely engender a concomitant development of application areas for large-scale networks, and a deeper understanding of their properties.

Road Map: Implementation and Analysis

Related Reading: Analog VLSI Implementations of Neural Networks; Digital VLSI for Neural Networks

References

- Bond, S.W., Vendier, O., Myunghee, L., Jung, S., Vrazel, M., Lopez-Lagunas, A., Chai, S., Dagnall, G., Brooke, M., Jokerst, N. M., Wills, D. S., and Brown, A., 1999, A three-layer 3-D silicon system using through-SI vertical optical interconnections and SI CMOS hybrid building blocks, *IEEE J. Selected Topics in Quantum Electronics*, 5:276–286.
- Dowling, J. E., 1992, *Neurons and Networks: An Introduction to Neuroscience*, Cambridge, MA: Belknap Press/Harvard University Press. ♦
- Fey, D., Erhard, W., Gruber, M., Jahns, J., Bartelt, H., Grimm, G., Hoppe, L., and Sinzinger, S., 2000, Optical interconnects for neural and reconfigurable VLSI architectures, *Proc. IEEE*, 88:838–848.
- Hubel, D. H., 1988, *Eye, Brain, and Vision*, New York: Freeman. ♦
- Jenkins, B. K., and Tanguay, A. R., Jr., 1992, Photonic implementations of neural networks, in *Neural Networks for Signal Processing* (B. Kosko, Ed.), Englewood Cliffs, NJ: Prentice Hall, pp. 287–382. ♦
- Jenkins, B. K., and Tanguay, A. R., Jr., 1995, Optical architectures for neural network implementations, and optical components for neural network implementations, in *Handbook of Brain Theory and Neural Net-*

- works, 1st ed. (M. Arbib, Ed.), Cambridge, MA: MIT Press, pp. 673–682. ♦
- Jutamulia, S., Ed., 1994, *Selected Papers on Optical Neural Networks*, Bellingham, WA: SPIE Press. ♦
- Li, Y., Tanida, J., Tooley, F., and Wagner, K., Eds., 1996, *Optical Computing* (special issue), *Appl. Opt.*, 35:1177–1380. ♦
- Mead, C., 1989, *Analog VLSI and Neural Systems*, Reading, MA: Addison-Wesley. ♦
- Petrisor, G. C., Goldstein, A. A., Jenkins, B. K., Herbulock, E. J., and Tanguay, A. R., Jr., 1996, Convergence of backward-error-propagation learning in photorefractive crystals, *Appl. Opt.*, 35:1328–1343.
- Pribram, K. H., 1991, *Brain and Perception: Holonomy and Structure in Figural Processing*, Hillsdale, NJ: Erlbaum.
- Tanguay, A. R., Jr., and Jenkins, B. K., 2002, Hybrid electronic/photonic multichip modules for vision and neural prosthetic applications, in *Toward Replacement Parts for the Brain: Implantable Biomimetic Electronics as the Next Era in Neural Prosthetics* (T. W. Berger and D. L. Glanzman, Eds.), Cambridge, MA: MIT Press (in press). ♦
- Tanguay, A. R., Jr., Jenkins, B. K., von der Malsburg, C., Mel, B., Holt, G., O'Brien, J., Biederman, I., Madhukar, A., Nasiatka, P., and Huang, Y., 2000, Vertically integrated photonic multichip module architecture for vision applications, in *Optics in Computing 2000* (R.A. Lessard and T. Galstian, Eds.), *Proc. SPIE*, 4089:584–600. ♦
- Veldkamp, W. B., 1993, Wireless focal planes “on the road to amacronic sensors,” *IEEE J. Quant. Electron.*, 29:801–813. ♦
- Wagner, K., and Psaltis, D., 1993, Eds., *Optical Implementations of Neural Networks* (special issue), *Appl. Opt.*, 32:1249–1476. ♦
- Wandell, B. A., 1995, *Foundations of Vision*, Sunderland, MA: Sinauer. ♦
- Worchesky, T. L., Ritter, K. J., Martin, R., and Lane, B., 1996, Large arrays of spatial light modulators hybridized to silicon integrated circuits, *Appl. Opt.*, 35:1180–1186.

Population Codes

Alexandre Pouget and Peter E. Latham

Introduction

Many sensory and motor variables in the brain are encoded by coarse codes, i.e., by the activity of large populations of neurons with broad tuning curves. For example, the direction of visual motion is believed to be encoded in the medial temporal (MT) visual area by a population of cells with bell-shaped tuning to direction, as illustrated in Figure 1A. Other examples of variables encoded by populations include the orientation of a line, the contrast in a visual scene, the frequency of a tone, and the direction of intended movement in motor cortex. These encodings extend to two dimensions—a single set of neurons might contain information about both orientation and contrast—or more.

Population codes are computationally appealing for at least two reasons. First, the overlap among the tuning curves allows precise encoding of values that fall between the peaks of two adjacent tuning curves (Figure 1A). Second, bell-shaped tuning curves provide basis functions that can be combined to approximate a wide variety of nonlinear mappings. This means that many cortical functions, such as sensorimotor transformations, can be easily modeled with population codes (see Pouget, Zemel, and Dayan, 2000, for a review).

In this article we focus on decoding, or reading out, population codes. Decoding is the simplest form of computation that one can perform over a population code, and as such, it is an essential step

toward understanding more sophisticated computations. It is also important for accurately identifying which variables are encoded in a particular brain area and how they are encoded.

A key element of population codes—and the main reason why decoding them is difficult—is that neuronal responses are noisy, meaning that the same stimulus can produce different responses. Consider, for instance, a population of neurons coding for a one-dimensional parameter: the direction, θ , of a moving object. An object moving in a particular direction produces a *noisy* hill of activity across this neuronal population (Figure 1C). On the basis of this noisy activity, one can try to come up with a good guess, or estimate, $\hat{\theta}$, of the direction of motion, θ . In the second and third sections of this article we review the various estimators that have been proposed, and in the fourth section we consider their neuronal implementations.

Additional sources of uncertainty, beside neuronal noise, can come from the variable itself. For example, there is intrinsically more variability in one's estimate of, say, motion on a dark night than motion in broad daylight. In cases such as this, it is not unreasonable to assume that population activity codes for more than just a single value, and in the extreme case the population activity could code for a whole *probability distribution*. The goal of decoding is then to recover an estimate of this probability distribution. We consider an example of this later in the article.

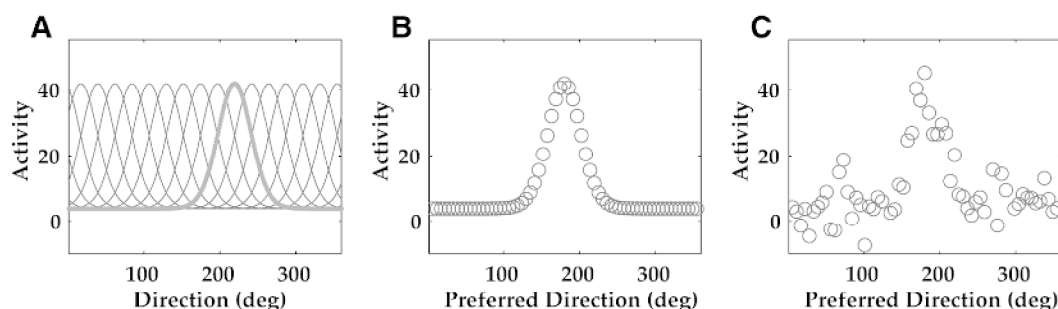


Figure 1. A, Idealized tuning curves for 16 direction-tuned neurons. B, Noiseless pattern of activity (○) from 64 simulated neurons with tuning curves like the ones shown in A, when presented with a direction of 180°.

The activity of each neuron is plotted at the location of its preferred direction. C, Same as B, but in the presence of Gaussian noise.

Models of Neuronal Noise and Tuning Curves

To read a population code, it is essential to have a good understanding of the relation between the patterns of activity and the encoded variables. One common assumption, particularly in sensory and motor cortex, is that patterns of activity encode a single value per variable at any given time. This is a reasonable assumption in many situations (although there are exceptions, as discussed later). For example, an object can move in only one direction at a time, so the neurons encoding its direction of motion have only one value to encode.

Under the assumption of a single value, neuronal responses are generally characterized by tuning curves, noted $f_i(\theta)$, which specify the mean activity of cell i as a function of the encoded variable. These tuning curves are typically bell shaped, and are often taken to be Gaussian for nonperiodic variables and circular normal for periodic ones.

Simply measuring the mean activity, however, is not sufficient for performing estimation. A neuron may fire at a rate of 20 spikes/s on one trial but only 15 spikes/s on the next, even though the same stimulus was presented both times. This trial-to-trial variability is captured by the noise distribution, $P(a_i = a|\theta)$, where a_i is the activity of cell i . The noise distribution is often assumed to be Gaussian, either with fixed variance or with a variance proportional to the mean (the latter being more consistent with experimental data), and independent. Such a distribution has the form

$$P(a_i = a|\theta) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(a - f_i(\theta))^2}{2\sigma_i^2}\right) \quad (1)$$

where σ_i^2 is either fixed or equal to the mean, $f_i(\theta)$. Another popular choice, especially useful if one is counting spikes, is the Poisson distribution:

$$P(a_i = k|\theta) = \frac{f_i(\theta)^k e^{-f_i(\theta)}}{k!} \quad (2)$$

Figure 1C shows a typical pattern of activity with Gaussian noise and σ_i^2 fixed.

Estimating a Single Value

We now consider various approaches to reading out a population code under the assumptions that (1) a single value is encoded at any given time, and (2) the only source of uncertainty is the neuronal noise. Most of these methods, known as estimators, seek to recover an estimate, $\hat{\theta}$, of the encoded variable. We first discuss how one assesses the quality of an estimator in general; we then provide descriptions of common estimators used for decoding population activity.

Fisher Information

An estimate, $\hat{\theta}$, is obtained by computing a function of the observed activity \mathbf{A} , where $\mathbf{A} \equiv (a_1, a_2, \dots)$. Because of neuronal noise, \mathbf{A} is a random variable and thus so is $\hat{\theta}$. This means that $\hat{\theta}$ will vary from trial to trial even for identical presentation angles. The best estimators are ones that are unbiased and efficient. An unbiased estimator is right on average: the conditional mean, $E[\hat{\theta}|\theta]$, is equal to the encoded direction, θ , where E denotes an average over trials. An efficient estimator, on the other hand, is consistent from trial to trial: the conditional variance, $E[(\hat{\theta} - \theta)^2|\theta]$, is minimal.

In general, the quality of an estimator depends on a compromise between the bias and the conditional variance. In this chapter, however, we consider unbiased estimators only, for which the conditional variance is the important measure because it fully determines how well one can discriminate small changes in the encoded variable based on observation of the neuronal activity. There exists a

theoretical lower bound on the conditional variance, which is known as the Cramér-Rao bound. For an unbiased estimator, this bound is equal to the inverse of the Fisher information (Paradiso, 1988; Seung and Sompolinsky, 1993), which leads to the inequality

$$E[(\hat{\theta} - \theta)^2|\theta] \geq \frac{1}{I_{\text{Fisher}}}$$

where

$$I_{\text{Fisher}} \equiv E\left[-\frac{\partial^2}{\partial \theta^2} \log P(\mathbf{A}|\theta)\right]$$

An efficient estimator is one whose conditional variance is equal to the Cramér-Rao bound, $1/I_{\text{Fisher}}$. When $P(\mathbf{A}|\theta)$ is known, it is often straightforward to compute I_{Fisher} . For example, for the Gaussian distribution given in Equation 1,

$$I_{\text{Fisher}} = \sum_{i=1}^N \frac{f_i'(\theta)^2}{\sigma_i^2}$$

and for the Poisson distribution given in Equation 2,

$$I_{\text{Fisher}} = \sum_{i=1}^N \frac{f_i'(\theta)^2}{f_i(\theta)}$$

(Seung and Sompolinsky, 1993).

In both of these expressions, the neurons that contribute most strongly to the Fisher information are those with a large slope (large $f_i'(\theta)$). Therefore, the most active neurons are not the most informative ones. In fact, they are the *least* informative: the most active neurons correspond to the top of the tuning curve, where the slope is zero, so these neurons make no contribution to Fisher information.

Voting Methods

Several estimators rely on the idea of interpreting the activity of a cell, normalized or not, as a vote for the preferred direction of the cell. For instance, the optimal linear estimator is given by

$$\hat{\theta}_{\text{OLE}} = \sum_{i=1}^N \theta_i a_i$$

where θ_i is the preferred direction of cell i , that is, the peak of the function $f_i(\theta)$. A variation on this theme is the center of mass estimator, defined as

$$\hat{\theta}_{\text{COM}} = \frac{\sum_{i=1}^N \theta_i (a_i - \gamma)}{\sum_{i=1}^N (a_i - \gamma)}$$

where γ is the spontaneous activity of the cells.

A third variation is known as a population vector estimator (Figure 2A). This has been extensively used for estimating periodic variables, such as direction, from real data (Georgopoulos et al., 1982). It is equivalent to fitting a cosine function through the pattern of activity and using the phase of the cosine as the estimate of direction:

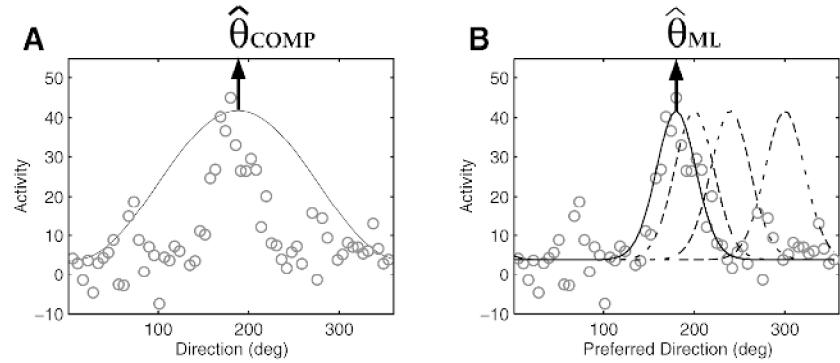
$$\hat{\theta}_{\text{COMP}} = \text{phase}(z)$$

where

$$z = \sum_{j=1}^N a_j e^{i\theta_j}$$

The first two methods work best for nonperiodic variables; the third one can only be used when the variables are periodic. All

Figure 2. *A*, The population vector estimator uses the phase of the first Fourier component of the input pattern (solid line) as an estimate of direction. It is equivalent to fitting a cosine function to the input. *B*, The maximum likelihood estimate is found by moving an “expected” hill of activity (dashed line) until the squared distance with the data is minimized (solid line).



three estimators are subject to biases, although careful tuning of the parameters can often correct for them. More important, all three methods are almost always suboptimal (the variance of the estimator exceeds the Cramér-Rao bound). The exceptions occur for a very specific set of tuning curves and noise distributions (Salinas and Abbott, 1994): the center of mass is optimal only with Gaussian tuning curves and Poisson noise, and the population vector is optimal only for cosine tuning curves and Gaussian noise of fixed variance.

Maximum Likelihood

A better choice than the voting methods, at least from the point of view of statistical efficiency, is the maximum likelihood (ML) estimator

$$\hat{\theta}_{ML} = \arg \max_{\theta} P(\mathbf{A}|\theta)$$

When there are a large number of neurons, this estimator is unbiased and its variance is equal to the Cramér-Rao bound for a wide variety of tuning curve profiles and noise distribution (Paradiso, 1988; Seung and Sompolinsky, 1993). The term maximum likelihood comes from the fact that $\hat{\theta}_{ML}$ is obtained by choosing the value of θ that maximizes the conditional probability of the activity, $P(\mathbf{A}|\theta)$, also known as the likelihood of θ .

Finding the ML estimate reduces to template matching (Paradiso, 1988), i.e., finding the noise-free hill that is closest to the activity, as illustrated in Figure 2B. If the noise is independent and Gaussian, then “closest” is with respect to the Euclidean norm, $\sum_i (a_i - f_i(\theta))^2$. For other distributions the norm is more complicated. Template matching involves a nonlinear regression, which is typically performed by moving the position of the hill until the distance from the data is minimized, as shown in Figure 2B. The position of the peak of the final hill corresponds to the ML estimate.

The main difference between the population vector and the ML estimator is the shape of the template being matched to the data. Whereas the population vector matches a cosine, the ML estimator uses a template that is directly derived from the tuning curves of the neurons that generated the activity (Figures 2A and 2B). (When all neurons have identical tuning curves, as for our examples, the template has the same profile as the tuning curves.) It is because the ML estimator uses the correct template that its variance reaches the Cramér-Rao bound. There is, however, a cost: one needs to know the profile of all tuning curves to use ML estimation, whereas only the preferred directions, θ_i , are needed for the population vector estimator.

Bayesian Approach

An alternative to ML estimation is to use the full posterior distribution of the encoded variable, $P(\theta|\mathbf{A})$. This is related to the distribution of the noise, $P(\mathbf{A}|\theta)$, through Bayes’s theorem:

$$P(\theta|\mathbf{A}) = \frac{P(\mathbf{A}|\theta)P(\theta)}{P(\mathbf{A})}$$

where $P(\mathbf{A})$ and $P(\theta)$ are the prior distributions over \mathbf{A} and θ . The value that maximizes $P(\theta|\mathbf{A})$ can then be used as an estimate of θ . This is known as a maximum a posteriori estimate, or MAP estimate. The main advantage of the MAP estimate over the ML estimate is that prior knowledge about the encoded variable can be taken into account. This is particularly important when the conditional distribution, $P(\mathbf{A}|\theta)$, is not sharply peaked compared to the prior, $P(\theta)$. This happens, for example, when only a small number of neurons are available, or when one observes only a few spikes per neuron. The MAP estimate is close to the ML estimate if the prior distribution varies slowly compared to the conditional, and the two are exactly equal when the prior is flat. Several authors have explored and/or applied this approach to real data (Foldiak, 1993; Sanger, 1996; Zhang et al., 1998).

Neuronal Implementations

Methods such as the voting schemes or ML estimator are biologically implausible, for one simple reason: they extract a single value, the estimate of the encoded variable. Such explicit decoding is very rare in the brain. Instead, most cortical areas and subcortical structures use population codes to encode variables. This means that, throughout the brain, population codes are mapped into population codes. Hence, V1 neurons, which are broadly tuned to the direction of motion, project to MT neurons, which are also broadly tuned, but in neither area is the direction of motion read out as a single number. The neurons in MT are nevertheless confronted with an estimation problem: they must choose their activity levels on the basis of the noisy activity of V1 neurons.

What is the optimal strategy for mapping one population code into another? We cannot answer this question in general, but we can address it for the broad class of networks depicted in Figure 3. In these networks, the input layer is a set of neurons with wide tuning curves, generating noisy patterns of activity like the one shown in Figure 1C. This activity, which acts transiently, is relayed to an output layer through feedforward connections. In the output layer the neurons are connected through lateral connections.

An update rule (discussed later) causes the activity in the output layer to evolve in time. In the next section we consider networks in which the update rule leads to a smooth hill. The peak of that hill can be interpreted as an estimate of the variable being encoded. As previously, we can assess how well the network did by looking at the mean and variance of this estimate.

We will consider two kinds of networks: those with a linear activation function and those with a nonlinear one.

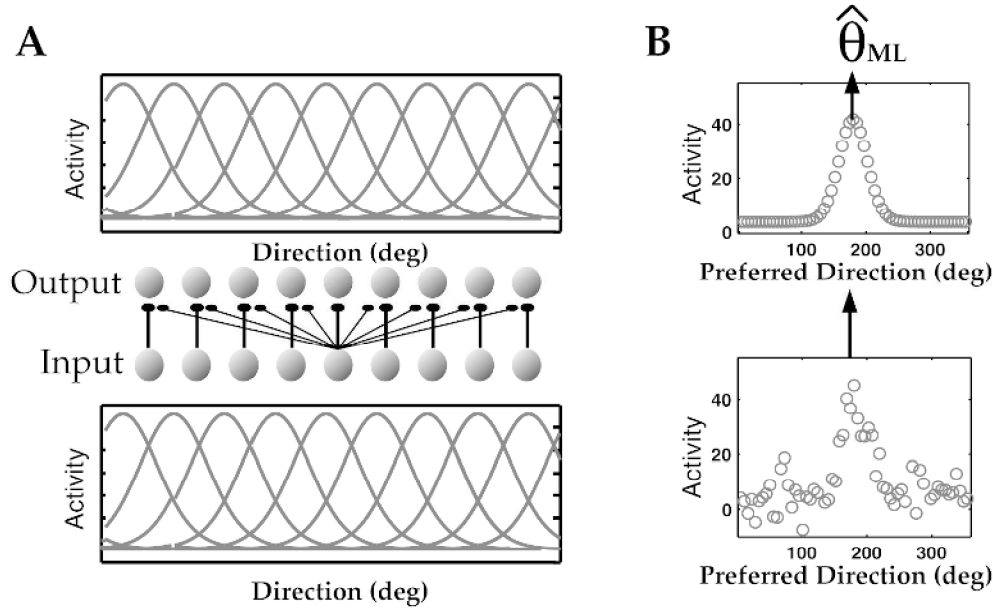


Figure 3. *A*, A set of units with broad tuning to a sensory variable (in this case direction) projects to another set of units also broadly tuned to the same variable. This type of mapping between population codes is very common throughout the brain. In this particular network, the output layer is fully interconnected with lateral connections, and receives feedforward connections from the input layer. *B*, Temporal evolution of the activity in the output layer for a nonlinear network. The activity in the output layer is

initiated with a noisy hill generated by the input units (bottom). For an appropriate choice of weights and activation function, these activities converge eventually to a smooth hill (top), which peaks close to the location of the maximum likelihood estimate of direction, $\hat{\theta}_{ML}$. This network is performing the template-matching procedure used in maximum likelihood and illustrated in Figure 2*B*.

Linear Networks

We first consider a network with linear activation functions in the output layer, so that the dynamics is governed by the difference equation

$$\mathbf{O}_t = ((1 - \lambda)\mathbf{I} + \lambda\mathbf{W})\mathbf{O}_{t-1} \quad (3)$$

where λ is a number between 0 and 1, \mathbf{I} is the identity matrix, and \mathbf{W} is the matrix for the lateral connections. The activity at time 0, \mathbf{O}_0 , is initialized to $\mathbf{W}\mathbf{A}$, where \mathbf{A} is an input pattern (like the one shown in Figure 1*C*) and \mathbf{W} is the feedforward weight matrix (for simplicity, the feedforward and lateral weights are the same, although this is not necessary).

The dynamics of such a network is well understood: each eigenvector of the matrix $(1 - \lambda)\mathbf{I} + \lambda\mathbf{W}$ evolves independently, with exponential amplification for eigenvalues greater than 1 and exponential suppression for eigenvalues less than 1. When the weights are translation invariant ($W_{ij} = W_{i-j}$), the eigenvectors are sines and cosine. In this case the network amplifies or suppresses independently each Fourier component of the initial input pattern, \mathbf{A} , by a factor equal to the corresponding eigenvalue of $(1 - \lambda)\mathbf{I} + \lambda\mathbf{W}$. For example, if the first eigenvalue of $(1 - \lambda)\mathbf{I} + \lambda\mathbf{W}$ is more than 1 (respectively less than 1), the first Fourier component of the initial pattern of activity will be amplified (respectively suppressed). Thus, \mathbf{W} can be chosen such that the network amplifies selectively the first Fourier component of the data while suppressing the others.

As formulated, the activity in such a network would grow forever. However, if we stop after a large yet fixed number of iterations, the activity pattern will look like a cosine function of direction with a phase corresponding to the phase of the first Fourier

component of the data. The peak of the cosine provides the estimate of direction. That estimate turns out to be the same as the one provided by the population vector discussed above.

The unchecked exponential growth of a purely linear network can be alleviated by adding a nonlinear term to act as gain control. This type of network was proposed by Ben-Yishai, Bar-Or, and Sompolinsky (1995) as a model of orientation selectivity.

Although such networks keep the estimate in a coarse code format, they suffer from two problems: it is not immediately clear how to extend them to periodic variables, such as disparity, and they are suboptimal, since they are equivalent to the population estimator.

Nonlinear Networks

To obtain optimal performance, one needs a network that can implement template matching with the correct template—the one used by the ML estimator (see Figure 2*B*). This requires templates that go beyond cosines to include curves that are consistent with the tuning curves of the input units (see Figure 2*B*).

Nonlinear networks that admit line attractors have this property (Deneve, Latham, and Pouget, 1999). In such networks, the line attractors correspond to smooth hills of activity, with profiles determined by the patterns of weights and the activation functions. For a given activation function, it is therefore possible to select the weights to optimize the profile of the stable state. Pouget et al. (1998) demonstrated that this extra flexibility allows these networks to act as ML estimators (see Figure 3*B*).

More recent work by Deneve et al. (1999) has shown that the ML property is preserved for a wide range of nonlinear activation functions. In particular, this is true for networks using divisive normalization, a nonlinearity believed to exist in cortical micro-

circuitry. It is therefore possible that all cortical layers are close approximations to ML estimators.

Estimating a Probability Distribution

So far we have reviewed decoding methods in which only one value is encoded at any given time and the only source of uncertainty comes from the neuronal activity. Situations exist, however, in which either (or both) of these assumptions are violated. For instance, imagine that you are lost in Manhattan on a foggy day, but you can see, faintly, the Empire State building and the Chrysler building in the distance. Because of the poor visibility, the views of these landmarks are not sufficient to specify your exact position, but they are enough to provide a rough idea of where you are (Harlem versus Little Italy). In this situation, it would be desirable to compute the probability distribution of your location given that you are seeing the landmarks; i.e., compute $P(\theta|w)$ where θ is the position (now a two-dimensional vector) in Manhattan and w represents the views of the buildings. Here, the uncertainty about θ comes from the fact that you do not have enough information to tell precisely where you are. In such a situation, the neurons could encode the *probability distribution*, $P(\theta|w)$.

Because the encoded entity is a probability distribution rather than a single value, we can no longer use either Equation 1 or Equation 2 as a model for the responses of the neurons; these equations provide only the likelihood of θ , $P(A|\theta)$. What we need instead is a model that specifies the likelihood of the whole encoded probability distribution, $P[A|P(\theta|w)]$. Note that $P(\theta|w)$ plays the same role as θ previously, which is to be expected, now that $P(\theta|w)$ is the encoded entity. It is beyond the scope of this discussion to provide equations for such models, but examples can be found in Zemel, Dayan, and Pouget (1998).

Since A is now a code for the probability distribution, the relevant quantity to estimate is $P(\theta|w)$, which we denote $\hat{P}(\theta|w)$. This is still within the realm of estimation theory, so we can use the same tools that we used for the simpler case, such as ML decoding (see Zemel et al., 1998).

To see the difference between encoding a single value and encoding a probability distribution, it is helpful to consider what happens when the neurons are deterministic—that is, when the neuronal noise goes to zero. In this case, the encoded variable can be recovered with infinite precision, since the only source of uncertainty, the neuronal noise, is gone. Thus the ML estimate would be exactly equal to the encoded value, and the posterior distribution, $P(\theta|A)$, would be a Dirac function centered at θ . If the activity encodes a probability distribution, on the other hand, one would recover the *distribution* with infinite precision. However, the uncertainty about θ may still be quite large (as was the case in our Manhattan example), potentially far from a Dirac function.

It is too early to tell whether neurons encode probability distributions; more empirical as well as theoretical work is needed. But if the cortex has the ability to represent probability distributions, it might be possible to determine how, and whether, the brain performs Bayesian inferences. Bayesian inference is a powerful method for performing computation in the presence of uncertainty. Many engineering applications rely on this framework to perform data analysis or to control robots, and several studies are now suggesting that the brain might be using such inferences for perception and motor control (see, e.g., Knill and Richards, 1996).

Conclusions

Understanding how to decode patterns of neuronal activity is a critical step toward developing theories of representation and computation in the brain. This article concentrated on the simplest case, a single variable encoded in the firing rates of a population of neurons. There are two main approaches to this problem. In the first, the population encodes a single value, and decoding can be done with Bayesian or maximum likelihood estimators. The underlying assumption in this case is that neuronal noise is the only source of uncertainty. We also saw that within this framework, one can design neural networks that perform decoding optimally. In the second approach, the population encodes a full probability distribution over the variable of interest. Here both the variable and its uncertainty can be extracted from the population activity. This scheme could be used to perform statistical inferences—a powerful way to perform computations over variables whose value is not known with certainty. The challenge for future work will be to determine whether the brain uses this type of code, and, if so, to understand how realistic neural circuits can perform statistical inferences over probability distributions.

Road Map: Neural Coding

Related Reading: Cortical Population Dynamics and Psychophysics; Motor Cortex: Coding and Decoding of Directional Operations

References

- Ben-Yishai, R., Bar-Or, R. L., and Sompolinsky, H., 1995, Theory of orientation tuning in visual cortex, *Proc. Natl. Acad. Sci. USA*, 92:3844–3848.
- Deneve, S., Latham, P. E., and Pouget, A., 1999, Reading population codes: A neural implementation of ideal observers, *Nature Neurosci.*, 2:740–745. ♦
- Foldiak, P., 1993, The “ideal homunculus”: Statistical inference from neural population responses, in *Computation and Neural Systems* (F. H. Eeckman and J. M. Bower, Eds.), Norwell, MA: Kluwer Academic, pp. 55–60. ♦
- Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., and Massey, J. T., 1982, On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex, *J. Neurosci.*, 2:1527–1537.
- Knill, D. C., and Richards, W., 1996, *Perception as Bayesian Inference*, New York: Cambridge University Press.
- Paradiso, M. A., 1988, A theory of the use of visual orientation information which exploits the columnar structure of striate cortex, *Biol. Cybern.* 58:35–49. ♦
- Pouget, A., Zhang, K., Deneve, S., and Latham, P., 1998, Statistically efficient estimation using population codes, *Neural computation*, 10:373–401.
- Pouget, A., Zemel, R. S., and Dayan, P., 2000, Information processing with population codes, *Nature Rev. Neurosci.*, 1:125–132.
- Salinas, E., and Abbott, L. F., 1994, Vector reconstruction from firing rate, *J. Computat. Neurosci.*, 1:89–107. ♦
- Sanger, T. D., 1996, Probability density estimation for the interpretation of neural population codes, *J. Neurophysiol.*, 76:2790–2793.
- Seung, H. S., and Sompolinsky, H., 1993, Simple model for reading neuronal population codes, *Proc. Natl. Acad. Sci. USA*, 90:10749–10753.
- Zemel, R. S., Dayan, P., and Pouget, A., 1998, Probabilistic interpretation of population code, *Neural Computat.*, 10:403–430. ♦
- Zhang, K., Ginzburg, I., McNaughton, B. L., and Sejnowski, T. J., 1998, Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells, *J. Neurophysiol.*, 79:1017–1044.

Post-Hebbian Learning Algorithms

Péter Érdi and Zoltán Somogyvári

Post-Hebbian Learning Rules: A Retrospective

Hebb's introduction of his learning rule inaugurated a new era and resulted in the appearance of many new branches of theory and new models of learning. Two characteristics of the original postulate (Hebb, 1949) played key roles in the subsequent development of post-Hebbian learning rules. First, despite being biologically motivated, Hebb's learning rule was a verbally described, *phenomenological* rule, unlinked to detailed physiological mechanisms. Second, because the learning rule was extremely convincing, it was widely adopted both as a theoretical framework and as a formal tool in the field of neural networks.

As a result, the etiolation of Hebb's idea occurred in two principal directions. First, the postulate inspired an intense and long-lasting search for the molecular and cellular basis of learning phenomena, which were assumed to be Hebbian; thus, this particular development has been absorbed by neurobiology. Second, because of its computational usefulness, many variations of the *biologically inspired* learning rules appeared and were applied to a huge number of very different problems in artificial neural networks, without any relation to a biological foundation being claimed. Several families of rules sprouted from the original idea. Before discussing them, we should note that what we consider a Hebbian, post-Hebbian, or non-Hebbian learning rule to be is subjective and time varying.

This *Handbook* includes a broad overview of the subject in the article HEBBIAN SYNAPTIC PLASTICITY (q.v.). Here we focus predominantly on computational implementations of Hebbian rules. First, however, we discuss the different roots and new developments related to Hebb's hypothesis, such as psychologically motivated conditioning, neural development, and physiologically realistic cellular level learning phenomena. Thereafter families of formal Hebbian learning, or *algorithms*, are considered.

Variations on the Hebbian Theme: Motivations

Conditioning

Since the end of the nineteenth century, physiologists have known that mature nerve cells cannot divide (though in the late 20th century they changed their minds!). Thus, because learning could not result from the proliferation of new neurons, the locus of learning had to be *the connections between cells*. Such phenomena are related to the neural basis of classical conditioning. The first attempt to model conditioning in terms of synaptic change was by Hebb.

Hebb's original intent was to connect the behavior of whole organisms to neural mechanisms by using concepts represented by cell assemblies. Specifically, classical conditioning involves the development of an association between two otherwise unrelated events over a number of trials in which the events are temporally paired. Typically, a neutral stimulus, one that does not naturally provoke behavior, is presented, immediately followed by an unconditioned stimulus, or an event that does not require training to produce a response, resulting in the eliciting of an unconditioned response.

Classical conditioning (see *CONDITIONING*) has been described by Rescorla and Wagner's (1972) model. They proposed a formal model of conditioning that expresses the capacity of a conditioned stimulus (CS) to become associated with an unconditioned stimulus (US) at any given time. The fulcrum of the Wagner-Rescorla model is that learning occurs if and when events violate expectations. More specifically, learning occurs whenever the actual level of US received during a trial differs from the level expected. The Res-

corla-Wagner rule can be interpreted as saying that the discrepancy between expected and actual values determines the measure of reinforcement. So, the rule and its many later modifications act over the unsupervised learning paradigm. One drawback of the Rescorla-Wagner model, however, is that it completely ignores the temporal sequence in which information is presented.

Development

The formation and refinement of neural circuits involves both the establishment of new connections and the elimination of already existing connections. The leading mechanism in synaptic elimination is considered to be *axonal* or *synaptic competition*. Neuro-muscular junctions and the visual system are the two best investigated examples in which synaptic competition plays an important role.

A large variety of generalized Hebbian learning rules have been applied to neural development (see review by van Ooyen, 2001). An example is the different mechanisms of competition elaborated in the context of population biology that have been adopted in the neural context. In *consumptive competition* in systems of consumers and resources (e.g., predators and prey), each individual consumer tries to avoid other consumers and hinders other consumers solely by consuming resources that they might have consumed. In other words, consumers hinder each other because they draw on the same resources. In neurobiology, competition is commonly associated with this dependence on shared resources. In *interference competition*, instead of hindrance through dependence on shared resources, there is direct interference between individuals. Such interference occurs, for example, if there are direct negative interactions (e.g., aggressive or toxic interactions) between individuals. In axonal competition, nerve terminals could seek to destroy each other by releasing proteases.

Long-Term Potentiation—Long-Term Depression

Long-term potentiation (LTP) was first discovered in the hippocampus and is very prominent there. LTP is an increase in synaptic strength that can be rapidly induced by brief periods of synaptic stimulation. It has been reported to last for hours in vitro, and for days to weeks in vivo.

LTP (and later long-term depression, or LTD) became regarded as the physiological basis of Hebbian learning. Subsequently, the properties of LTP and LTD became clearer, and the question then arose as to whether LTP and LTD could really be considered the microscopic basis of the phenomenological Hebbian type of learning. Formally, the question is how to specify the general function F to serve as a learning rule with the known properties of LTP and LTD. Recognizing the existence of this gap between biological mechanisms and the long-used Hebbian learning rule, many workers have attempted to derive the corresponding phenomenological rule based on more or less detailed neurochemical mechanisms.

The time course of LTP may be insufficient to sustain long-term memory, but there appear to be multiple LTP mechanisms, and one dependent on protein synthesis might serve long-term memory: inhibition of protein synthesis disrupts the maintenance of LTP but leaves the induction of LTP relatively or totally intact. It is possible to relate the properties and mechanisms of long-term synaptic plasticity in the mammalian brain to learning and memory.

An example of the new synaptic bidirectional Hebbian rules was introduced by Grzywacz and Buzsáki in 1998. When this rule was

compared with physiological homosynaptic conditions in the hippocampus, the results indicated that this rule was consistent with LTP and LTD phenomenologies. The phenomenologies considered included the reversible dynamics of LTP and LTD and the effects of *N*-methyl-D-aspartate (NMDA) blockers and phosphatase inhibitors.

Timing

Studies in cortical and hippocampal slices have shown that back-propagating action potentials may contribute to the induction of persistent synaptic potentiation or depression. The timing of presynaptic and postsynaptic action potentials plays a decisive role in determining the sign of synaptic modification (Markram et al., 1997). The temporal order of the synaptic input and the postsynaptic spike within a narrow window of time determines whether LTP or LTD is elicited, according to a temporally asymmetric Hebbian learning rule.

Bi and Poo (1998) showed that postsynaptic spiking that peaked within 20 ms *after* synaptic activation resulted in LTP, whereas spiking within 20 ms *before* synaptic activation led to LTD. They suggested that a narrow and asymmetric window for the induction of synaptic modification should be taken into account.

Most generalized Hebbian rules are based on the statistical properties of presynaptic and postsynaptic activity (e.g., activity product, activity covariance) and do not consider the detailed temporal structure of the spike patterns. Relative spike timing, however, had been taken into account as early as 1981 by Sutton and Barto.

Since changes in synaptic efficacy can depend on the precise temporal relations of pre- and postsynaptic spikes, phenomenological “temporal learning rules” generate opposite changes in synaptic efficiency, depending on whether the postsynaptic spike occurs before or after the presynaptic spike. Roberts (1999) attempted to show that differential Hebbian learning could take into account the timing effects.

Generalized Hebbian Rules and Their Phenomenological Derivations

Hebb’s idea has been formalized in many variations. The first and simplest versions of the Hebbian learning rule have the important properties of being *local* and *interactive* (specifically, *conjunctive* and *time dependent*), as we will now explain. We will consider what happens when we attempt to preserve these properties in the course of generalizing the Hebbian learning rule.

The most general form of Hebb’s rule is that the synaptic weight from neuron *i* to neuron *j* changes according to

$$\frac{d}{dt} w_{ij}(t) = F(a_i, a_j) \quad (1)$$

where *F* is a functional, and *a_i* and *a_j* are presynaptic and postsynaptic activity functions (i.e., they may include activity levels over some period of time and not just the current activity values). To define specific learning rules (i.e., the form of *F*), a few points should be clarified.

1. What are the assumptions about the *locality* of the modifying signal? In many cases the modification of a synapse between neurons *i* and *j* depends on the state of these two cells alone; i.e., the mechanism is local. In this case, teacher or external reinforcement signals are not explicitly involved: local synapses are the bases of the unsupervised learning.
2. How, if at all, do the presynaptic and postsynaptic cells *interact*? Consider first the potential answers for the “if at all” part of the question. The modification can be interactive if both the pre-

and postsynaptic cells are involved, and noninteractive if either the pre- or postsynaptic cell alone influences the modification. The mechanism of the interaction may be conjunctive or correlational. In the first case, co-occurrence of pre- and postsynaptic activity is sufficient to cause synaptic change, while in the second case, covariance of the two activities must be taken into account. (From a formal point of view, additive interactions, such as those given by the function $F(a_i \pm a_j)$, could have been defined, but they are considered as noninteractive rules. In other words, not only an entire rule but each term of it can be evaluated as interactive or noninteractive.)

3. What are the assumptions about the form of the *time-dependent* activity functions? In the simplest case, only the actual activity values are involved. In somewhat more complex situations, short-term averaged activity values determine the synaptic change. More generally, the history of the activity values plays a role in the modification process.

The simplest Hebbian learning rule can be formalized as

$$\frac{d}{dt} w_{ij}(t) = k a_i(t) a_j(t), \quad k > 0 \quad (2)$$

This rule expresses the conjunction among pre- and postsynaptic elements (using neurobiological terminology) or associative conditioning (in psychological terms), by a simple product of the actual states of pre- and postsynaptic elements, *a_i(t)* and *a_j(t)*.

A characteristic and unfortunate property of the simplest Hebbian rule in Equation 1 is that the synaptic strengths are ever increasing (see HEBBIAN LEARNING AND NEURONAL REGULATION for solutions to the problems this property raises).

$$\frac{d}{dt} w_{ij}(t) = k g(a_i(t)) h(a_j(t)) \quad (3)$$

where *g* and *h*, functions of the actual activity, serve as some measure of the post- and presynaptic activity (i.e., *g*, *h* > 0), and

$$\frac{d}{dt} w_{ij}(t) = k g(a_i(\cdot)) h(a_j(\cdot)) \quad (4)$$

where *g* and *h* are now functionals of the activity function. A special case is

$$\frac{d}{dt} w_{ij}(t) = k \int_0^t a_i(t) dt \int_0^t a_j(t) dt \quad (5)$$

which takes into account the total activity history.

There is a particular time-dependent, local, and conjunctive rule that does not increase the synaptic weight. This is the case in which the pre- and postsynaptic activities are negatively correlated:

$$\frac{d}{dt} w_{ij}(t) = k a_i(t) a_j(t), \quad k < 0 \quad (6)$$

This “anti-Hebbian” rule (there is some confusion in the literature concerning this terminology; here it is used in the sense that *k* < 0) or “decorrelation” rule was suggested to describe features of dissociations of patterns (Barlow and Földiák, 1989).

There are both brutal and sophisticated methods to eliminate the unpleasant property of ever-increasing weights, which, unless compensated for, yield a network with saturated synaptic weights, and thus no effective pattern discrimination. The adjective “brutal” was adopted for the situation in which some external constraint (somehow taking into account the finiteness of resources) is applied to the internal mechanism. First, a predetermined upper bound can be given, such as the maximal value of the synaptic strength. Second, the so-called normalization procedure (described in Rochester et al., 1956) gives a finite-sum constraint on all synaptic strengths, and can be interpreted as a competition of the presynaptic elements

for postsynaptic resources (therefore, it violates locality). Such rules may explain some aspects of neural development

More sophisticated methods decrease the synaptic strengths selectively. Brown, Kairiss, and Keenan (1990) use the expression *generalized Hebbian synaptic mechanism* for cases in which interactive synaptic increase is combined with activity-dependent synaptic depression. The underlying mechanism behind synaptic depression may be of interactive or noninteractive type.

Instead of giving a formal derivation of the rules that are able to describe selective decrease, we will mention two important special cases. First, the rule

$$\frac{d}{dt} w_{ij}(t) = kg(a_i(t))(h(a_j(t)) - \theta(t)) \quad (7)$$

implements synaptic increase only if the $h(a_j(t))$ presynaptic activity is larger than the $\theta(t)$ modification threshold. If the presynaptic activity is smaller than the threshold, the synaptic weight decreases. Second,

$$\frac{d}{dt} w_{ij}(t) = k(g(a_i(t)) - \theta(t))h(a_j(t)) \quad (8)$$

implements a postsynaptic control mechanism on the modification process.

The learning rules in Equations 8 and 9 can be written in the form $kgh - k\theta g$ and $kgh - k\theta h$, respectively. Each of these expressions may be interpreted as the sum of a Hebbian interactive term and a noninteractive term. In the first case, the decrease is due to postsynaptic activity g and is called *heterosynaptic depression*, while in the second case it depends on the presynaptic activity h and is called *homosynaptic depression*. Learning rules of the form of Equation 9 were suggested by Bienenstock, Cooper, and Munro (1982) and so are sometimes referred as the BCM theory; they are used to model the plasticity of visual cortex. $\theta(t)$ was identified with a nonlinear function of the averaged postsynaptic activity:

$$\theta(t) = [g(t)]^2 \quad (9)$$

where $[\cdot]$ is the average taken for a period of time. The suggestion that the occurrence of either homosynaptic LTP or LTD depends on the strength of the depolarizing current induced by an NMDA blocker (which increases the modification threshold) in the visual cortex seemed to be justified experimentally.

The learning expression has also been described in the form $\phi(g, [g])h$, where the two-variable function ϕ depends on an actual value and an averaged quantity, so an underlying microscopic stochastic mechanism should exist behind the phenomenological and deterministic formalism.

The weaker form of the interactive rule (namely, when correlational and nonconjunctive interactions were assumed), or

$$\frac{d}{dt} w_{ij}(t) = k(a_i(t) - [a_i(t)])(a_j(t) - [a_j(t)]) \quad (10)$$

was offered by Rochester et al. (1956). Depending on the sign of the correlation, the rule is capable of describing either synaptic enhancement or decrease. Covariance was suggested to induce associative LTD in the hippocampus.

Another way to describe the decrease of synaptic weights is the introduction of a spontaneous decay (or “forgetting”) term. The original Hebbian rule (Equation 2) supplemented with a decay term reads as

$$\frac{d}{dt} w_{ij}(t) = -k_1 w_{ij}(t) + k_2 a_i(t) a_j(t) \quad (11)$$

(Instead of first-order decay, a quadratic forgetting term was also introduced and studied to improve the stability properties of the

learning rule.) If the decay is not spontaneous but modulated with the postsynaptic activity, the rule has the form

$$\begin{aligned} \frac{d}{dt} w_{ij}(t) &= -k_1 w_{ij}(t) a_j(t) + k_2 a_i(t) a_j(t) \\ &\equiv a_j(t)(k_2 a_i(t) - k_1 w_{ij}(t)) \end{aligned} \quad (12)$$

and describes the phenomenon called competitive learning. Postsynaptic neurons compete for incoming resources: the larger the postsynaptic activity, the larger the measure of learning:

$$\frac{d}{dt} w_{ij}(t) = k \frac{d}{dt} a_i(t) \frac{d}{dt} a_j(t) \quad (13)$$

This rule is an example of a differential learning mechanism (Klopf, 1986). Obviously, the rate of change of activities may be positive or negative; that is, both synaptic increase and decrease may occur. The differential competitive rule,

$$\frac{d}{dt} w_{ij}(t) = \frac{d}{dt} a_i(t)(k_2 a_j(t) - k_1 w_{ij}(t)) \quad (14)$$

implements the “learn only if change” principle.

In some cases, the time delay due to signal transmission is explicitly taken into account; consequently, earlier presynaptic activities, rather than current activities, are in conjunction:

$$\frac{d}{dt} w_{ij}(t) = k a_i(t) a_j(t - \tau) \quad (15)$$

This spirit of “timing sensitivity” is materialized in the rule

$$\frac{d}{dt} w_{ij}(t) = k_1 \frac{d}{dt} a_i(t) [a_j(t)] \quad (16)$$

used to describe conditioning (see, e.g., Sejnowski and Tesauro, 1990).

Hebbian Mechanisms and Hebbian Algorithms

Hebb proposed that the connection between two neurons will increase if activity in the neurons is temporally paired. More specifically, the Hebbian model proposes that the strength of a particular connection will increase if use of the synapse contributes to the occurrence of an action potential in the postsynaptic neuron. This account critically depends on coincidence detectors in the postsynaptic neuron.

The underlying biophysical mechanisms and algorithms of even generalized Hebbian synaptic modification were reviewed by Brown et al. (1990). Over the next several years, system-level computational models of the neural bases of learning and memory began to proliferate.

The general question has been, and still is, how to connect the formal algorithms of the neural basis of learning phenomena. Although many commonly used learning rules lead to successful models of plasticity and learning, they are inconsistent with what is known about neurophysiology. Other, more physiologically plausible rules fail to specify relevant properties, such as bidirectionality and the biological mechanism that prevents synapses from changing from excitatory to inhibitory, and vice versa. More recent attempts have tried to overcome these difficulties.

Discussion: Over the Hebbian Paradigm

It is certainly not true that all learning rules could be interpreted in even a generalized Hebbian sense. It is difficult, however, to discriminate precisely between Hebbian and non-Hebbian frameworks. One way to do so might be to consider a learning rule Hebbian if only two elements, one presynaptic, one postsynaptic,

are involved. If we accept this limitation, we know by exclusion what a non-Hebbian learning rule is. Many types of supervised learning rules used in artificial neural networks, such as delta rules, and their variations certainly belong to this category. Heterosynaptic plasticity and the modifiability of synaptic triads and glomeruli, in which more than two cells are explicitly involved in the modification process, could also be understood as non-Hebbian. Such a choice, however, would also exclude rules with the normalization procedure.

What is the relationship between homosynaptic (or Hebbian activity-dependent) and heterosynaptic (or modulatory input-dependent) plasticity? It has often been suggested (see, e.g., Bailey et al., 2000) that Hebbian mechanisms are used primarily for learning and for forming short-term memory traces, but they are not sufficient to recruit the events required to maintain a long-term memory. In contrast, heterosynaptic plasticity commonly recruits long-term memory mechanisms that lead to transcription and to synaptic growth. When jointly recruited, homosynaptic mechanisms ensure that learning is effectively established and heterosynaptic mechanisms ensure that memory is maintained.

The spirit of the Hebbian idea survived more than half a century. It will be interesting to see what kinds of phenomenological learning rules will be derived in the next several years, starting from cellular level experimental and modeling studies of synaptic modifiability.

Road Map: Neural Plasticity

Background: Hebbian Synaptic Plasticity

Related Reading: Hebbian Learning and Neuronal Regulation

References

- Bailey, C., Giustetto, M., Huang, Y., Hawkins, R., and Kandel, E., 2000, Is heterosynaptic modulation essential for stabilizing Hebbian plasticity and memory? *Nature Rev. Neurosci.*, 1:11–20.
- Barlow, H., and Földiák, P., 1989, Adaptation and decorrelation in the cortex, in *The Computing Neuron* (R. Durbin, C. Miall, and G. Mitchison, Eds.), Wokingham, Engl.: Addison-Wesley, pp. 54–72.
- Bi, G., and Poo, M., 1998, Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type, *J. Neurosci.*, 18:10464–10472.
- Bienenstock, E., Cooper, L., and Munro, P., 1982, Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex, *J. Neurosci.*, 2:32–48.
- Brown, T., Kairiss, E., and Keenan, C., 1990, Hebbian synapses: Biophysical mechanisms and algorithms, *Annu. Rev. Neurosci.*, 13:475–511. ♦
- Grzywacz, N., and Burgi, P., 1998, Toward a biophysically plausible bidirectional Hebbian rule, *Neural Computat.*, 10:499–520.
- Hebb, D., 1949, *The Organization of the Behavior*, New York: Wiley.
- Klopf, A., 1986, A drive-reinforcement model of single neuron function: An alternative to the Hebbian neuronal mode, in *Proceedings of the American Institute of Physics: Neural Networks for Computing*, New York: American Institute of Physics pp. 265–270.
- Markram, H., Lubke, J., Frotscher, M., Roth, A., and Sakmann, B., 1997, Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex, *J. Physiol.*, 500(Pt. 2):409–440.
- Rescorla, R., and Wagner, A., 1972, A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement, in *Classical Conditioning II* (A. Black and W. Prokasy, Eds.), New York: Appleton-Century-Crofts.
- Roberts, P., 1999, Computational consequences of temporally asymmetric learning rules: I. Differential Hebbian learning, *J. Comput. Neurosci.*, 7:235–246.
- Rochester, N., Holland, J., Haibt, L., and Duda, W., 1956, Tests on a cell assembly theory of the action of the brain, using a large scale digital computer, *IRE Trans. Inform. Theory*, IT-2:80–93.
- Sejnowski, T., and Tesauero, G., 1990, Building network learning algorithms for Hebbian synapses, in *Brain Organization and Memory Cells, Systems, and Circuit* (N. McGaugh, N. Weinberger, and G. Lynch, Eds.), New York: Oxford University Press. ♦
- Sutton, R., and Barto, A., 1981, Toward a modern theory of adaptive networks: Expectation and prediction, *Psychol. Rev.*, 88:135–170. ♦
- van Ooyen, A., 2001, Competition in the development of nerve connections: A review of models, *Network*, 12:R1–R47.

Potential Fields and Neural Networks

Jiming Liu and Oussama Khatib

Introduction

In this article, we review the ideas and observations involved in the concept of potential fields. *Potential fields* are often defined to characterize the vector-field output of a behavior module, the impedance of a structure, or the external constraints of an environment. Such a characterization is useful for modeling motor behavior control in humans and robots. Here we also examine how neural networks are related to potential fields, as underlying architectures for enabling sensorimotor control and behavior learning.

Biological Relevance of Potential Fields in Behavior Modeling

In biological studies, potential fields offer a way to represent and measure the motor output behavior of a physiological mechanism. Many of the reported biological studies that have taken into account potential fields have sought to understand how the neural circuits in the frog's spinal cord are organized and how the motor behaviors of the frog, such as postures, are governed by the neural control modules.

Complex Motor Behaviors and Vector Fields

The theoretical work of Mussa-Ivaldi and Giszter (e.g., 1992), known as vector-field approximation, characterizes the elicited outputs of distinct postural control modules as basis fields. Simple motor tasks can therefore be described by the specific features of vector fields, such as stable equilibrium, impedance, unstable equilibrium, saddle point, uniform field, and circulation. In their simulations, they show that a repertoire of convergent force patterns can readily be approximated based on the superposition of basis fields (as training examples). Furthermore, by combining the field features, it is possible to produce a variety of more complex control patterns.

Linear Combination of Force Fields

In relation to vector-field approximation and complex pattern generation, Mussa-Ivaldi, Giszter, and Bizzi (1994) have tested the hypothesis that vectorial superposition—that is, linear combination—of the motor primitives in the neural circuits of the frog's spinal cord can lead to a variety of motor behaviors. They conducted experiments to measure the force fields in the workspace of

the frog's leg while stimulating the premotor sites in the frog's spinal cord. Their experimental results have shown that simultaneously stimulating two regions in the spinal cord generates the vector summation of the end-point forces generated by the separate stimulation of each spinal region (see MOTOR PRIMITIVES).

Adaptive Combination of Motor Primitives

Based on results achieved in the linear combination of force fields, we might ask whether motor behavior learning can also be modeled in terms of changes in the primitive combination. Thoroughman and Shadmehr (2000) believe that a flexible combination of human motor primitives provides a model for experience-based complex movement learning. They suggest that the human brain builds a state-dependent internal model of muscle forces for generating the dynamic trajectories of movements. This internal model is essentially a sensorimotor map that is constructed by combining a set of motor primitives. The primitives have Gaussian-like turning functions for encoding arm velocities. The experience-based learning of a movement is therefore reflected in the adaptive adjustment of the weight matrix that defines the primitive combination.

Designing Force-Field Motor Primitives

Insights into the organization of motor primitives and their role in the generation of force fields not only enhance our understanding of how biological neural circuits produce motor patterns, they also provide guidance in designing behavior-based robotic systems.

One of the earliest studies demonstrating the use of motor primitives in robot motion control was reported by Arkin in 1989. The object was to develop an experimental mobile robot system that would perform path planning and execution in a way resembling the Arbib and House (1987) model of detour behavior in the frog. The Arbib and House model is an abstraction that characterizes prey acquisition behavior in an obstacle-strewn environment, with primitive vector fields resulting from the frog's perception of its environment. The primitive vector fields are the prey-attractant field, barrier repellent field, and frog representation field. The path-determining vectors can therefore be calculated based on the summation of the primitive vector fields. In autonomous robot architecture (ARA), the path execution mechanism employs a set of concurrently activated motor behaviors called motor schemas. These schemas include *move-to-goal* (similar to the Arbib and House prey-attractant field), *avoid-static-obstacles* (similar to the Arbib and House repellent field), *stay-on-path*, *avoid-moving-obstacles*, *find-intersection*, and *find-landmark* behaviors (see REACTIVE ROBOTIC SYSTEMS).

Force-Field Motor Primitives in a Humanoid

Mataric et al. (1998), drawing on biological findings concerning the organization of force-field motor primitives, have investigated the idea of generating complex motion based on a collection of basic motion primitives, and have demonstrated such a behavior-based motion control approach to a 20-degrees-of-freedom simulated humanoid torso. In their simulation, three behavior primitives are defined as a basic behavior set: *move-to-point*, *get-posture*, and *avoid*. *Move-to-point* is implemented using impedance control such that the arm of the humanoid, as if connected to virtual springs and dampers, can interact with complex environments with stability. *Get-posture* resembles the spinal field of the frog, which is generated by the spring-like muscles in the leg. Running concurrently with other two behavior primitives, *avoid* follows the virtual repulsive forces generated from the obstacles in the torso's environment. Based on the basic behavior primitives, more complex motor

behaviors can be produced, such as *touch right hand to top of left shoulder*.

Behavior Templates and Anchors

With the demonstration that complex motor behaviors can be induced in both animals and animats by incorporating various motor primitives, a new question concerning generalizing the concept of motor primitives arises: Can the dynamically coupled interactions between an organism and its environment, such as locomotion on land, be decoded as the neural control of motor primitives?

Full and Koditschek (1999) have suggested a neuromechanical approach to characterizing legged locomotion on land. Their approach involves the notions of template and anchor. The former provides a minimal behavioral model for guiding the control of locomotion; the latter offers elaborated morphological and physiological mechanisms for a template. Full and Koditschek hypothesize that the control of locomotion incorporates both neural and mechanical systems. During slow, variable-frequency locomotion, the neural system dominates. In the case of rapid, rhythmic locomotion, the mechanical system plays a key role.

Artificial Potential Fields in Modeling External Task Constraints

The problem of robot motion planning has traditionally been treated as an optimization problem in which the configuration of a robot is represented in a parameter space and a solution to this problem is computed by searching the parameter space in an attempt to satisfy a predefined cost function, such as the distance between the robot and a goal point. The limitation of this approach is that it is computationally too costly to generate new plans when dealing with dynamic environments that involve unexpected obstacles. As a more practical approach to the real-time planning of collision-free motions for manipulators and mobile robots, the concept of an artificial potential field (APF) was proposed by Khatib (1986). The APF approach incorporates dynamic sensing feedback into robot control, and hence overcomes the aforementioned limitation by extending the reactivity of the low-level motion control.

APF theory states that for any goal-directed robot in an environment that contains stationary or moving obstacles, an APF can be formulated and computed, taking into account an attractive pole at the goal position of the robot and repulsive surfaces of the obstacles in the environment. This potential field can be expressed as follows:

$$U_{\text{art}}(x) = U_{\text{goal}}(x) + U_{\text{obs}}(x) \quad (1)$$

where $U_{\text{art}}(x)$, $U_{\text{goal}}(x)$, and $U_{\text{obs}}(x)$ denote the APF, the attractive potential from the goal, and the repulsive potential from the obstacles, respectively, and x denotes a set of independent parameters, called operational coordinates, that describe the position and orientation of the robot end-effector.

Generally speaking, U_{obs} is chosen such that U_{art} is a non-negative continuous and differentiable function that tends to infinity when x approaches the surface of an obstacle and tends to zero when x approaches the goal position, x_{goal} .

Given Equation 1, the force resulting from the APF at x can therefore be derived as follows:

$$F_{\text{art}} = -\nabla[U_{\text{art}}(x)] \quad (2)$$

where ∇ denotes a gradient.

Potential Fields for Guiding Motion

Equation 2 tells us that applying APF $U_{\text{art}}(x)$ to a robot end-effector can here be realized by using F_{art} as a command vector to control

the end-effector in operational space (because the motion of the end-effector can be decoupled in operational space; Khatib, 1986). In so doing, the joint forces corresponding to F_{int} must be obtained using the Jacobian matrix. Under such a control, the robot will be able to avoid obstacles (as the repulsive force in the potential field “pushes” it away into the valleys of the field) and at the same time move toward a goal position (as the attractive force in the potential field “pulls” it in the direction of a global zero-potential pole).

By following the potential field that models the spatial constraints in an environment, a robot will be able to avoid obstacles and at the same time achieve stable configurations in its operational space. However, the stable configurations may not be guaranteed to include the goal configuration. In this case, a global motion plan may be used to guide the robot out of a local stable configuration and set it moving toward a goal position. Another alternative is to define an APF function that does not contain a local minimum. An example is a harmonic function defined in the configuration space of a robot where the boundaries of all obstacles and goals are treated as the boundary for the domain of the function.

Other Forms of Potential Fields for Modeling External Task Constraints

Artificial potential fields for modeling external task constraints can take various forms. In the following discussion we consider two generalized APF formulations, *elastic bands* and *elastic strips*, that draw on and generalize the previous work on APF-oriented robot planning and control. These two approaches effectively allow real-time planning and control of robot motion that is both locally *reactive* to any dynamically changing obstacles and globally *optimal* with respect to any motion criteria for attaining a predefined goal.

Elastic Bands

As implemented by Quinlan and Khatib (1993) in a mobile manipulation system, an elastic band has its own internal contraction force when it is in a stretched configuration; at the same time, it receives an external repulsive force if it is close to an obstacle. A *global* collision-free path corresponds to an elastic band at equilibrium that connects initial and goal locations. Whenever a new obstacle approaches, the elastic band will *react* to the situation by deforming itself until a new equilibrium is reached. To compute its artificial forces, an elastic band is represented as a series of consecutive *bubbles*. A bubble at a certain configuration is a spherical free subspace whose radius corresponds to the minimum distance between the configuration and the environment. The total force on the bubble is calculated as follows: The internal contraction force is created by a series of springs connecting the bubbles, whereas the repulsive force is exerted by an obstacle that pushes away the bubble and increases its size. Based on the calculated artificial forces, the position of the bubble will be locally updated, and hence the elastic band deforms.

Elastic Strips

That elastic strips approach real-time motion planning and control was demonstrated by Brock and Khatib (1998). Their work generalizes the notion of bubbles centered at via points into protective hulls that consist of bubbles centered on the spines covering the individual rigid bodies of the manipulator and its mobile platform. Next, an elastic strip is formed by connecting consecutive configurations of the robot on its trajectory, and a tunnel of local free subspace is formed by connecting a series of consecutive protective hulls. With such a representation, it is possible to find the internal contraction force on the elastic strip by calculating the tension between two consecutive configurations for each respective joint of

the robot. It is also possible to find the external repulsive force on the elastic strip by calculating the force acting on the bubbles of a protective hull. Hence, we can determine the deformation of the strip by taking into account the total force acting on the strip, and joint displacements by computing the respective joint torques. The equilibrium elastic strip provides a global collision-free trajectory for the robot that connects the initial and the goal configurations.

Potential Fields in Interaction Controllers

So far we have discussed how potential fields are useful for representing the spatial constraints of a robot task environment in such a way as to effectively guide real-time motion planning and control. Another effective use of potential fields is in the design of robot dynamics that exhibit certain behaviors as if governed by a potential field.

This idea has been part of control systems theory for some time. For instance, Colgate and Hogan (1988) proposed designing a controller for a manipulator that would be capable of dynamically interacting with a diverse set of environments of unknown dynamics and parametric uncertainty. Their specifications for an *interaction controller* emphasize both coupled stability (in addition to nominal stability) and desirable interactive behavior (in addition to command following). The interactive behavior can be established by controlling the impedance of the manipulator as if it were connected with virtual springs and dampers. The stability of a linear system coupled to a passive environment can be guaranteed by making sure the driving point impedance of the system is real and positive.

Artificial Neural Networks

Artificial neural networks (ANNs) are biologically inspired computational models. Each network is composed of a set of neurons connected by fixed synapses. A neuron receives stimuli from external input sources and exchanges messages with other neurons through the connecting synapses. As the strengths of connections between neurons are varied, the neural network builds an associative map between input data patterns and output values. With this computational capability, neural networks have been widely used to solve recognition and classification problems in control, pattern analysis, function learning, feature pattern extraction, and signal processing.

Depending on the homogeneity of neurons, the layered structure of networks, the algorithms for learning input-output associations, and the error propagation mechanisms, different ANNs can be developed, such as the multilayer perceptron, Hopfield networks, backpropagation networks, and Kohonen networks.

APF versus ANN Approaches

Both APF and ANN approaches have been applied to solving practical problems ranging from robot motion planning and control to conceptual mapping and learning.

Given a certain geometric model of a physical environment, it is possible to derive an analytical form of APF for the environment as a function of operational coordinates. An advantage to having an analytical form of APF is that the analytical expression is easier to update if changes occur in the environment. An ANN, on the other hand, builds a numerical input-output map for a set of empirical observations based on a weight-updating and error-correction algorithm. When new empirical data arrive, it is necessary to update the existing ANN by iterating the learning algorithm with each sample of the new data set. As a result of such step-by-step relearning, the weights of an ANN will be modified to some extent,

reflecting the discovery and acquisition of new patterns from the data.

With an APF, a robot can reach a stable configuration in its environment by following the negative gradient of its potential field. In this case, locally stable configurations are inevitable. Nevertheless, they can be readily overcome by either incorporating a global motion planner, or utilizing a harmonic function that does not contain any local minima, or applying generalized APF formulations such as elastic bands and elastic strips. The APF approach is particularly advantageous when dealing with robots with many degrees of freedom in dynamically changing environments.

Similarly, an ANN offers another practical way to solve optimization problems, especially when the search space is of high dimension. With an ANN, the goal is to build an optimal association from given input data patterns to desirable output values. Thereafter the optimal association can be incrementally obtained as the network evolves toward a stable equilibrium state. Unlike the case of an APF, where a stable configuration is approached through updating the configurations following a potential field, the evolution of the neural network relies on updating connection weights and error corrections for the network in accordance with a learning algorithm.

Artificial Neural Network–Based Potential Field Motor Control

An important challenge in the practical applications of APF methodology is to formulate a potential field. For a given robot environment, this task can be decomposed into the subtasks of identifying geometrical primitives in the environment, calculating individual repulsive potential functions for the primitives, and composing a global potential field function based on the individual potential functions. The question that remains is how the APF methodology can be used if the robot environment concerned is not given as a priori knowledge. In such a situation, it would be essential to dynamically derive a numerical potential field representation based on the sensory data obtained during the interaction between the robot and its environment. As mentioned earlier, the ANN methodology is well suited to derive associative maps from certain available input data. In this respect, the ANN approach enables the application of APF in unknown environments.

Operational Space Motor Control

In the past, several researchers have focused on the research question of how to build an operational space potential field map based on real-time sensory measurements. For instance, Prassler (1995) has proposed the use of a massively parallel network of simple processing elements, arranged in a rectangular grid structure, for computing and manipulating a two-dimensional (2D) potential field. The structure is of three layers: a long-term map (LTM) that describes the stationary parts in the environment, a short-term map (STM) that describes a more recent state of the environment, and an occupancy grid representation of the current sensory readings.

Collective Self-Organization

One of the practical concerns in merging APF with an ANN approach is input data requirement. Generally speaking, evolving a stable APF is a time-consuming learning process that requires a large amount of input data coming from the robot-environment interaction. In order to overcome this shortcoming, Liu and Wu (2001) have proposed an evolutionary self-organization learning approach that can efficiently build a potential field map by determining and collecting locally most informative sensory measurements from an unknown environment. This approach can readily

be used by a group of cooperative robots for collective exploration and world modeling.

Joint-Space Motor Control

Besides developing an APF that explicitly models the geometrical characteristics (e.g., clearance) of a robot environment, some studies have focused on how to acquire an APF map directly encoded in the joint space of a robot. Falling into this category are efforts to enable the robot to learn reactive joint activation strategies in response to different external sensory conditions.

An example of such studies is the work on the operant conditioning–based learning of approach and avoidance behaviors in a mobile robot by Chang and Gaudiano (1998). In their study, a wheeled mobile robot is developed that uses a form of self-supervised learning based on an operant conditioning neural network. The output of this neural network is a one-dimensional (1D) population of neurons that encodes the robot's angular velocities, called an angular velocity map, for its left and right wheels. The robot learns its avoidance behavior through "punishment" signals produced by the collision of the robot during random exploratory motion. As a result, a given pattern of sensory inputs will tend to suppress movements that would yield punishment. On the other hand, an excitatory association may be acquired by the robot as it receives a reward such as higher light intensity, with the result that the robot reinforces its movements toward light sources. Unlike the aforementioned 2D operational space APF, the acquired angular velocity map is a 1D representation that directly encodes the potential fields in the joint space of the robot's two wheels.

Artificial Neural Network–Based Perception and Inverse Kinematics for Navigation

Neural networks have been applied not only to build a robot's internal representation of its task environment but also to acquire the control strategies for its collision-free motion. In a vision-based manipulator navigation system, Blase, Pauli, and Bruske (1998) have shown the use of two layers of radial basis function (RBF) networks in the three-dimensional (3D) reconstruction of obstacles from their optical flow vectors. The RBF networks in the first layer are trained for respective image areas to give the depth coordinate z . The depth coordinate z is in turn used by the second layer to generate the corresponding x and y coordinates of the obstacle. In addition, their system also uses one layer of RBF networks to construct the inverse kinematics of a manipulator. Based on the 3D models of obstacles and the inverse kinematics built, the manipulator knows where the obstacles are and how to reach a goal position. While navigating toward the goal position, it also avoids the detected obstacles. In so doing, it dynamically constructs and follows a vector field of simulated forces generated by repulsive forces encoding the obstacle constraints and attractor forces encoding the desired goal.

Summary

In this article, we have examined some of the important biological findings in the use of potential fields to characterize the control and learning of motor primitives. Such biological insights can lead to the development of primitive motor behavior–based robots capable of interacting with dynamic environments. Similarly, the concept of potential fields can be incorporated to model the externally induced constraints as well as the internally constructed sensorimotor maps for robot motion control. Apart from its biological relevance, potential field–based motion control can benefit from the use of ANN-based learning.

Road Map: Robotics and Control Theory

Related Reading: Cognitive Maps; Hippocampus: Spatial Models; Motor Primitives; Reactive Robotic Systems

References

- Arbib, M., and House, D., 1987, Depth and detours: An essay on visually guided behavior, in *Vision, Brain and Cooperative Computation* (M. A. Arbib and A. R. Hanson, Eds.), Cambridge, MA: A Bradford Book/MIT Press, pp. 129–163. ♦
- Arkin, R., 1989, Neuroscience in motion: The application of schema theory to mobile robotics, in *Visuomotor Coordination: Amphibians, Comparisons, Models, and Robots* (J.-P. Ewert and M. Arbib, Eds.), New York: Plenum Press, pp. 649–672.
- Blase, W., Pauli, J., and Bruske, J., 1998, Vision-based manipulator navigation using mixtures of RBF neural networks, in *Proceedings of the International Conference on Neural Networks and Brain*, Peking, China, pp. 531–534.
- Brock, O., and Khatib, O., 1998, Executing motion plans for robots with many degrees of freedom in dynamic environment, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1–6. ♦
- Chang, C., and Gaudiano, P., 1998, Application of biological learning theories to mobile robot avoidance and approach behaviors, *J. Complex Syst.*, 1:79–114.
- Colgate, J. E., and Hogan, N., 1988, Robust control of dynamically interacting systems, *Int. J. Control*, 48:65–88.
- Full, R. J., and Koditschek, D. E., 1999, Templates and anchors: Neuro-mechanical hypotheses of legged locomotion on land, *J. Exp. Biol.*, 202:3325–3332.
- Khatib, O., Spring 1986, Real-time obstacle avoidance for manipulators and mobile robots, *Int. J. Robot. Res.*, 5:90–98.
- Liu, J., and Wu, J., 2001, *Multi-Agent Robotic Systems*, Boca Raton, FL: CRC Press.
- Mataric, M. J., Williamson, M., Demiris, J., and Mohan, A., 1998, Behavior-based primitives for articulated control, in *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB-98)* (R. Pfeifer, B. Blumberg, J.-A. Meyer, and S. W. Wilson, Eds.), Cambridge, MA: MIT Press.
- Mussa-Ivaldi, F. A., and Giszter, S. F., 1992, Vector field approximation: A computational paradigm for motor control and learning, *Biol. Cybern.*, 67:491–500.
- Mussa-Ivaldi, F. A., Giszter, S. F., and Bizzi, E., 1994, Linear combination of primitives in vertebrate motor control, *Proc. Natl. Acad. Sci. USA*, 91:7534–7538.
- Prassler, E., 1995, Robot navigation: A simple guidance system for a complex changing world, in *Modeling and Planning for Sensor Based Intelligent Robot Systems* (H. Bunke, T. Kanade, and H. Noltemeier, Eds.), Singapore: World Scientific, pp. 86–103.
- Quinlan, S., and Khatib, O., 1993, Elastic bands: Connecting path planning and control, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 802–807. ♦
- Thoroughman, K. A., and Shadmehr, R., 2000, Learning of action through adaptive combination of motor primitives, *Nature*, 407:742–747. ♦

Prefrontal Cortex in Temporal Organization of Action

Joaquín M. Fuster

Introduction

The prefrontal cortex is the association cortex of the frontal lobes. It is one of the cortical regions to develop last and most in the course of evolution (Figure 1). In the human brain it constitutes nearly one-third of the totality of the neocortex. Also in the course of individual ontogeny, the prefrontal cortex is one of the cortices to develop last and most. The cellular and connective architecture of the human prefrontal cortex does not reach full maturity until young adulthood. Presumably, the reason for the late morphological development of this cortex, in both phylogeny and ontogeny, is its support of higher cognitive functions related to the capacity to execute novel and complex actions, which reaches its maximum in the adult human brain. The prefrontal cortex of primates can be anatomically subdivided into three major regions: inferior or orbital, medial-cingulate, and lateral. Of the three, the lateral prefrontal cortex, that is, the association cortex of the convexity of the frontal lobe, is the one that undergoes the most development phylogenetically and ontogenetically and is the most implicated in cognitive functions. In the human it is appropriate to label it “the organ of creativity.”

In primates, the cortex of the frontal lobe in its entirety can be considered *motor cortex* in the broadest sense of the term, for it is cortex dedicated to the representation and execution of all manner of actions of the organism: actions in the skeletal domain, in the oculomotor domain, in the visceral domain, in the language domain, and in the domain of complex cognitive operations, such as reasoning and problem solving. The inferior (orbital) and medial regions of the prefrontal cortex are involved in the representation and enactment of emotional behavior and related visceral and autonomic manifestations. The lateral region, on the other hand, is involved in the representation and temporal organization of se-

quential behavior and, in the human, of speech and reasoning. In this article I consider, in particular, the physiological functions of the lateral prefrontal cortex in the temporal organization of behavior.

Anatomy and Connections

By anatomical definition, the prefrontal cortex of the primate is comprised of three major regions (Figure 2), each with a somewhat different cytoarchitecture (Petrides and Pandya, 1994): the *lateral* prefrontal cortex (LPC), or association cortex of the frontal convexity (Brodmann's area 46, and lateral parts of areas 8, 9, 10, and 11); the *medial* and *cingulate* prefrontal cortex, which is nearly flat and faces the medial surface of the contralateral frontal pole (areas 12, 24, and 32, and medial parts of areas 8, 9, 10, and 11); and the *inferior* or *orbital* prefrontal cortex, directly above the orbit of the eye (areas 13, 47, and inferior parts of 10, 11, and 13). The lateral region is bordered in the back by the premotor cortex (area 6); the medial and orbital prefrontal cortices lie anterior to the corpus callosum and limbic structures (piriform cortex and amygdala, cingulate cortex, septum, and hypothalamus).

The prefrontal cortex is one of the best connected of all cortices (Fuster, 1997). It maintains reciprocal fiber connections with a wide range of subcortical and cortical structures. Especially prominent are its connections with the anterior, medial, and dorsal nuclei of the thalamus. In addition, all three prefrontal regions are connected with several limbic structures, especially the hippocampus, the amygdala, and the hypothalamus. Further, the lateral region sends important efferents to the basal ganglia, notably the caudate nucleus, the globus pallidus, and the substantia nigra. Finally, the prefrontal cortex—especially its lateral component—is topologically and reciprocally connected with other frontal areas (premotor,

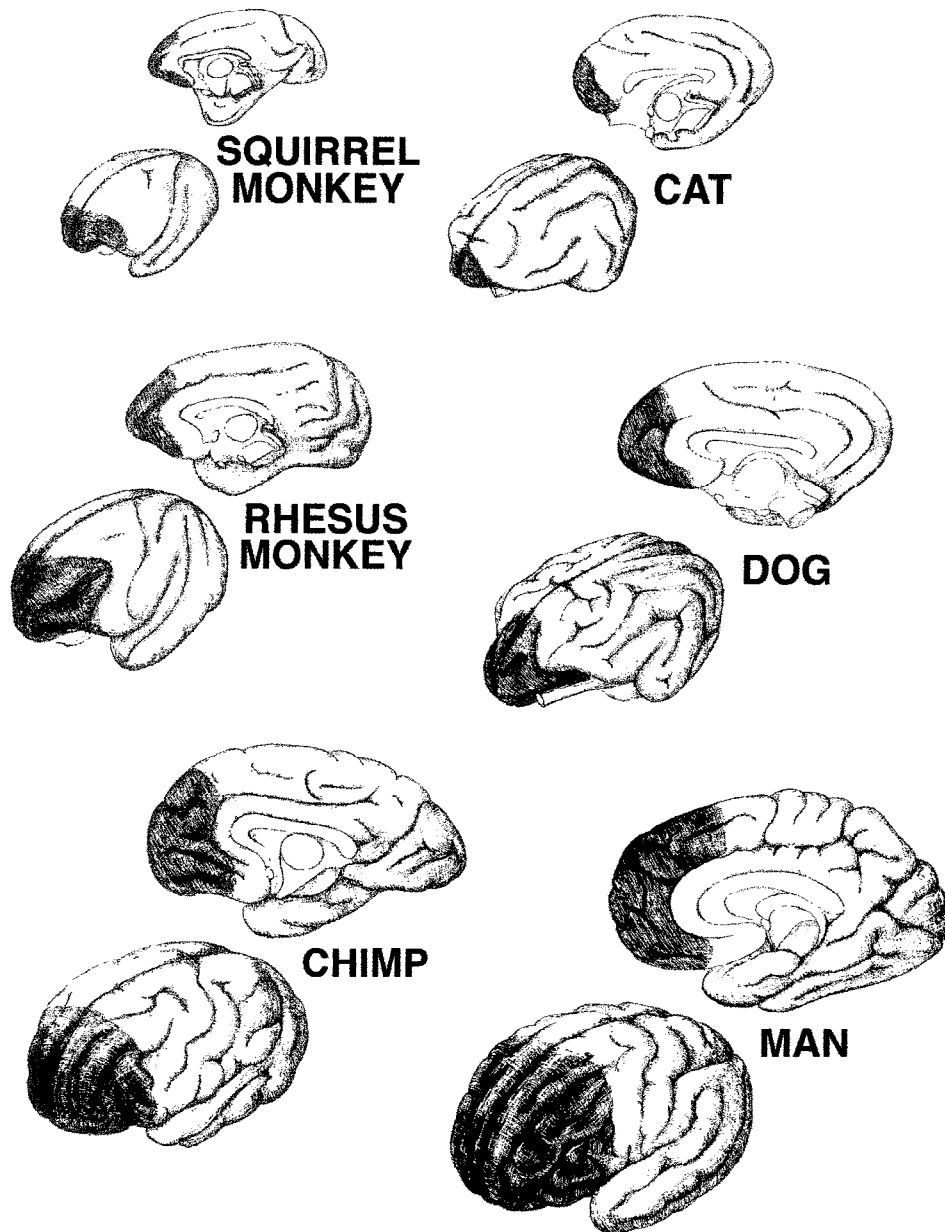


Figure 1. The prefrontal cortex (dark shading) in six mammalian species.

motor cortices) and with areas of posterior cortex, that is, with posterior cortical areas of association in the parietal and temporal lobes (Pandya and Yeterian, 1985).

The connections of the prefrontal cortex with the thalamus mediate inputs from the brainstem and limbic structures to that cortex, as well as inputs from itself (reentrant) and from other cortical regions. However, the functional role of the thalamic connectivity of the prefrontal cortex is still largely unknown. The direct limbic connections of the prefrontal cortex, especially with the amygdala and the hippocampus, are most likely essential for the formation of executive (motor) memory in that cortex, as well as for its retrieval. Connections of ventral and medial prefrontal cortex with limbic and brainstem structures probably serve the functions of the prefrontal cortex in drive, motivation, and the control of emotional behavior. Its outputs to basal ganglia are part of reentry loops that, beyond these nuclei, course through the thalamus and return to frontal cortex; those connective loops play a major, though still

poorly understood, role in motor control. All the corticocortical connections of the prefrontal cortex, especially the lateral region, are important for the cognitive functions of that cortex in the temporal organization of behavior and, presumably, also speech and reasoning.

Physiology

The medial and orbital prefrontal areas are involved in motivation, emotional behavior, and visceral functions. The precise nature of those involvements and their mechanisms is unknown, especially because the relevant evidence derives almost exclusively from the study of the effects of lesions. Large lesions of the medial prefrontal cortex in the human cause apathy and lack of spontaneity in speech, behavior, and reasoning. The orbital prefrontal cortex is essential for the inhibitory control of internal drives and instinctual behavior. Animals or humans with orbitofrontal lesions commonly manifest

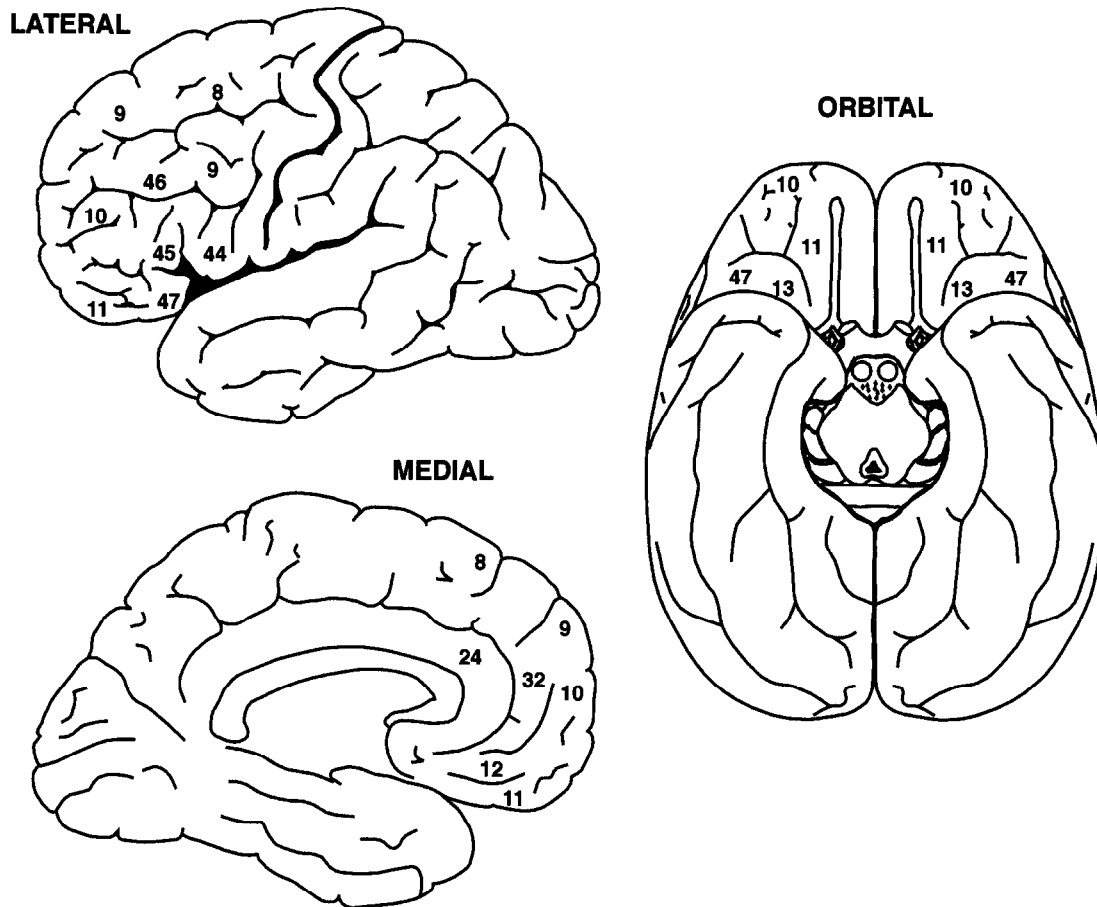


Figure 2. Three views of the frontal lobe. Prefrontal areas designated by numbers in accord with Brodmann's cytoarchitectonic map.

disinhibition of eating, sex, and aggression, as well as abnormal social behavior related to those disinhibitions. The human subject with lesions of that cortex is prone to impulsivity, risk taking, gross humor, and lack of moral judgment. Insofar as appropriate drive and impulse control are essential to sustain attention, the orbito-medial prefrontal cortex is important for the general role of the prefrontal cortex in rational decision making and the organization of cognition and behavior.

The prefrontal cortex of the lateral convexity plays a role in the representation as well as the execution of temporally organized sequences of behavior and cognition (Fuster, 1997); that role extends to language and reasoning. Those representations can be generally categorized as motor or executive memory. This form of memory includes, in particular, the representations of new and complex schemes of sequential action. After they have been learned and automatized, sequential behaviors become part of procedural executive memory and are no longer represented or enacted by the prefrontal cortex (Grafton et al., 1992) but by other structures at lower stages of the neural hierarchy for the organization of movement (e.g., premotor cortices, basal ganglia).

Anatomical, physiological, and neuropsychological studies have led some investigators to infer areas within LPC that specialize in various aspects of cognitive information. In the human as in the monkey, the so-called frontal eye field of area 8 is implicated in visual attention. Further, single-cell studies in the monkey support the commitment of that and other lateral areas to the representation of separate categories of visual information, such as spatial location

or shape (Goldman-Rakic, 1995). Other studies, however, point to the supramodal associative characteristics of cells in all prefrontal areas. Lateral prefrontal cells have been found to associate information across sensory modalities and across time (Fuster, Bodner, and Kroger, 2000). Some of those cells associate sensory stimuli with subsequent motor actions (Rainer, Assad, and Miller, 1998) or rewards (Hikosaka and Watanabe, 2000). These associative properties of lateral prefrontal cells emphasize their integrative functions. Their apparent sensory or motor specificity may be determined by the specificity of their inputs from sensory structures or outputs to motor structures. That specificity seems subservient and secondary to their functions of cognitive integration, which are essential for the temporal organization of behavior.

The connections of the prefrontal cortex with the hippocampus may be critical for the formation of the procedural memory of behavioral sequences in that cortex, in a similar manner as connections between hippocampus and posterior cortex play an important role in the formation of so-called declarative memory (episodic and semantic). The procedural or executive memory networks of the cortex of the frontal lobe seem hierarchically organized (see CORTICAL MEMORY). The primary motor cortex, which is the lowest stage of that organization, represents the elementary motor "memories" of somatic movements executed by synergistic muscle groups. Above the motor cortex, the premotor cortex represents movements by goal and trajectory. At the summit of the cortical motor hierarchy, the LPC represents schemes and sequences of goal-directed action. In addition to representing those schemes and

sequences, the LPC plays a critical role in their execution, especially if they are novel and complex.

Temporal Organization of Behavior

The execution of a sequence of new and complex behavior is a highly elaborate, dynamic process of temporal integration that engages numerous cortical and subcortical brain regions. That process takes place under what may be called the orchestrating functions of the LPC, which ensure the structuring of behavior toward its ultimate goal, whether that goal is the satisfaction of a drive or the solution of a problem (Fuster, 2001). Two temporal integrative functions seem to be at the root of the role of LPC in the formation of behavioral structures. The two are temporally symmetrical and mutually complementary. One is a *retrospective* function of short-term memory, the other a *prospective* function of preparatory set. The two together help the organism to mediate cross-temporal contingencies of behavior. Both engage the executive networks of LPC in functional interactions with lower frontal cortices, subcortical structures, and posterior association cortex.

Short-Term Memory

The LPC function of short-term memory in support of temporal integration approximately coincides with what in cognitive psychology has been named *working memory* (Baddeley, 1992). This function is essentially the temporary retention—"on-line"—of information, old or new, for the execution of an action in the short term, as required for complex behavior or the solution of problems. There is a wealth of evidence that the LPC is critically involved in this process of active short-term memory toward a goal (Fuster, 1997). Selective cortical lesion (or inactivation) of LPC induces deficits in the performance of "delay" tasks (e.g., delayed response, delayed matching). In these tasks, the animal is presented on successive trials with an item of sensory information (which varies from trial to trial) and is required to retain it in memory for executing a given motor response seconds or minutes later. Monkeys with lesions of LPC seem unable to keep the working memory of the stimulus for each trial, as they make many errors, even with short delays. Further evidence for prefrontal short-term memory derives from single-cell recording in animals performing delay tasks. Neurons of LPC fire persistently, and often at stimulus-specific frequency, during the delay (memory period) of every trial (Fuster and Alexander, 1971; Niki, 1974; Funahashi, Bruce, and Goldman-Rakic, 1989). These observations indicate the importance of the neuronal dynamics of LPC for working memory and for the performance of delay tasks that depend on this form of memory.

The function of the prefrontal cortex in active short-term memory is essentially defined by its teleological quality, that is, by the presence of an objective in the near future, which is the construction of prospective action toward a goal. As noted above, some prefrontal areas seem to specialize in the sensory and motor aspects of the information retained in working memory. That apparent specialization may derive from different concentrations of specific inputs and outputs in those areas. In general terms, however, each area is probably engaged in the temporary activation of specific sensory-sensory or sensory-motor associations leading to organized sequential action. In those terms, the cell groupings and areas of LPC are components of teleological associations, the mnemonic paths to prospective action.

The precise neural mechanism of working memory is still unknown. Based on physiological evidence from the monkey (summary review in Fuster, 1997), it appears that working memory is maintained by the persistent activation of the widely distributed cortical network that represents its content (see CORTICAL MEMORY). The network contains neuronal assemblies of sensory rep-

resentation in posterior cortex and neuronal assemblies of motor or executive representation in LPC. A plausible assumption, still unproved, is that the activation of one such network serving working memory depends on the reverberation of neural excitation within it. The excitation of the network would thus be sustained through reentrant loops of reciprocal connection between lateral prefrontal and posterior—e.g., inferotemporal, parietal—cortex. Hence, recurrent cortical architecture is an essential feature of the most plausible and empirically testable computational models of working memory, as well as of other cognitive functions (Zipser et al., 1993; Duncan, 2001; Wang, 2001).

Prospective Set

Likewise, from the teleological nature of the temporal structuring of behavior derives the prefrontal function of readiness to act. The setting of sensory receptors and motor effectors for prospective actions is, in the time domain, the mirror image of active short-term memory. This complementary integrative function of preparatory set is the counterpart of working memory. It serves to prepare the subject for anticipated actions; it is the "memory of the future" that will complement retrospective memory in the bridging of a cross-temporal contingency. The neural mechanisms of prospective set are not yet well understood. These mechanisms probably result in the priming of sensory and motor structures by way of efferent connections of the prefrontal cortex. In both human and nonhuman primates, the involvement of LPC in preparatory set is reflected by the progressive increase in cell discharge and the slow surface-negative potentials that take place in that cortex before a stimulus-contingent act. Both memory and set seem to occur at the same time and in the same areas of LPC, both simultaneously during the interval that separates two mutually contingent events. While some prefrontal neurons engage in the memory of the cue, others engage in the preparation of the consequent and subsequent behavioral response. Neuronal recording from the LPC shows that, while the memory of the sensory information wanes, the representation of the response in preparation increases (Quintana and Fuster, 1999).

The selection of an action among many is the motor equivalent of focusing sensory attention. Consequently, the prefrontal function of prospective set may be appropriately considered *motor attention*, that is, the selective focusing of attention on a motor act to ensure the prompt and efficient execution of that act in the near future. In conclusion, the symmetry of the two temporal integrative functions of the LPC, active memory and set, parallels the symmetry of attentive processes, one sensory and the other motor. Active short-term memory is attention focused on the representation of a recent sensory stimulus, while prospective set is attention focused on a subsequent and consequent act. Whereas attention, in particular sensory attention, is commonly associated with conscious awareness, neither of the two temporal integrative functions of the LPC, active memory or set, need be conscious to accomplish their objective.

Perception-Action Cycle

The temporal organization of complex behavior requires a continuous succession of sensorimotor integrations, that is, a succession of temporal mediations of contingencies between sensory events and consequent motor acts. To some degree, each integration depends on previous ones. Each sensorimotor integration induces certain changes in the environment that will determine and modify subsequent sensory inputs, and these will determine and modify subsequent acts, and so on. This cybernetic cycle of interactions of the organism with its environment, through a series of sensorimotor integrations, is a fundamental principle in biology: the perception-action cycle.

The neural operations of the perception-action cycle ensure that mutually contingent percepts and acts are properly integrated in the progression of behavior toward its goal. The attainment of that goal requires the attainment of lesser or subordinate goals, each dependent on a particular translation of perception to action and its consequent change in the environment. Innumerable neural structures participate in the mediation of perception-action contingencies that one such sequence requires. These structures, both on the sensory side and on the motor side, are hierarchically organized throughout the nerve axis, from the spinal cord to the cerebral cortex, and also within the latter. Figure 3 shows highly schematically the cortical layers of the perception-action cycle. If the behavior is new, all layers of that cycle are involved in its temporal organization, up to and including the cortex. Routine and automatic sequences, such as walking, are relegated to and organized at lower levels. The cortex remains involved even if the behavioral sequence is old and well rehearsed but contains uncertainties or ambiguities.

Goal-directed behavioral sequences may contain delays or temporary interruptions imposed by the subject or by the environment. Such is the case with the so-called delay tasks, where time intervenes between a percept and an action that are contingent from each other. Because that contingency must be mediated cross-temporally, the correct performance of those tasks depends on the LPC. The LPC, through the two temporally integrative functions of short-term memory and prospective set, integrates perception with action across time at the highest level of the hierarchy of neural structures serving the perception-action cycle. There is considerable empirical evidence that the LPC performs those temporal integrative functions in dynamic cooperation with posterior cortical areas of association, which constitute the highest substrate for the perceptual side of the cycle. The highest and most characteristically human operations of the perception-action cycle take place in the

construction of the spoken language. Given what we know about the role of posterior cortex (Wernicke's area) and LPC in the perceptual and motor aspects of language, respectively, it is reasonable to view those cortices as the highest levels of the perception-action cycle for speech. At the neural root of a dialogue between two persons, it is appropriate to hypothesize a dynamic interaction of the cycles of the two interlocutors. The action of one is the perception of the other, and vice versa. The frontal and postcentral cortices of the two subjects would thus interact reciprocally in the coordination of two perception-action cycles organizing the dialogue in the temporal domain.

Discussion

The association cortex of the frontal lobe, or prefrontal cortex, is highly heterogeneous, anatomically as well physiologically. Further, this cortex is widely connected with many other brain structures, cortical and subcortical. The heterogeneity and diverse connectivity of the prefrontal cortex serve a variety of functions related to the organization of actions in the temporal domain. This article has focused on the temporal organizing functions of the LPC, the prefrontal region that undergoes the greatest phylogenetic and ontogenetic development in primates. Two of those functions have been highlighted that are especially evident in the primate: active short-term memory (working memory) and prospective set. The two cooperate toward temporally integrating sensory and motor information by mediating cross-temporal contingencies of behavior. This integrative role of the prefrontal cortex covers a wide range of sensory inputs and motor outputs; hence the apparent sensory or motor specificity of certain prefrontal areas. Temporal integration seems to be the overarching function that, despite their heterogeneity, unifies the many activities and functions of the LPC.

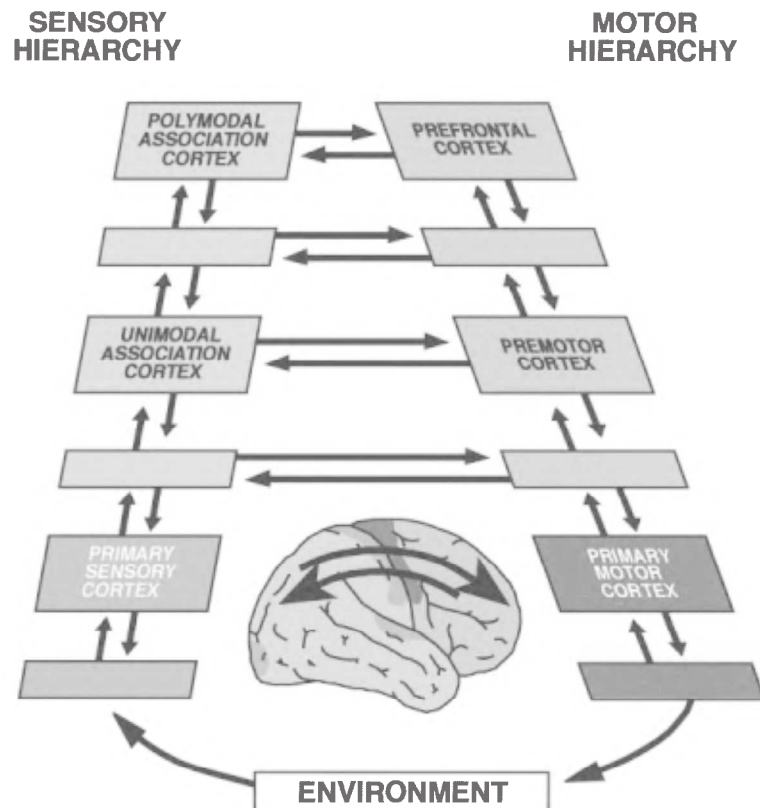


Figure 3. The cortical hierarchies of the perception-action cycle depicted around a lateral view of the human brain. Unlabeled cortical entities represent subareas of labeled regions or areas interposed between them. The arrows represent connections that have been demonstrated in the monkey.

Temporal integration, through memory and set, supports the goal-directed performance of the perception-action cycle. It is a role that extends to the temporal organization of higher cognitive operations, including spoken language.

Road Map: Mammalian Brain Regions

Related Reading: Basal Ganglia; Competitive Queuing for Planning and Serial Performance; Grasping Movements: Visuomotor Transformations; Sequence Learning; Thalamus; Visual Scene Perception

References

- Baddeley, A., 1992, Working memory, *Science*, 255:556–559.
- Duncan, J., 2001, An adaptive coding model of neural function in prefrontal cortex, *Nature Neurosci.*, 2:820–829. ♦
- Funahashi, S., Bruce, C. J., and Goldman-Rakic, P. S., 1989, Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex, *J. Neurophysiol.*, 61:331–349.
- Fuster, J. M., 1997, *The Prefrontal Cortex*, 3rd ed., Philadelphia: Lippincott-Raven. ♦
- Fuster, J. M., 2001, The prefrontal cortex—an update: Time is of the essence, *Neuron* ♦
- Fuster, J. M., and Alexander, G. E., 1971, Neuron activity related to short-term memory, *Science*, 173:652–654.
- Fuster, J. M., Bodner, M., and Kroger, J., 2000, Cross-modal and cross-temporal association in neurons of frontal cortex, *Nature*, 405:347–351.
- Goldman-Rakic, P. S., 1995, Architecture of the prefrontal cortex and the central executive, *Proc. Natl. Acad. Sci. USA*, 769:71–83. ♦
- Grafton, S. T., Mazziotta, J. C., Woods, R. P., and Phelps, M. E., 1992, Human functional anatomy of visually guided finger movements, *Brain*, 115:565–587.
- Hikosaka, K., and Watanabe, M., 2000, Delay activity of orbital and lateral prefrontal neurons of the monkey varying with different rewards, *Cereb. Cortex*, 10:263–271.
- Niki, H., 1974, Prefrontal unit activity during delayed alternation in the monkey: I. Relation to direction of response, *Brain Res.*, 68:185–196.
- Pandya, D. N., and Yeterian, E. H., 1985, Architecture and connections of cortical association areas, *Cereb. Cortex*, 4:3–61. ♦
- Petrides, M., and Pandya, D. N., 1994, Comparative architectonic analysis of the human and the macaque frontal cortex, in *Handbook of Neuropsychology* (F. Boller and J. Grafman, Eds.), Amsterdam: Elsevier, pp. 17–58.
- Quintana, J., and Fuster, J. M., 1999, From perception to action: Temporal integrative functions of prefrontal and parietal neurons, *Cereb. Cortex*, 9:213–221. ♦
- Rainer, G., Assad, W. F., and Miller, E. K., 1998, Selective representation of relevant information by neurons in the primate prefrontal cortex, *Nature*, 393:577–579.
- Wang, X., 2001, Synaptic reverberation underlying mnemonic persistent activity, *Trends Neurosci.*, 24:455–463.
- Zipser, D., Kehoe, B., Littlewort, G., and Fuster, J., 1993, A spiking network model of short term active memory, *J. Neurosci.*, 13:3406–3420. ♦

Principal Component Analysis

Erkki Oja

Introduction

Principal Component Analysis (PCA) and the closely related Karhunen-Loève transform, or the Hotelling transform, are standard techniques in feature extraction and data compression (see, e.g., Devijver and Kittler, 1982; Oja, 1983). In general terms, the input vectors in PCA are random vectors x with K elements. In PCA, no assumptions on the probability density of the vectors are needed, as long as their means and covariances are known or can be estimated from a sample. This is in contrast to another well-known statistical technique, factor analysis, that is based on an explicit Gaussian model. Typically the elements of x are measurements such as pixel gray levels or the values of a signal at different time instants. They are mutually correlated.

In the PCA transform, vector x is linearly transformed to another vector y with N elements, $N < K$, so that the redundancy induced by the correlations is removed. This is done by finding a rotated coordinate system such that the elements of x in the new coordinates become uncorrelated. For instance, if x has a Gaussian density that is constant over ellipsoidal surfaces in the K -dimensional space, then the rotated coordinate system coincides with the principal axes of the ellipsoid. Even if the density is not Gaussian, similar principal axes can be computed. For most practical data, the density is not spherical but strongly elongated, and thus the axes have very different lengths. A considerable number of the axes are so small that the components they represent can be discarded altogether. Those components that are left constitute vector y .

As an example, take a sequence of small 32×32 digital images of hand-written characters. In real-time digital video transmission, it would be essential to reduce such images as much as possible without losing too much of the visual quality, because the total amount of data is very large: 1024 pixels for each image. Using PCA, a compressed representation vector is obtained for each

image, which can be stored or transmitted. Figure 1 shows two examples of such characters from a large database and the reconstruction from the compressed PCA representation. This is the fundamental idea behind practical image and signal compression methods like the discrete cosine transform.

In mathematical terms, consider a linear combination

$$y_1 = \sum_{k=1}^K w_{k1} x_k = w_1^T x$$

of the elements x_1, \dots, x_K of vector x , where w_{11}, \dots, w_{K1} are scalar coefficients or weights, elements of a K -dimensional vector w_1 , and w_1^T denotes the transpose of w_1 . Usually it is assumed that x has zero mean; if not, then the mean vector is estimated separately and subtracted from x to obtain a zero mean vector.

The factor y_1 is the first principal component of x if the variance of y_1 is maximally large under the constraint that the norm of w_1 is constant (see, e.g., Devijver and Kittler, 1982). Then the weight vector w_1 maximizes the PCA criterion



Figure 1. Results of using PCA to compress images. In the leftmost column are two digital images in a 32×32 grid. The next column shows the mean of all the samples. The remaining columns show the images as reconstructed by PCA when 1, 2, 5, 16, 32, or 64 principal components (out of the total of 1024) were used in the expansion.

$$\begin{aligned} J_1^{PCA}(w_1) &= E\{y_1^2\} = E\{(w_1^T x)^2\} = E\{(w_1^T x)(x^T w_1)\} \\ &= w_1^T E\{xx^T\} w_1 = w_1^T C w_1, \|w_1\| = 1 \end{aligned} \quad (1)$$

where $E\{\cdot\}$ is the expectation over the density of input vector x , and the norm of w_1 is defined as

$$\|w_1\| = (w_1^T w_1)^{1/2} = \left[\sum_{k=1}^K w_{k1}^2 \right]^{1/2}$$

The matrix C in Equation 1 is the $K \times K$ covariance matrix defined by

$$C = E\{xx^T\} \quad (2)$$

The solution is given in terms of the unit-length eigenvectors c_1, \dots, c_K of the matrix C . With $\lambda_1, \dots, \lambda_K$ the corresponding eigenvalues in decreasing order (or nonincreasing, in case of multiple eigenvalues), the solution is given by

$$w_1 = c_1$$

Thus the first principal component of x is given by $y_1 = c_1^T x$.

The criterion J_1^{PCA} in Equation 1 can be generalized to N principal components, with N any number between 1 and K . Denoting the n th ($1 \leq n \leq N$) principal component by $y_n = w_n^T x$ with w_n the corresponding weight vector, the variance of y_n is maximized under the constraints

$$\|w_n\| = 1, w_n^T w_m = 0, m < n \quad (3)$$

Note that compared to the first principal component, there is now the extra constraint that the weight vector w_n must be orthonormal with all the previous weight vectors. The solution is

$$w_n = c_n$$

thus the n th principal component is $y_n = c_n^T x$. It follows that

$$E\{y_n^2\} = E\{c_n^T x x^T c_n\} = c_n^T C c_n = \lambda_n$$

This can often be used in advance to determine N , if the eigenvalues are known. The eigenvalue sequence $\lambda_1, \lambda_2, \dots$ is usually sharply decreasing, and it is possible to set a limit below which the eigenvalues, hence principal components, are insignificantly small. This limit determines how many principal components are used.

Another possible extension of Equation 1 is:

$$\begin{aligned} J_N^{PCA}(w_1, \dots, w_N) &= E\left\{ \sum_{n=1}^N y_n^2 \right\} = E\left\{ \sum_{n=1}^N (w_n^T x)^2 \right\} \\ &= \sum_{n=1}^N w_n^T C w_n = \max \end{aligned} \quad (4)$$

$$w_m^T w_n = \delta_{mn} \quad (5)$$

This criterion determines the subspace spanned by vectors w_1, \dots, w_N in a unique way as the subspace spanned by the N first eigenvectors c_1, \dots, c_N , but does not specify the basis of this subspace at all.

To use the closed-form solutions given above, the eigenvectors of the covariance matrix C must be known. This is rarely true in practice. In an on-line data compression application like image or speech coding, it is usually not possible to solve the eigenvector-eigenvalue problem for computational reasons. The PCA solution is then replaced by suboptimal standard transformations. Another alternative is to derive gradient ascent algorithms for the maximization problems above. The algorithms will then converge to the solutions of the problems, i.e., to the eigenvectors. This approach is the basis of the neural network learning rules.

PCA Learning Algorithms and Neural Networks

Neural networks provide a novel way for parallel on-line computation of the PCA expansion. The PCA network (Oja, 1992) is a

layer of parallel linear artificial neurons (Figure 2). The output of the n th unit ($n = 1, \dots, N$) is $y_n = w_n^T x$, with x denoting the K -dimensional input vector of the network and w_n denoting the weight vector of the n th unit. The number of units, N , will determine how many principal components the network will compute. Sometimes this can be determined in advance for typical inputs, or N can be equal to K if all principal components are required.

The PCA network learns the principal components by unsupervised learning rules, by which the weight vectors are gradually updated until they become orthonormal and tend to the theoretically correct eigenvectors. The network also has the ability to track slowly varying statistics in the input data, maintaining its optimality when the statistical properties of the inputs do not stay constant. Because of their parallelism and adaptivity to input data, such learning algorithms and their implementations in neural networks are potentially useful in feature detection and data compression tasks.

Some basic learning algorithms are listed here. In the following, k denotes discrete time; thus $x(k)$ is a stream of input data vectors (e.g., image windows or segments of a time-varying signal) entering the learning neural network. The learning weight vectors are $w_j(k)$, $j = 1, \dots, N$.

The Stochastic Gradient Ascent (SGA) Algorithm

This algorithm (Oja, 1983) is obtained from the maximum variance criterion by taking the gradients with respect to weight vector w_j and using the normalization constraints. Denoting

$$\Delta w_j(k-1) = w_j(k) - w_j(k-1) \quad (6)$$

the learning rule is

$$\begin{aligned} \Delta w_j(k-1) &= \gamma(k) y_j(k) [x(k) - y_j(k) w_j(k-1) \\ &\quad - 2 \sum_{i \neq j} y_i(k) w_i(k-1)] \end{aligned} \quad (7)$$

There, $\gamma(k)$ are the step sizes in the gradient ascent, typically a sequence of small numbers tending slowly to zero.

The first term on the right contains the product $y_j(k)x(k)$, which is a Hebbian term—note that $y_j(k) = w_j(k-1)^T x(k)$ is the output of the j th neuron at time k , and $x(k)$ is the input—and the other terms are implicit orthonormality constraints. The case $j = 1$ gives

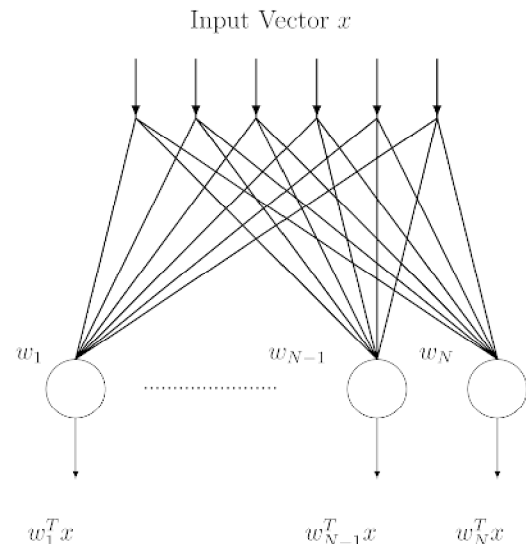


Figure 2. The basic linear PCA layer.

the constrained Hebbian learning rule of the basic PCA neuron introduced by Oja (1982). The convergence of the vectors $w_1(k), \dots, w_N(k)$ to the eigenvectors c_1, \dots, c_N was established by Oja (1983). A modification called the generalized Hebbian algorithm (GHA) was presented by Sanger (1989), who also applied it to image coding, texture segmentation, and the development of receptive fields.

The algorithm may have significance in hierarchical clustering of learned cues in the cerebral cortex. Ambros-Ingerson, Granger, and Lynch (1990) performed simulations on the olfactory paleocortex receiving inputs from the olfactory bulb. They used a network model resembling the SGA algorithm in which the first neuron (in their case, a competitive subnet) learned the input as such, and consequent neurons (subnets) learned progressively masked versions of the input. Masking corresponds to subtracting from the input the previous weight vectors, as in Equation 7. The simulation revealed how perceptual hierarchies may arise for recognizing environmental cues.

The Subspace Network Learning Algorithm

The subspace network learning algorithm (Oja, 1983; Williams, 1985) is formulated as follows:

$$\Delta w_j(k-1) = \gamma(k) y_j(k) [x(k) - \sum_{i=1}^N y_i(k) w_i(k-1)] \quad (8)$$

It is obtained as a gradient ascent maximization of criterion 4. The network implementation is analogous to the SGA algorithm but still simpler, because the feedback term, depending on the other weight vectors, is the same for all neuron units. Thus, learning at an individual connection weight w_{ji} is local, as it depends only on y_j, x_j , and the feedback term, all of which are easily accessible at that position in a hardware network. The convergence was studied by Williams (1985), who showed that the weight vectors $w_1(k), \dots, w_N(k)$ will not tend to the eigenvectors c_1, \dots, c_N but only to a rotated basis in the subspace spanned by them, in analogy with the subspace criterion discussed earlier. A global convergence analysis was given by Yan, Helmke, and Moore (1994).

The Recursive Least Squares Algorithm and Extensions

On-line algorithms typically suffer from slow convergence. The learning rate would have to be tuned optimally. One way of doing this is to use recursive least squares methods. It is well known that they converge much faster at the expense of a somewhat larger computational cost. An efficient algorithm called the Projection Approximation Subspace Tracking (PAST) was introduced by Yang (1996).

Also, minor components defined by the eigenvectors corresponding to the smallest eigenvalues can be computed by similar algorithms (see Oja, 1992). A recent overview of these and related neural network realizations of signal processing algorithms is given by Cichocki and Unbehauen (1993) and Diamantaras and Kung (1996). Extensions to nonlinear PCA learning rules are given in Hyvärinen, Karhunen, and Oja (2001).

Learning PCA by Backpropagation Learning

Another possibility for PCA computation in neural networks is the multilayer perceptron network, which learns by the backpropagation algorithm in unsupervised autoassociative mode. This network was suggested for data compression by Cottrell, Munro, and Zipser (1987).

In the three-layer autoassociative perceptron net, the input and output layers have K units and the one hidden layer has $N < K$ units. The outputs of the hidden layer are given by

$$h = \sigma(W_1 x + w_1) \quad (9)$$

where W_1 is the input-to-hidden layer weight matrix, w_1 is the corresponding bias vector, and σ is the usually nonlinear activation function, to be applied elementwise. The output y is an affine linear function of hidden layer outputs:

$$y = W_2 h + w_2 \quad (10)$$

with obvious notation. In autoassociative mode, the same vectors x are used both as inputs and as desired outputs in backpropagation learning. If σ is linear, then the hidden layer outputs will become the principal components of x .

For the linear net, backpropagation learning is especially feasible because it was shown by Baldi and Hornik (1989) that the “energy” function has no local minima. The nonlinear case was analyzed by Japkowitz, Hanson, and Gluck (2000). The three-layer net has been used for image compression by Cottrell et al. (1987).

Discussion

The algorithms reviewed in this article are typical learning rules for adaptive PCA extraction, and they are especially suitable for neural network implementations. In numerical analysis and signal processing, many other algorithms have been reported for different computing hardware (for a review, see Comon and Golub, 1990). PCA is a useful technique for, e.g., spatial decorrelation and denoising. Experimental results on the PCA algorithms both for finding the eigenvectors of stationary training sets and for tracking the slowly changing eigenvectors of nonstationary input data streams have been reported by Oja (1983) and Sanger (1989). An obvious extension of PCA neural networks would be to use nonlinear units, such as perceptrons, instead of the linear units, as suggested by Xu (1991). Such nonlinear PCA networks will in some cases give the independent components of the input x , instead of just uncorrelated components. This is due to the higher-order statistics that are induced by the nonlinearities. The technique of INDEPENDENT COMPONENT ANALYSIS (i.c.a.) is very useful in blind source separation (Hyvärinen et al., 2001). In fact, a useful preprocessing step in ICA is whitening, which means performing PCA followed by variance normalization.

Road Map: Learning in Artificial Networks

Related Reading: Data Clustering and Learning; Independent Component Analysis; Learning Vector Quantization; Perceptrons, Adalines, and Backpropagation

References

- Ambros-Ingerson, J., Granger, R., and Lynch, G., 1990, Simulation of paleocortex performs hierarchical clustering, *Science*, 247:1344–1348.
- Baldi, P., and Hornik, K., 1989, Neural networks and principal components analysis: Learning from examples without local minima, *Neural Netw.*, 2:52–58.
- Cichocki, A., and Unbehauen, R., 1993, *Neural Networks for Optimization and Signal Processing*, New York: Wiley. ♦
- Comon, P., and Golub, G., 1990, Tracking a few extreme singular values and vectors in signal processing, *Proc. IEEE*, 78:1327–1343. ♦
- Cottrell, G. W., Munro, P. W., and Zipser, D., 1987, *Image Compression by Back-propagation: A Demonstration of Extensional Programming*, Technical Report 8702, University of California, San Diego, Institute of Cognitive Science.
- Devijver, P. A., and Kittler, J., 1982, *Pattern Recognition: A Statistical Approach*, London: Prentice-Hall. ♦
- Diamantaras, K., and Kung, S., 1996, *Principal Component Neural Networks: Theory and Applications*, New York: Wiley. ♦
- Hyvärinen, A., Karhunen, J., and Oja, E., 2001, *Independent Component Analysis*, New York: Wiley. ♦
- Japkowitz, N., Hanson, S. J., and Gluck, M. A., 2000, Nonlinear autoassociation is not equivalent to PCA, *Neural Comput.*, 12:531–545.

- Oja, E., 1982, A simplified neuron model as a principal components analyzer, *J. Math. Biol.*, 15:267–273.
- Oja, E., 1983, *Subspace Methods of Pattern Recognition*, Letchworth, Engl.: Research Studies Press and Wiley. ♦
- Oja, E., 1992, Principal components, minor components, and linear neural networks, *Neural Netw.*, 5:927–935. ♦
- Sanger, T. D., 1989, Optimal unsupervised learning in a single-layer linear feedforward network, *Neural Netw.*, 2:459–473.
- Williams, R., 1985, *Feature Discovery Through Error-Correcting Learn-*

- ing*, Technical Report 8501, University of California, San Diego, Institute of Cognitive Science.
- Xu, L., 1991, Least mean square error reconstruction principle for self-organizing neural nets, *Neural Netw.*, 6:627–648.
- Yan, W., Helmke, U., and Moore, J., 1994, Global analysis of Oja's flow for neural networks, *IEEE Trans. Neural Netw.*, 5:674–683.
- Yang, B., 1996, Asymptotic convergence analysis of the Projection Approximation Subspace Tracking algorithm, *Signal Process.*, 50:123–126.

Probabilistic Regularization Methods for Low-Level Vision

Jose L. Marroquin and Mariano Rivera

Introduction

Current research in computational vision follows two main paradigms. In the first paradigm, the primary task that a visual system must solve is considered to be reconstructing, from the set of images that constitute the sensory input, a set of fields that represent, on the one hand, the physical properties of the three-dimensional surfaces around the viewer, and on the other, the boundaries between patches that “belong together” in some sense and thus may correspond to the outlines of plausible physical objects in the scene. This process, which is usually called early or low-level vision, is supposed to be performed in natural systems by a set of loosely coupled neural networks (computational modules), each specializing in the reconstruction of a particular field. Thus, specific modules have been proposed for the computation of brightness edges; depth (from stereo, shading, and motion); color, lightness and albedo; velocity and optical flow; spatial and spatiotemporal interpolation and approximation, and so on.

In the second paradigm, many of the problems to be solved using vision are thought not to need a complete reconstruction of the three-dimensional world; for a given task, it may be possible to feed the raw sensory data to a network (such as a multilayer perceptron) that directly generates the desired control commands. The plausibility of this approach is illustrated, for example, in Pomerleau (1991), where such a network is used for an autonomous navigation task. In this case, however, it is also necessary to determine a set of fields defined on the same lattice as the observations: these fields represent the weights that indicate the relative importance of each pixel value for the subnetwork of the corresponding hidden unit.

In both cases, the determination of the corresponding fields exhibits an important common characteristic due to the loss of information inherent to imaging and sensory transduction processes and, in the second case, to the fact that one usually has a limited number of available examples to train the network: the values of the fields are constrained by the data but are not determined in a unique and stable way (i.e., the reconstruction problems are mathematically ill-posed). This means that the networks that implement the solutions must incorporate in their structure prior knowledge about the reconstructed fields.

For the sake of clarity, this article focuses on the reconstruction (multimodule) paradigm, although most of the results may be extended to the action-oriented case as well. The general problem that we consider is the following.

Suppose that we are given sensory measurements in the form of a set of observed fields g at the nodes of a regular lattice L (usually a square lattice is assumed, although other arrangements are possible). From these measurements, we wish to reconstruct a field f

$= \{f_i, i \in L\}$, given the “direct” equations that model g in terms of f and some noise process n :

$$\phi(g, f, n) = 0 \quad (1)$$

The simplest instance of this problem is image filtering, in which g consists of a single field (the noisy observed image), f is the desired reconstructed image, and the observation model is

$$g - f - n = 0 \quad (2)$$

Another example is the recovery of depth from stereoscopic pairs of images. Here, the observations $g = (g_L, g_R)$ are the gray levels measured in the left and right retinas, respectively, and f is the associated disparity between pairs of corresponding points (if this “correspondence problem” is solved, and if the geometry of the sensors is known, the actual recovery of depth is a matter of simple geometric computations). If the sites of the lattice are identified by a two-dimensional (2D) index $i = (i_x, i_y)$, and assuming horizontal epipolar lines, a simplified direct equation is

$$g_L(i_x, i_y) - g_R(i_x + f_i, i_y) - n_i = 0$$

for each $i \in L$.

Another example is image segmentation. Here, the input lattice is partitioned into a set of nonoverlapping regions $\{R_1, \dots, R_M\}$ so that the spatial variation of the observed images is represented by a parametric model $\Psi(i, \theta_k)$ inside region R_k :

$$g_i = \sum_{k=1}^M \Psi(i, \theta_k) f_{ik} + n_i \quad (3)$$

where f_{ik} is the indicator variable of region R_k : $f_{ik} = 1$ iff $i \in R_k$ and $\{\theta_1, \dots, \theta_M\}$ are the parameter vectors.

In the first example, the field f is underconstrained, because the noise field is not known. In the second example, even in the absence of noise, the field f is not uniquely determined, because there may be many points in the right image with the same gray level of a given point in the left one. Finally, in the third example, non-uniqueness arises because of measurement noise and because neither the parameter vectors nor the indicator variables are known. Similar ambiguous situations arise in other early vision problems for different reasons, and in all these cases it is necessary to introduce additional prior constraints.

In this article we present systematic ways for adding prior constraints, and for embedding the solution algorithms in suitable networks.

Probabilistic Regularization

The classical way of finding solutions to ill-posed problems is based on regularization methods, where stability and uniqueness of

the solution are enforced by the introduction of prior smoothness constraints in the solution. A more general approach, and one that includes the classical solution as a particular case, is probabilistic, and considers f and g as realizations of random fields, so that the reconstruction of f is understood as an estimation problem. The prior knowledge about the solution is expressed in the form of a joint probability distribution for f that specifies the desired dependencies between values at neighboring sites. In this way, one may specify not only global smoothness constraints (as in standard regularization) but also piecewise smoothness, as well as constraints on the shape of the discontinuities.

The basic tool in this approach is Bayes's rule, which specifies the way in which prior information (i.e., the prior distribution P_f) is to be combined with the constraints generated by the observations (i.e., the conditional distribution $P_{g|f}$) to generate the posterior distribution $P_{f|g}$:

$$P_{f|g}(f; g) = \frac{P_f(f)P_{g|f}(f; g)}{P_g(g)} \quad (4)$$

Note that since the observations g are given, $P_g(g)$ is a constant. The optimal estimator \hat{f}^* is then obtained as the minimizer of the expected value (taken with respect to the posterior distribution) of an appropriate cost function $C(f, \hat{f})$.

This approach, then, requires the specification of three basic components (besides the cost function): the observation model $P_{g|f}$, the prior distribution P_f , and the network that will effect the reconstruction. We will now analyze them in detail.

The Observation Model

The form of the constraints that sensor measurements impose on the reconstructed field depends on the particular assumptions that are made about the image formation process. If the random variables n_i , $i \in L$, are assumed to be independent, identically distributed with distribution P_n , then the conditional distribution is found by solving for n in Equation 1: $n_i = \phi^{-1}(f, g)$ and setting

$$P_{g|f}(f; g) = \prod_{i \in L} P_n(\phi^{-1}(g, f))$$

which can be written in the general form:

$$P_{g|f}(f; g) = \exp \left[\sum_{i \in L} -\Phi_i(f, g) \right] \quad (5)$$

In most cases, the functions Φ_i are quadratic—i.e., the noise is assumed to be Gaussian—although other forms that reduce the influence of gross measurement errors have also been used (see Black and Rangarajan, 1996).

Prior Distributions

The success of the Bayesian approach depends on the specification of a probability distribution $P_f(f)$ that models the desired behavior of the solution. In particular, one would like to be able to specify a distribution in which fields where neighboring sites exhibit the appropriate dependencies are more probable than those in which these local constraints are violated. A general way of constructing such distributions is by defining an “energy” function $U(f)$, which is formed by a sum of terms that measure the violation of the local constraints. The probability distribution of the field is then given by the Gibbs measure:

$$P_f(f) = \frac{1}{Z} \exp[-U(f)] \quad (6)$$

where Z is a normalizing constant.

More precisely, if we define a neighborhood system $\{N_i, i \in L\}$, that is, a collection of subsets of sites indexed by the sites of L : $\{N_i \subset L, i \in L\}$ with the properties:

$$i \notin N_i$$

$$i \in N_j \Leftrightarrow j \in N_i$$

its *cliques* consist of either single sites or subsets of sites such that any two belonging to the same clique are neighbors of each other. With this definition, the energy may be written as:

$$U(f) = \sum_C V_C(f) \quad (7)$$

where C ranges over all the cliques of the neighborhood system, and each “potential function” V_C depends only on $\{f_i, i \in C\}$.

A random field F whose probability distribution is given by Equations 5 and 6 is called a *Markov random field* on L (Geman and Geman, 1984; Chellapa and Jain, 1993; Li, 2001). From (4), (5) and (7), one can see that the posterior distribution is also of the form (6), but now the energy includes the data term:

$$U(f) = \sum_{i \in L} -\Phi_i(f, g) + \sum_C V_C(f)$$

so that the Maximum a Posteriori (MAP) optimal estimator for f is obtained by minimizing this function.

The potential functions represent the “user interface” of the model, since through them one may specify the desired characteristics of the sample fields. Although they may be arbitrarily specified, there are four basic types that are generally used, depending on the characteristics of the desired reconstruction:

1. *Piecewise constant fields*: Here, each f_i may take only a finite (usually small) number of values. These fields are mostly used in segmentation problems, in which case it is often convenient that each f_i takes the form of a binary unit vector whose elements correspond to the indicator variables in Equation 3. The most widely used potential is the generalized Ising potential for cliques of size 2:

$$V_C(f_i, f_j) = -\beta, \text{ if } f_i = f_j \\ = \beta, \text{ otherwise}$$

2. *Globally smooth fields*: This case corresponds to standard regularization; the potentials are obtained as the squares of finite difference approximations of differential operators. For first-order differences, one obtains the “membrane” model:

$$V_C(f_i, f_j) = (f_i - f_j)^2 \quad (8)$$

where i and j denote a pair of nearest neighbor sites in the lattice.

If one adopts the observation model (Equation 2) and assumes that P_n is a zero-mean Gaussian distribution, the posterior energy becomes equivalent to the discretized functional of standard regularization, and its (unique) maximizer corresponds to the MAP estimator (see below).

3. *Piecewise smooth fields*: This is a very important and general case. There are two basic approaches for the construction of the potentials. In the first case, the discontinuities of the field are explicitly modeled by means of an auxiliary “line field” s (originally introduced by Geman and Geman, 1984), which is defined on a “dual” lattice whose sites are between each pair of (horizontal or vertical) neighboring sites of L ; s is thus indexed by a pair of indices corresponding to sites of L . Each line element s_{ij} may take values on the set $\{0, 1\}$, indicating, respectively, the absence or presence of a line (discontinuity) (in some models, s is allowed to take noninteger values in the interval $[0, 1]$ as well; see Geman and Reynolds, 1992; Black and Rangarajan, 1996).

The prior energy takes the form:

$$U(f, s) = \sum_{(i,j)} [(f_i - f_j)^2 s_{ij} + \Psi(s_{ij})] + \sum_D W_D(s) \quad (9)$$

where $\Psi(s_{ij})$ is a function that assigns a penalty for the introduction of a discontinuity between pixels i and j .

The line potentials $W_D(s)$ assign penalties to different local line configurations. They are summed over the cliques D of a neighborhood system defined on the dual lattice, and they are used to favor, for example, piecewise smooth lines, and to prevent the formation of smooth patches that are too thin or too small.

In the second case, the discontinuities are implicitly modeled by nonquadratic potentials $\rho(f_i - f_j)$, where ρ behaves like a quadratic function for small values of its argument but grows at a smaller rate as its argument becomes large. The derivatives of these potentials are related to influence functions of robust statistical estimators, and are therefore called *robust potentials*.

If the term $\sum_D W_D(s)$ is omitted, it is always possible to express Equation 9 in the form of a sum of robust potentials, simply by putting

$$\rho(f_i - f_j) = \inf_{s_{ij}} [(f_i - f_j)^2 s_{ij} + \Psi(s_{ij})]$$

where the right-hand side may be explicitly evaluated in many cases. If certain technical conditions on the ρ function are fulfilled, it is also possible to write a robust potential in the line field form (Charbonnier et al., 1997). Being able to go from one representation to the other, one may add spatial interaction terms to robust potentials, or use continuation methods that have been developed for robust potentials in the line field case (see Blake and Zisserman, 1987; Black and Rangarajan, 1996).

4. *Piecewise parametric models*: In this case the smooth patches are assumed to follow a parametric model with a relatively small number of parameters; for example, in the case of the reconstruction of the velocity field (optical flow) from a sequence of images, an affine model for the velocity of the form $f_i = A_i + b$ is often used, e.g., Black, Fleet, and Yakoob, 2000. In other cases, spline models with controlled stiffness are more appropriate (Marroquin et al., 2000). The problem here is that not only the parameters for each model have to be determined, but also the domain of validity of each model, i.e., a field of indicator variables, as in Equation 3. The prior constraints refer in this case to the spatial coherence of these domains, and may be enforced by Ising potentials.

Other examples of the application of these approaches to a variety of problems, as well as extensions and theoretical results, may be found in Chellapa and Jain (1993), Li (2001), and Marroquin et al. (2000, 2001).

Networks

Since the reconstruction is needed at the sites of the pixel lattice L , it is very natural to model the reconstructing network as a cellular automaton that consists of an array of processors or cells also located at the sites of L . The state of these processors at a given time t is denoted by $\xi^{(t)} = \{\xi_i^{(t)}, i \in L\}$. The interconnection pattern between processors is specified by the defined neighborhood system. The state of each processor changes from time to time with a rule that depends on its own state and that of its neighbors:

$$\xi_i^{(t+1)} = R(\xi_j^{(t)}, j \in N_i \cup \{i\})$$

Cellular automata (CA) may be deterministic (DCA) or stochastic (SCA), depending on the nature of the rule R .

Given this model for the architecture of a computational module, the important question is how to specify R , so that, in the deterministic case, the DCA has a fixed point and the reconstructed field f is obtained from it, and in the stochastic case, the automaton is regular and f is obtained from time averages of functions of its state.

In the case of globally smooth reconstructions, the energy function is usually convex, and the best estimator is obtained by min-

imizing this energy. The reconstructing networks are in this case equivalent to distributed iterative methods for matrix inversion (Bertsekas and Tsitsiklis, 1989). They may also be implemented analogically with pure resistor networks (see Marroquin, Mitter, and Poggio, 1987).

In the case of piecewise smooth potentials, when these are represented in the line field form and the term $\sum_D W_D(s)$ is not included, the energy function becomes quadratic in f for a given value of s , and therefore it may be minimized by the methods described in Bertsekas and Tsitsiklis (1989). On the other hand, if f is kept fixed, one may find the value of the s variables that minimizes U in closed form. By alternating these two steps, one gets an effective algorithm for the computation of the optimal estimator (Geman and Reynolds, 1992; Charbonnier et al., 1997). If the energy is represented in terms of robust potentials, often local descent schemes combined with continuation methods are most effective (Blake and Zisserman, 1987).

An important issue in all these cases is the determination of the parameters included in the energy function. In many cases these are hand-adjusted for a given class of images; it is better, however, to determine them automatically, as in Zhang (1993) or Chen, Chen, and Zhou, (2000).

Discussion

The key idea of this article is that Bayesian Estimation Theory, using prior MRF models, constitutes a general method that permits the formulation of many reconstruction problems in computational vision, so that they become equivalent to the minimization of an energy function that includes two terms: one that requires the solution to be consistent with the data (the likelihood term), and another that embodies prior constraints about its behavior. For the case of piecewise constant fields, however, the best estimator is not necessarily obtained by minimizing the posterior energy (i.e., the maximum a posteriori or MAP estimator). It has been shown that the estimator that maximizes the posterior marginal probabilities (the MPM estimator) has better behavior, particularly for low signal-to-noise ratios (Marroquin et al., 1987). In both cases, the cost for the exact computation of the optimal estimators is too high, so that approximations must be made. The most precise approximations are obtained with stochastic cellular automata, which mathematically correspond to regular Markov chains whose invariant measures correspond to the posterior distribution $P_{f|g}$. In this case, one can estimate the posterior marginals, by counting the number of times a given cell is in each state, from which the MPM estimator may be obtained. It is also possible to approximate the MAP estimator by introducing a "temperature" parameter that goes slowly to zero (a procedure known as simulated annealing; see SIMULATED ANNEALING AND BOLTZMANN MACHINES and Geman and Geman, 1984).

The main drawback of these stochastic methods is their computational complexity, since many iterations are needed to obtain accurate results. This is especially important in the case of the estimation of piecewise parametric models, since here the most effective procedures consist of two steps that are performed alternately in an iterative manner until convergence is achieved. These steps are:

1. Estimate the best segmentation (i.e., the f indicator variables in Equation 3), given the model parameters.
2. Estimate the model parameters given the segmentation.

An appropriate initialization step is also required. Instances of these procedures are found in Marroquin et al. (2000) and Black et al. (2000).

To perform step 1, it is necessary to have efficient estimators for piecewise constant fields. One way to obtain them is derived from mean-field (MF) theory of statistical physics. The MF-based estimation algorithm may be implemented by a deterministic cellular automaton with M layers, where each unit corresponds to a specific marginal probability. The update rule for each node involves the computation of the exponential of the sum of the local contributions of neighboring sites plus a normalization step (see Zhang, 1993).

A different approach is based on the idea of constructing a random field of discrete probability distributions using a Gauss-Markov model, so that the mean value of this field corresponds to the posterior marginal probabilities. Since this field is Gaussian, its mean value is found by the minimization of a quadratic form, which, because of the Markovian property, has a particularly simple structure. The network that computes the optimal estimator is shown in Figure 1. Note that, unlike the MF network, in this case there is no need either of exponentiation or of normalization; as a result, one can get better results at a fraction of the computational cost (see Marroquin et al., 2000, 2001).

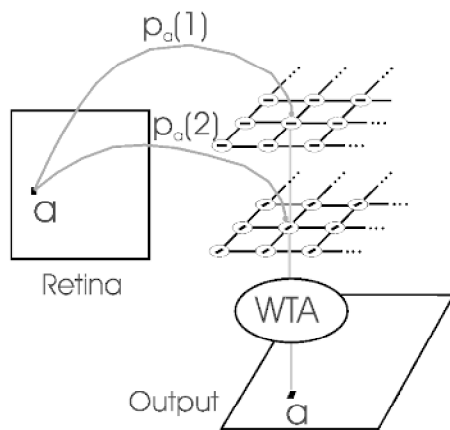


Figure 1. Network that computes the optimal estimator for a discrete-valued field. Each layer corresponds to a valid value for the field (in this example, orientation). Each cell computes the average of the state of its neighbors and the corresponding likelihood p_a obtained from the retina. A Winner Takes All mechanism outputs the most probably state at each time. Note that the layers are decoupled; they must, however, be synchronized for the system to work properly.

Road Map: Vision

Related Reading: Generalization and Regularization in Nonlinear Learning Systems; Hidden Markov Models; Statistical Mechanics of Neural Networks

References

- Bertsekas, D. P., and Tsitsiklis, J. N., 1989, *Parallel and Distributed Computation: Numerical Methods*, Englewood Cliffs, NJ: Prentice Hall. ♦
- Black, M. J., Fleet, D. J., and Yakoob, Y., 2000, Robustly estimating changes in image appearance, *Comput. Vision Image Understand.*, 78:8–31.
- Black, M. J., and Rangarajan, A., 1996, On the unification of line processes, outlier rejection, and robust statistics with applications in early vision, *Int. J. Comput. Vision*, 19:57–91. ♦
- Blake, A., and Zisserman, A., 1987, *Visual Reconstruction*, Cambridge, MA: MIT Press. ♦
- Charbonnier, P., Blanc-Feraud, L., Aubert, G., and Barlaud, M., 1997, Deterministic edge-preserving regularization in computer imaging, *IEEE Trans. Image Proc.*, 6:298–311. ♦
- Chellapa, R., and Jain, A., Eds., 1993, *Markov Random Fields: Theory and Practice*, Boston: Academic Press.
- Chen, W., Chen, M., and Zhou, J., 2000, Adaptively regularized constrained total least-squares image restoration, *IEEE Trans. Image Proc.*, 9:588–596.
- Geman, D., and Reynolds, G., 1992, Constrained restoration and the recovery of discontinuities, *IEEE Trans. Image Proc.*, 14:367–383.
- Geman, S., and Geman, D., 1984, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741.
- Li, S. Z., 2001, *Markov Random Field Modeling in Image Analysis*, New York: Springer-Verlag. ♦
- Marroquin, J. L., Botello, S., Calderon, F., and Vemuri, B. C., 2000, The MPM-MAP algorithm for image segmentation, *Proceedings of the 15th International Conference in Pattern Recognition (ICPR-2000)*, IEEE Computer Society, Barcelona, Spain, pp. 303–308.
- Marroquin, J. L., Mitter, S., and Poggio, T., 1987, Probabilistic solution of ill-posed problems in computational vision, *J. Am. Statist. Assoc.*, 82:76–89. ♦
- Marroquin, J. L., Velasco, F., Rivera, M., and Nakamura, M., 2001, Gauss-Markov measure field models for low-level vision, *IEEE Trans. Pattern Anal. Machine Intell.*, 23:337–348.
- Pomerleau, D. A., 1991, Efficient training of artificial neural networks for autonomous navigation, *Neural Computat.*, 3:88–97.
- Zhang, J., 1993, The mean field theory in EM procedures for blind Markov random field image restoration, *IEEE Trans. Image Proc.*, 2:27–40.

Programmable Neurocomputing Systems

Krste Asanović

Introduction

General-purpose personal computers and workstations are the most popular computing platforms used by researchers to simulate artificial neural network (ANN) algorithms. They provide a convenient and flexible programming environment, and advances in technology have been rapidly increasing their performance and reducing their cost. But large ANN simulations can still overwhelm the capabilities of even the most powerful workstations. For example, computations may require more than 10^{15} arithmetic operations and may operate on data sets containing several gigabytes of data (Ellis and Morgan, 1999).

Many neural net algorithms are highly parallelizable, allowing simulation speed to be improved by employing a network of work-

stations (NOW) (Anderson et al., 1995). Compared with more specialized hardware, a NOW can be an expensive solution. Fast network hardware increases the cost per node, and parallelization overheads reduce the performance per node.

For constrained application domains, fixed-function neural computing circuits can provide extremely high compute speeds at low cost, but they do not have the flexibility required to support experimentation with ANN algorithms.

To meet the need for high performance on large ANN simulations with flexible software control and reasonable cost/performance ratio, several groups have proposed and built *programmable neurocomputers*. Programmable neurocomputers (hereafter abbreviated to “neurocomputers”) attempt to maintain most of the flex-

ibility of a general-purpose computer system while improving the cost/performance ratio by specializing processors for neural computation.

This article reviews the most significant neurocomputer architectures, discusses their design and use, and concludes with predictions of future trends.

Survey of Neurocomputer Architectures

Programmable neurocomputers can be classified into four major categories. The first category uses commercial digital signal processors (DSP); the last three categories are based on custom-designed silicon.

Commercial DSP Arrays

Several neurocomputers have been built using arrays of commercial DSPs. Two notable examples are the RAP (Ring Array Processor), developed at the International Computer Science Institute (Morgan et al., 1992), and the MUSIC system, developed at the Swiss Federal Institute of Technology. (Muller et al., 1992). Both of these systems connect the DSPs in a unidirectional ring topology with communication circuitry built from field-programmable gate arrays (FPGAs). The RAP supports up to 40 Texas Instruments TMS320C30 floating-point DSPs, with a peak performance of 32 MFLOPS per node. The MUSIC system connects up to 45 Motorola DSP96002 floating-point DSPs, with a peak performance of 60 MFLOPS per node.

Both of these systems have distributed memories and are programmed using a Single Program Multiple Data (SPMD) model, in which all nodes run identical programs but operate on different portions of the data. A separate host computer manages the overall program flow and handles data input and output.

SIMD Processor Arrays

Another popular approach in neurocomputer design is a Single Instruction Multiple Data (SIMD) array of processors with some limited form of interprocessor interconnect. In these SIMD designs, a central sequencer broadcasts instructions that are executed simultaneously by all processors. The processors in a SIMD system can be much simpler than those in a SPMD system because they do not have to fetch and decode instructions. Also, SIMD processing elements do not require separate synchronizing operations because all processors work in lockstep.

Example SIMD neurocomputers include the CNAPS systems, from Adaptive Solutions (Hammerstrom, 1990), and the SNAP system, from HNC (Means and Lisenbee, 1991). The CNAPS system is built around large custom chips containing an array of 64 SIMD processing elements. Each processing element contains a fixed-point 16-bit \times 8-bit multiplier, a 24-bit accumulator, a set of 32 16-bit registers, and 4 Kbytes of local on-chip memory. The processing elements are connected by two 8-bit broadcast busses and a 2-bit interprocessor ring connect. The HNC system is built from SNAP chips, each of which contains four 32-bit floating-point multiply-add datapaths with access to local off-chip memory. Both of these systems allow multiple chips to be interconnected and controlled by the same central sequencer.

Systolic Processor Arrays

Several neurocomputers have been built around systolic processor arrays that perform the matrix operations at the heart of most neural algorithms. A systolic processor contains an array of interconnected pipelines through which operands flow in a regular rhythmic fashion.

The most advanced of these systems is the Synapse-1, constructed and sold by Siemens (Ramacher et al., 1991). This system is based on a custom systolic multiply-accumulate chip, the MA-16, which integrates 16 16-bit fixed-point multipliers. The Synapse-1 system employs multiple-chained MA-16 chips to give higher throughput. Large quantities of off-chip memory are provided, split into several disjoint memory areas. Operands must be located in the correct memory region before performing a given systolic matrix operation. Additional special-purpose fixed-point datapath hardware is provided to support ANN node activation functions not provided by the MA-16 chips, and Motorola 68040 CISC processors are used to perform all other operations. The entire system is controlled by a host workstation.

Another example of a systolic neural net engine is the Mantra machine, built at EPFL, Switzerland (Ienne and Viradez, 1994). Mantra can have up to 1600-bit serial processing elements arranged in a 40×40 systolic mesh. A commercial DSP acts as system controller.

Vector or SIMD Coprocessor

The machines discussed to this point all rely on some form of off-chip control sequencer or host computer to manage the matrix computations occurring on the parallel processor arrays. An alternative approach is to tightly integrate the control processor with the parallel execution units on the same die. Two examples of this type of design are the T0 vector microprocessor (Wawrzynek et al., 1996) and the L-Neuro 2.3 multi-DSP (Duranton, 1996).

The T0 vector microprocessor integrates an industry-standard MIPS-II 32-bit integer scalar RISC processor with a tightly coupled fixed-point vector coprocessor. The vector coprocessor contains a central vector register file with 16 vector registers each holding 32 elements of 32 bits, two vector arithmetic units, and a vector memory unit. The two vector arithmetic units each contain eight parallel pipelines and can produce up to eight 32-bit results per clock cycle. One of the arithmetic units contains 16-bit fixed-point multipliers, but otherwise the two pipelines are identical. The memory unit connects to off-chip memory over a 128-bit data bus. T0 has a single flat memory space equally accessible by the scalar unit and any element in the vector unit. T0 is similar in design to vector supercomputers (Russel, 1978), and scalar and vector instructions can be freely intermixed in the single instruction stream. The instruction set was designed to enable object-code compatibility with future higher-performance implementations.

The L-Neuro 2.3 design contains a 16-bit RISC controller plus an array of 12 DSP datapaths. The DSP datapaths are controlled via a writable microinstruction store indexed by the RISC controller macroinstructions. The wide microinstruction words allow pairs of DSP datapaths to perform different functions, and a flexible inter-DSP communication network is provided. The L-Neuro 2.3 supports an off-chip memory connection for each DSP datapath.

Neurocomputers Versus General-Purpose Processors

Neurocomputers are distinguished from general-purpose processors by their specialization for neural computations. If we examine the range of neurocomputers above, we find that three features specific to neural computation have been exploited to improve cost/performance:

- *Limited numeric precision.* Many neural algorithms can be coded to require only 8–16 bits of fixed-point arithmetic precision (Asanović and Morgan, 1991). The reduced precision allows reductions in the area required for arithmetic circuits, particularly multipliers, and also reduces the bandwidth required to transfer operands.

- *Data parallelism.* Most neural algorithms are inherently highly data parallel, where the same operation is performed across large arrays of data. Data parallelism is the simplest form of parallelism to exploit because a single block of control hardware can be shared over many datapaths.
- *Restricted communication patterns.* Broadcast buses or unidirectional rings are sufficient to support parallel execution of many common neural network algorithms. These simplified communication networks reduce the cost of interconnecting large numbers of parallel processing elements.

All neurocomputers aim to achieve high performance on the matrix computations at the heart of many neural algorithms by exploiting these features. But real-world neural net programs require operations other than these basic matrix operations. For example, an ANN training run may require significant disk I/O to retrieve training patterns and to checkpoint trained weights. Also, the training patterns may need preprocessing before being presented to the network, and the network outputs may require postprocessing to obtain results. If the neurocomputer is too slow at performing these other nonmatrix tasks, then the overall system performance will be dominated by these nonneural components. This result is well known in general-purpose computing as Amdahl's law (Amdahl, 1967):

$$S = \frac{1}{(1 - f) + f/E}$$

where E is the factor by which performance is improved by some new technique, f is the fraction of the program execution time for which the new technique is applicable, and S is the resulting overall speedup. For example, if 90% of the execution time of a computation is taken by matrix arithmetic, then even an infinitely fast matrix computation engine will never achieve an overall speedup greater than 10. Some neural computer designs have exacerbated this problem by imposing a *slowdown* for nonneural computations by requiring slow communication to a remote host processor to implement the required functionality. Ideally, fast general-purpose computing should be tightly integrated with the special-purpose processing units.

Another related issue is that the existing neurocomputers have been primarily designed to accelerate neural algorithms with dense connectivity (e.g., backpropagation), which can be expressed using dense matrix-vector operations. However, researchers are also interested in algorithms that explore sparse connectivity and sparser activation. These require fast scatter/gather memory operations and support for rapid irregular communications.

Flexible software support is perhaps the most important aspect of a successful neurocomputer design. Most neurocomputers are intended to be programmed using libraries of optimized matrix-vector functions. This approach is adequate for the computation portion of the code provided the libraries are extensive and are easy to compose in arbitrary ways. In particular, difficulties can arise if the libraries place constraints on the location of operands when the machine has multiple distributed memories.

To support experimentation with new ANN models, it is important to provide tools to allow users to extend the libraries to provide missing functionality. Ideally, this would consist of an optimizing high-level language compiler, but in practice, assembler or micro-code programming is usually required. Some of the neurocomputers have extremely complicated microarchitectures that are difficult to program efficiently at this low level. The features that complicate the task of the low-level programmer include:

- *Multiple distributed memories.* These require the programmer to carefully position data and to manage movement of data between memories.

- *Deep exposed execution pipelines.* These require the programmer to explicitly schedule operations occurring over many clock cycles.
- *Multiple levels of control flow.* Some systems have several levels of control flow (e.g., controller macroinstructions, microcontroller microinstructions, nanocontroller nanoinstructions) that must be jointly scheduled for peak performance.

Architectures that expose many details of an implementation to a programmer incur a significant programming overhead. If the same programmer-visible architecture is not preserved in subsequent machines, the software investment in low-level library code cannot be carried forward to new technology.

It is also important to provide libraries for I/O functions as well as computation, as often the amount of code required to manage data input and output dwarfs that required for matrix computation.

Discussion

The development of programmable neurocomputers was based on the premise that neurocomputing was significantly different from general-purpose computing and thus that a new type of computer architecture was warranted. But the preceding sections outlined many concerns shared with conventional computer architectures, namely, the need for flexible software development, high-performance library code, reasonable performance on arbitrary code, and fast I/O.

The primary distinguishing characteristics, namely, limited numeric precision and large-scale data parallelism, are features not only of ANN algorithms but also of many other algorithms in the areas of digital signal processing and multimedia. In recent years, many general-purpose microprocessors have added multimedia processing extensions (Lee and Smith, 1996) that provide support for data-parallel fixed-point processing. Typically, these multimedia extensions partition an existing 64-bit-wide datapath into a short vector of lower-precision subword components, e.g., 4×16 -bit values, with new instructions that operate on all subword components simultaneously, e.g., adding two vectors of 4×16 -bit operands to produce a vector of 4×16 -bit results. Microprocessors with multimedia extensions are very similar to the vector or SIMD coprocessor-based neurocomputer architectures, and share the same advantage of a tightly coupled general-purpose scalar unit.

Although the multimedia extensions implemented to date provide only a limited boost to the performance of general-purpose processors on fixed-point matrix code, they signal an intent by commercial microprocessor manufacturers to perform well on these types of code. As commercial design teams incorporate multimedia-style kernels into the workloads they consider during the design of new microprocessors, we can expect performance to increase rapidly also for ANN algorithms. The continuing tremendous investment placed in high-volume microprocessors ensures that these devices will use the most advanced fabrication technologies and the most aggressive circuit design styles yielding the highest clock rates. Given these trends, there will be greatly reduced interest in future special-purpose neurocomputers.

In attempting to optimize microprocessors for these multimedia codes, microprocessor architects will face many of the same challenges that confronted neurocomputer architects. Perhaps the greatest limitation on performance of highly data-parallel codes is sustainable memory bandwidth. Sustaining high bandwidth to off-chip memory requires both high raw memory bandwidth and the ability to tolerate long memory latencies by overlapping many concurrent memory requests. New off-chip memory architectures and packaging techniques will help improve raw memory bandwidths, but significant improvements in on-chip processor architecture will be required to provide sufficient parallelism to tolerate large off-chip

memory latencies. The current multimedia extensions provide only very limited data parallelism of four or eight elements at a time. It is likely that future designs will exploit much longer vectors to increase the level of parallelism supported without incurring additional instruction bandwidth costs. In addition, although current microprocessor multimedia extensions have no support for scatter/gather operations, it is likely that these will eventually be added to accelerate the large set of applications that rely on sparse matrix calculations.

A promising future direction is to integrate processor and main memory together on the same die, as in the Berkeley IRAM project (Kozyrakakis et al., 1997). The IRAM project is placing a vector processor similar to T0 on the same die as a large DRAM-based main memory. The vector processor is a simple hardware scheme for controlling a large degree of parallelism, while the on-chip memory both reduces latencies and dramatically increases available memory bandwidths. The vector unit provides fast scatter/gather operations from multiple on-chip memory banks. The combination should provide high sustained performance at low cost for data-parallel codes, including both dense and sparse ANN algorithms.

Road Map: Implementation and Analysis

Background: Digital VLSI for Neural Networks

Related Reading: Neuromorphic VLSI Circuits and Systems; Neurosimulation: Tools and Resources

References

- Amdahl, G. M., 1967, Validity of the single processor approach to achieving large scale computing capabilities, in *AFIPS Conference Proceedings*, Reston, VA: AFIPS Press, pp. 483–485. ♦
- Anderson, T. E., Culler, D. E., Patterson, D. A., and the NOW Team, 1995, A case for NOW (networks of workstations), *IEEE Micro.*, 15(1):54–64.
- Asanović, K., and Morgan, N., 1991, Experimental determination of precision requirements for back-propagation training of artificial neural networks, in *Proceedings of the 2nd International Conference on Microelectronics for Neural Networks*, Munich: Kyriall & Method Verlag.
- Durantoni, M., 1996, Image processing by neural networks, *IEEE Micro.*, 16(5):12–19.
- Ellis, D., and Morgan, N., 1999, Size matters: An empirical study of neural network training for large vocabulary continuous speech recognition, in *Proceedings of an International Conference on Acoustics, Speech, and Signal Processing*, Piscataway, NJ: IEEE Press.
- Hammerstrom, D., 1990, A VLSI architecture for high-performance, low-cost, on-chip learning, in *Proceedings of an International Joint Conference on Neural Networks*, Piscataway, NJ: IEEE Press, pp. II-537–II-543.
- Inenne, P., and Viredaz, M. A., 1994, Implementation of Kohonen's self-organizing maps on MANTRA-I, in *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, IEEE Computer Society Press, pp. 273–279.
- Kozyrakakis, C., Perissakis, S., Patterson, D., Anderson, T., Asanović, K., Cardwell, N., Fromm, R., Golbus, J., Gribstad, B., Keeton, K., Thomas, R., Treuhart, N., and Yelick, K., 1997, Scalable processors in the billion-transistor era: IRAM, *IEEE Comput.*, 30(9):75–78.
- Lee, R. B., and Smith, M. D., 1996, Special issue on media processing, *IEEE Micro.*, 16(4):6–9.
- Means, R., and Lisenbee, L., 1991, Extensible linear floating-point SIMD neurocomputer array processor, in *Proceedings of the International Joint Conference on Neural Networks*, Piscataway, NJ: IEEE Press, pp. 587–592.
- Morgan, N., Beck, J., Kohn, P., Bilmes, J., Allman, E., and Beer, J., 1992, The Ring Array Processor (RAP): A multiprocessing peripheral for connectionist applications, *J. Parallel Distrib. Comput.*, 14:248–259. ♦
- Muller, U. A., Baumie, B., Kohler, P., Gunzinger, A., and Guggenbuhl, W., 1992, Achieving supercomputer performance for neural net simulation with an array of digital signal processors, *IEEE Micro.*, 12(5):55–64. ♦
- Ramacher, U., Beichter, J., Raab, W., Anlauf, J., Bruls, N., Hachmann, M., and Wesseling, M., 1991, Design of a 1st generation neurocomputer, in *VLSI Design of Neural Networks*, Boston: Kluwer.
- Russel, R. M., 1978, The CRAY-1 computer system, *Commun. ACM*, 21:63–72.
- Wawrzynek, J., Asanović, K., Kingsbury, B., Beck, J., Johnson, D., and Morgan, N., 1996, Spert-II: A vector microprocessor system, *IEEE Comput.*, 29(3):79–86. ♦

Prosthetics, Motor Control

Gerald E. Loeb and Ning Lan

Introduction

This article deals with the subset of neural prosthetic interfaces that employ electrical stimulation to alter the function of motor systems, either directly or indirectly. The general biophysical considerations and technology are described in PROSTHETICS, NEURAL (q.v.).

Clinical Applications

Therapeutic Electrical Stimulation

Therapeutic electrical stimulation (TES) is an electrically produced exercise in which the beneficial effect occurs primarily off-line as a result of trophic effects on muscles and perhaps the central nervous system (CNS). One simple example is periodic exercise of the shoulder muscles to prevent disuse atrophy after a stroke (Faghri et al., 1994), which otherwise often results in chronically painful subluxation of the joint. TES effects have also been used to reduce spasticity following spinal cord injury (Stefanovska et al., 1989), presumably by downregulating the gain of hyperactive spinal reflex circuits. TES systems are relatively simple to implement because the patient chooses when and where to administer

the treatment and does not require any immediate effects from the stimulation. Stimulation programs are usually devised by the caregiver, but some parameters may be adjusted manually by the patient during self-treatment sessions.

Neuromodulatory Stimulation

Neuromodulatory stimulation (NMS) involves preprogrammed stimulation that directly triggers or modulates a function without ongoing control or feedback from the patient. Perhaps the oldest clinically successful neural prosthesis is phrenic nerve pacing to provide respiration in patients with central hypoventilation (Glenn and Phelps, 1985). More recently, sacral nerve stimulation has been used successfully to empty the bladder (Brindley and Rushton, 1990) and to reduce detrusor spasticity in patients with urge incontinence (Dijkema et al., 1993). NMS systems must be portable and reliable, but they function mostly autonomously.

Functional Electrical Stimulation

Functional electrical stimulation (FES) involves precisely controlled muscle contractions that produce specific movements re-

quired by the patient to perform a task. Much motor prosthetic research has been aimed toward permitting paraplegic patients to walk, a high-risk, high-energy activity that requires sophisticated interactions among the patient's immediate intentions, the pattern of stimulation applied to multiple muscles, and the ongoing movement elicited in the limbs. There have been some laboratory demonstrations of relatively complex but still crude systems that permit slow locomotor progress, but none is yet available clinically. The WalkAide is an FDA-approved (but not widely available) prosthesis that uses transcutaneous stimulation of the peroneal nerve to correct foot drop (Wieler et al., 1999). Research emphasis has shifted to FES-assisted grasp in quadriplegic patients, using residual motor function in the proximal and contralateral limb to control stimulation of finger muscles (Prochazka et al., 1997; Smith et al., 1998; Figure 1). Most of the subsystems described in the next section are in development to improve on-line control of FES.

Subsystems

Muscle Stimulation

Most research has been performed with skin surface electrodes, percutaneous wire electrodes, and implanted multichannel stimulators (Figure 1). The development of advanced stimulation techniques that require less extensive surgery (e.g., intramuscular

BIONs, Figure 2; Loeb et al., 2001) promises to improve the practicality of FES systems that require specific and reliable control of large numbers of individual muscles. Electrical activation of muscles by any route does not replicate the natural orderly recruitment of different types of muscle fibers, which gives rise to the high efficiency and fatigue resistance of normal force production. However, artificially stimulated muscles gradually undergo fiber-type conversions as a result of trophic effects that improve their aerobic capacity (Peckham, Mortimer, and Van der Meulen, 1973).

Active muscle has complex intrinsic mechanical properties that complicate attempts to develop feedforward control strategies based on predicting joint torques and movements. Muscle force depends nonlinearly on the number and frequency of firing of recruited muscle units and on the length and velocity of the muscle fibers. Many muscles have substantial amounts of series-elastic connective tissue (tendon and aponeurosis), which means that the length and velocity of the muscle fibers depend, in turn, on the amount of stretch that they produce in that connective tissue, as well as on the trajectory of the limb. While difficult to model mathematically, these complexities appear to play an important role in stabilizing the limb during rapid perturbations (Brown and Loeb, 2000) and in storing and releasing energy to improve the efficiency of cyclical movements such as walking. Many muscles cross more than one joint, further complicating their effects on the overall trajectory of movements.

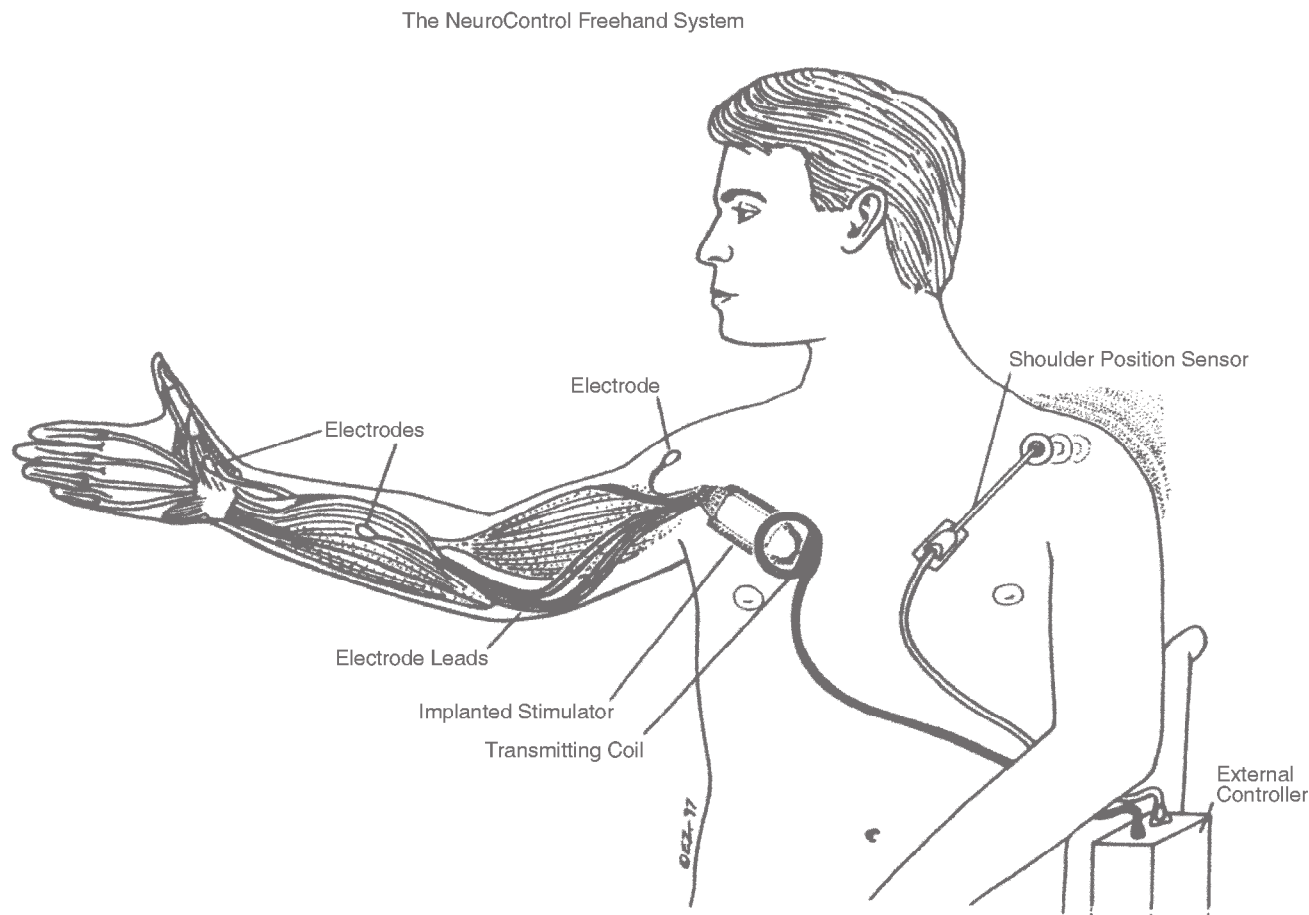


Figure 1. Freehand multichannel implanted stimulation system, approved by the U.S. Food and Drug Administration for control of grasp in spinal cord-injured patients. Voluntary shoulder motion detected by the external sensor triggers a stimulation control program that is transmitted to the im-

planted stimulator and routed to epimysial electrodes implanted near the nerve entry zones of various muscles operating the wrist and digits. (From Smith et al., 1998; photograph courtesy of the manufacturer, NeuroControl Corp., Cleveland, Ohio.)

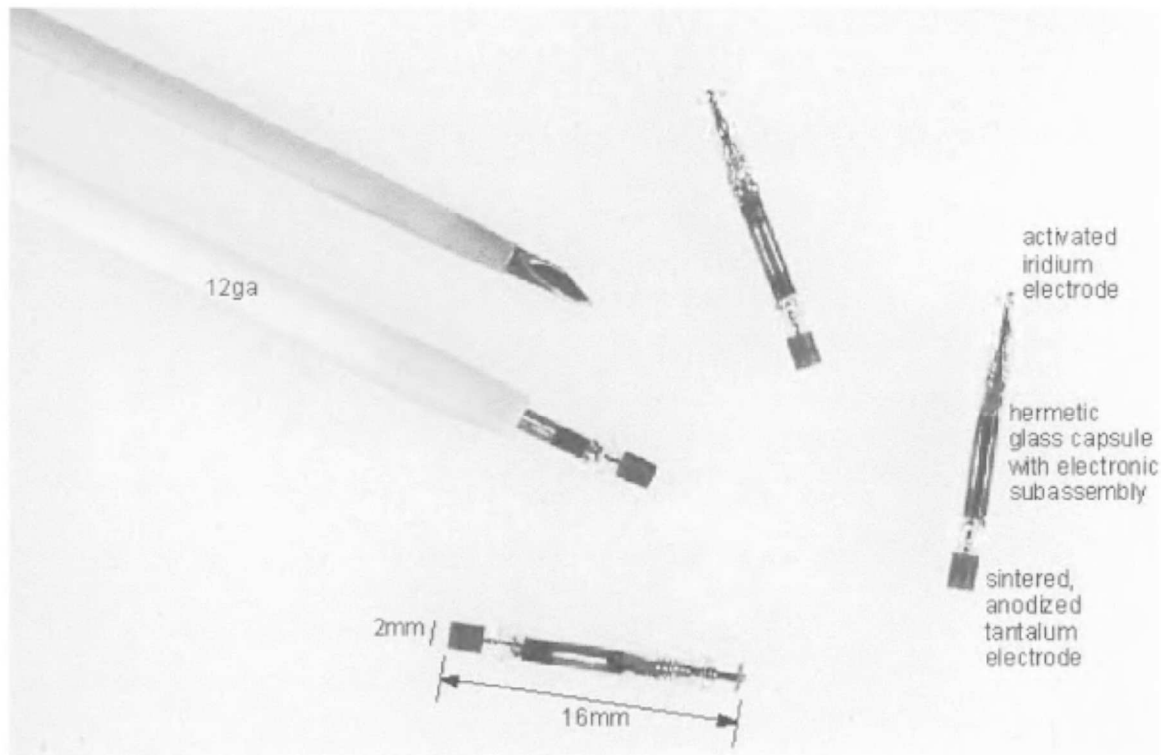


Figure 2. BION injectable microstimulators, now in clinical trials of TES to prevent shoulder subluxation due to muscle atrophy following stroke. Each implant (2 mm diameter \times 16 mm long) receives power and digital command signals from an amplitude-modulated 2 MHz magnetic field cre-

ated by an externally worn controller and transmitter coil. Each command specifies the address of one BION, the stimulus current (0.2–30 mA, in 30 steps), and the pulse width (4–514 μ s, in 512 steps).

Sensory Feedback

In biological sensorimotor control, an order of magnitude more neural information comes from intramuscular proprioceptors (muscle spindle and tendon organ afferents) than goes out to control motor units. When this information is absent, both animals and humans have a great deal of difficulty making stable and accurate movements. For tasks requiring manipulation of objects, information from cutaneous mechanoreceptors is even more important. Most rehabilitation therapists believe that an insensate hand is actually less useful than a paralyzed hand.

There are three general approaches to providing sensory feedback signals to implement biological-like control systems:

- Recording the proprioceptive and cutaneous signals that are still largely present in the peripheral nerves and dorsal root ganglia of patients with upper motor lesions. Microelectrode arrays have been implanted long term into these structures in animals (Loeb, Bak, and Duysens, 1977), but the technology is not yet robust enough for clinical use. Nerve cuff electrodes can record the aggregate activity of the large-diameter fibers in peripheral nerves, which can be useful in nerves with fairly homogeneous populations of afferents such as those innervating the digits (Haugland et al., 1999).
- Affixing various electromechanical sensors to the surface of the skin or to worn components of the prosthetic system, such as braces and gloves. While useful as research tools, such external appliances generally result in unacceptable problems related to mechanical maintenance, donning time, and physical appearance.
- Implanting artificial sensors into the sites where they are needed. In addition to the design problems inherent in protecting electro-

mechanical sensors from body fluids, such systems also require electrical leads or wireless communication to handle data and power requirements from large numbers of distributed transducers.

Sensorimotor Regulation

It has long been known that biological systems use a form of servocontrol. Mechanical perturbations sensed by mechanoreceptors give rise to specific reflex responses that tend to stabilize posture and force in the limb (e.g., the stretch reflex). More recently, spinal neurophysiologists have made substantial progress in unraveling the complexities of the spinal interneuronal circuitry and its role in coordinating descending commands with continuous sensory feedback.

The peripheral motor control system is substantially different in its organization from the servocontrollers used in robots. Spinal interneurons receive convergent input from many different modalities and origins of proprioceptive and cutaneous afferents, and they tend to project directly and indirectly to motor neurons controlling many different muscles and joints. Furthermore, most of the descending command signals from the brain that control limb movements terminate on these interneurons rather than directly on motor neurons. This has three important implications for the design of biological control systems (the ramifications for FES control remain unclear):

- The effects of command signals are essentially continuously modulated by the background activity from somatosensory afferents converging on the spinal interneurons.

- The brain can achieve a particular pattern of muscle activation via many different programs of interneuronal activation and inhibition, with each program resulting in potentially different patterns of reflex responses to perturbations.
- Descending pathways appear to be organized to produce various synergies of muscle recruitment and derecruitment rather than specific control of individual muscles.

FES control systems are starting to employ state-dependent logic to switch among different regulatory algorithms as different phases of the movement are detected from patterns in the signals from sensors (Kostov et al., 1995).

Control Systems

Controllers convert a given volitional command signal into a set of time-varying outputs from which the instantaneous intensities of muscle stimulation can be computed. Robotic engineering approaches that are based on complete knowledge of the sensorimotor plant have proved difficult to apply to the many degrees of freedom to be controlled and the number and complexity of the muscles to be stimulated for an FES task. An alternative approach may be to duplicate the adaptive control strategies of the CNS, which tends to perform much better in the low-precision but unpredictable demands of most activities of daily living. Biological sensorimotor systems appear to be organized in a hierarchical manner (Loeb, Brown, and Cheng, 1999), in which each layer of information processing plays a distinctive and important role in achieving goals with reasonable accuracy, stability, and energy efficiency. It remains to be elucidated how a motor goal is translated into a pattern of muscle activation through this hierarchical process, and what the organizing principles behind the formation of motor programs are. Given a sufficiently rich set of sensory feedback and informative commands, it may be possible to create interneuron-like networks for sensorimotor regulation and to use neural networks to learn how to control them to achieve similar goals for FES. For this strategy to be acceptable clinically, however, it will have to minimize the sort of trial-and-error sensorimotor learning that occupies so much of an infant's first few years of life.

Command Signals

Command signals convey the intent of the user to the control system of the prosthetic device. The control system senses and interprets the user's intent and computes an appropriate pattern of muscle stimulation. The controllers of all current FES systems and motorized artificial limbs obtain command signals from the myoelectrical activity or mechanical motion produced by those muscles that the subject can still control voluntarily. There is a general paradox in prosthetic motor control, however: the higher the level of the injury, the more degrees of freedom the prosthetic system must control, but the fewer the sources of voluntary command signals. For FES control of grasp, contralateral shoulder motion and residual wrist movement have been used to provide relatively simple commands (Smith et al., 1998). Subjects are able to produce reasonably high information rates on one channel by modulating rapidly among several distinguishable positions, probably because these muscles are still equipped with proprioceptive feedback. Other command sources, such as EMG and voice, tend to be slower and/or less precise. It remains to be seen whether systems can be designed to command the multiple simultaneous degrees of freedom involved in tasks such as coordinated reach and grasp.

An alternative approach to sensing residual voluntary muscle activity is to record command signals directly from the CNS above the level of the lesion. Attempts to record "brainwaves" via EEG and gross electrocortical electrodes have produced only very low

data rates (McFarland, McCane, and Wolpaw, 1998), probably because they reflect the aggregate activity of millions of neurons carrying very different signals simultaneously. Chronic unit recording techniques have been used for many years as a research tool to understand the role of the sensorimotor cortex in controlling natural motor behaviors; technologies feasible for clinical use are starting to emerge (Rousche and Normann, 1998). Limited functional use of such signals has been demonstrated in animals (Chapin et al., 1999) but the ultimate potential is likely to depend on the representation of complex movement in the brain, a subject that is still hotly debated by neurophysiologists. For example, it has been variously proposed that the primary motor cortex (Brodman's area 4) contains a representation of the desired position in space of the hand, the angles of the joints required to achieve a desired posture, the amount of force required from the individual muscles, and the states of the spinal interneurons. These have very different implications for the design of a controller required to respond to and interpret such command signals.

Conclusions

At one extreme, motor prostheses require only very simple exercise of one or a few muscles. At the other extreme, they require sophisticated bidirectional interfaces with the patient and on-line solution of problems in motor coordination that are normally solved by complex and poorly understood circuitry in the brain and spinal cord. FES applications provide particularly interesting challenges to our theoretical understanding of the normal roles of muscles, proprioceptors, spinal reflex pathways, and trajectory planning by the brain. They have also sparked attempts to reconcile traditional engineering approaches for the control of robotic manipulators with the very different but still obscure strategies for adaptive sensorimotor control in living organisms.

Road Maps: Applications; Mammalian Motor Control

Related Reading: Motor Control, Biological and Theoretical; Motoneuron Recruitment; Muscle Models; Prosthetics, Neural; Prosthetics, Sensory Systems

References

- Brindley, G. S., and Rushton, D. N., 1990, Long-term follow-up of patients with sacral anterior root stimulator implants, *Paraplegia*, 28:469–475.
- Brown, I. E., and Loeb, G. E., 2000, A reductionist approach to creating and using neuromusculoskeletal models, in *Neuro-Control of Posture and Movement* (J. Winters and P. Crago, Eds.), New York: Springer-Verlag, pp. 148–163.
- Chapin, J. K., Moxon, K. A., Markowitz, R. S., and Nicolelis, M. A. L., 1999, Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex, *Nature Neurosci.*, 2:664–670.
- Dijkema, H. E., Weil, E. H. J., Mijs, P. T., and Janknegt, R. A., 1993, Neuromodulation of sacral nerves for incontinence and voiding dysfunctions: Clinical results and complications, *Eur. Urol.*, 24:72–76.
- Faghri, P. D., Rodger, M. M., Glaser, R. M., Bors, J. G., Ho, C., and Akuthota, P., 1994, The effects of functional electrical stimulation on shoulder subluxation, arm function recovery, and shoulder pain in hemiplegic stroke patients, *Arch. Phys. Med. Rehabil.*, 75:73–79.
- Glenn, W. W. L., and Phelps, M. L., 1985, Diaphragm pacing by electrical stimulation of the phrenic nerve, *Neurosurgery*, 17:974–984.
- Haugland, M., Lickel, A., Haase, J., and Sinkjaer, T., 1999, Control of FES thumb force using slip information obtained from the cutaneous electro-neurogram in quadriplegic man, *IEEE Trans. Rehabil. Eng.*, 7(2):215–227.
- Kostov, A., Andrews, B. J., Popovic, D., Stein, R. B., and Armstrong, W. W., 1995, Machine learning in control of functional electrical stimulation systems for locomotion, *IEEE Trans. Biomed. Eng.*, 42:541–551.
- Loeb, G. E., Bak, M. J., and Duysens, J., 1977, Long-term unit recording from somatosensory neurons in the spinal ganglia of the freely walking cat, *Science*, 197:1192–1194.

- Loeb, G. E., Brown, I. E., and Cheng, E., 1999, A hierarchical foundation for models of sensorimotor control, *Exp. Brain Res.*, 126:1–18.
- Loeb, G. E., Peck, R. A., Moore, W. H., and Hood, K., 2001, BION system for distributed neural prosthetic interfaces, *Med. Eng. Phys.*, 23:9–18. ♦
- McFarland, D. J., McCane, L. M., and Wolpaw, J. R., 1998, EEG-based communication and control: Short-term role of feedback, *IEEE Trans. Rehabil. Eng.*, 6:7–11.
- Peckham, P. H., Mortimer, J. T., and Van der Meulen, J. P., 1973, Physiologic and metabolic changes in white muscle of cat following induced exercise, *Brain Res.*, 50:424–429.
- Prochazka, A., Gauthier, M., Wieler, M., and Kenwell, Z., 1997, The bionic glove: An electrical stimulator garment that provides controlled grasp and hand opening in quadriplegia, *Arch. Phys. Med. Rehabil.*, 78:608–614.
- Rousche, P. J., and Normann, R. A., 1998, Chronic recording capability of the Utah Intracortical Electrode Array in cat sensory cortex, *J. Neurosci. Methods*, 82:1–15.
- Smith, B., Tang, Z., Johnson, M. W., Pourmehdi, S., Gazdik, M. M., Buckett, J. R., and Peckham, P. H., 1998, Externally powered, multichannel, implantable stimulator-telemeter for control of paralyzed muscle, *IEEE Trans. Biomed. Eng.*, 45:463–475. ♦
- Stefanovska, A., Vodovnik, L., Gros, N., Rebersek, S., and Acimovic-Janezic, R., 1989, FES and spasticity, *IEEE Trans. Biomed. Eng.*, 36:738–745.
- Wieler, M., Stein, R. B., Ladouceur, M., Whittaker, M., Smith, A. W., Naaman, S., Barbeau, H., Bugaresti, J., and Aimone, E., 1999, Multicenter evaluation of electrical stimulation systems for walking, *Arch. Phys. Med. Rehabil.*, 80:495–500.

Prosthetics, Neural

Gerald E. Loeb

Introduction

This article provides an overview of the physical components that tend to be common to all neural prosthetic systems. It emphasizes the biophysical factors that constrain the sophistication of those interfaces. Specific applications to neural prosthetic systems for sensory replacement and motor control are covered in PROSTHETICS, SENSORY SYSTEMS (q.v.) and PROSTHETICS, MOTOR CONTROL (q.v.), respectively. Electrical stimulation of the nervous system is also being used to treat other disorders; examples include spinal cord stimulation to control pain and basal ganglia stimulation to control parkinsonian dyskinesias.

Electroneural Interfaces

Two types of physical system are known to be capable of real-time information processing: electronic circuits, in which information is carried by the flow of electrons in metal conductors, and neural circuits, in which information is carried by ions in water. Much contemporary research in computational neurobiology is concerned with discovering or exploiting common principles of information processing in these two systems. Thus, it is natural that real-time interfaces between these systems have been developed so that electronic instrumentation can be used to study neural systems. Neural prosthetics are clinical applications of neural control interfaces whereby information may be exchanged between neural and electronic circuits. Their technology to date has been derived largely from cardiac pacemakers, which themselves have evolved from the fixed-rate, single-channel stimulators of the 1950s to become programmable and adaptive systems equipped with sensors and sophisticated data processing.

In principle, information could be transferred into and out of the nervous system by any of several means, including chemical, magnetic, optical, and ultrasonic. In practice, neural prostheses require temporospatial resolution and physical portability, which have only been achieved with the types of electrical signals that are familiar to most electrophysiologists. Thus, the future of neural prosthetics depends heavily on the well-understood biophysical properties of excitable membranes and on the development of technology that can approach physical limits that are readily predictable from those properties.

In addition to the obvious goal of restoring function to patients with disabilities, the field of neural prosthetics offers important opportunities for pure research:

- The clinical and commercial value of neural prostheses justifies the development of technology that is also useful in basic research.
- The implantation of sophisticated neural control interfaces in sentient observers creates unique opportunities for a new class of psychophysical research into neural computation.
- The development of functional replacement parts for the nervous system forces researchers to examine and test theories of neural computing more rigorously than they might do otherwise.
- The development of neural prosthetic controllers that can deal successfully with the exigencies of daily life will almost certainly require advancement of principles and methods for neural networks and other forms of adaptive control.

Stimulation

Most neural prosthetic devices operate by injecting electrical current into the extracellular fluids surrounding excitable neurons in order to elicit action potentials in those neurons. Action potentials so elicited are indistinguishable from those that arise through the natural mechanisms of sensory transduction or synaptic input. When these action potentials arrive at their destinations, the receiving cells respond to and interpret the signals as if they arose from naturally occurring neural activity. Thus, the goal of the neural prosthetic device is to recreate the temporospatial pattern of activity that would have occurred normally during the particular function that is being replaced or augmented prosthetically (see PROSTHETICS, SENSORY SYSTEMS, for a discussion of these factors in cochlear implants).

Biophysics. Topologically, a neuron in its resting state is essentially a charged spherical capacitor with elongated deformations comprising its axon and dendrites. The cell membrane is the dielectric, the ionic solutions on either side of the membrane constitute the plates, and differences in the concentrations of ions on each side generate the charging potential of about -70 mV (inside versus outside). In order to generate an action potential, the membrane capacitance must be discharged by about 15 mV in a small region. This results in a brief sequence of openings and closings of sodium and potassium channels in the membrane, which results in the flow of the action current. The action current depolarizes and then repolarizes adjacent regions of the cell membrane, giving rise to the propagating wave of activity known as an action potential or spike.

The process of evoking an action potential through extracellular stimulation is somewhat counterintuitive. In order to depolarize a capacitor, it is necessary to pass charge across the dielectric, i.e., into the cell body. However, neither the source nor the sink electrode of the stimulator is actually inside the cell. Instead, the stimulator creates a voltage gradient in the tissues surrounding the target cell. This gradient induces charge to flow across the cell membrane by capacitive conductance in response to the rate of change of the voltage gradient, i.e., dV/dt . The amount and extent of the depolarization so produced depend on the intensity and time course of the pulse of stimulation current, its propagation through the various conductances in the tissues through which it diffuses, and the physical dimensions and consequent electrical properties of the excitable target cell (Ranck, 1975).

The most important physical dimension of highly elongated neurons is their diameter, which affects the ratio of membrane surface area (which determines capacitance) to axonal cross-sectional area (which determines resistance to current spread inside the cell). The presence, thickness, and disposition of myelin are also important because myelin greatly reduces the capacitance that must be discharged in order to reach threshold depolarization. It should also be remembered that electrical current must form a complete circuit, so that any capacitive stimulation current that enters a cell at one point, depolarizing its cell membrane locally, must be balanced by equal current leaving the cell and causing some degree of local hyperpolarization in other regions.

In order to predict accurately the effects of stimulating a complex structure such as a part of the cerebral cortex or a muscle with embedded sensory and motor axons, it is necessary to have a great deal of quantitative information about the neural architecture and the disposition of the stimulating electrodes. There are some useful rules of thumb, however, that cover most of the important phenomena:

- The most important consequence of a stimulation pulse is the steepness of the extracellular field gradient that it produces in the vicinity of the target neurons. Small electrodes positioned close to excitable processes are most effective.
- The important stimulus variable is the charge of a pulse, which is current times duration. Voltage is not important, as most of the voltage tends to be dissipated across the metal-electrolyte interface (see below) rather than contributing to the voltage gradient in the tissue surrounding the neurons.
- Stimulation pulses are most efficient when they have a duration that is somewhat shorter than the membrane time constant of the target cells, which is usually on the order of 100–200 μ s for myelinated axons and 500–1,000 μ s for unmyelinated axons and cell bodies.
- The first recruited elements tend to be the largest, most elongated, and most myelinated elements, namely large-diameter myelinated axons and large cell bodies attached to myelinated axons.
- Most body tissues (including bone and scar tissue) are sufficiently conductive that they tend, in aggregate, to act as volume conductors in which stimulus current density steadily decreases with distance from the electrode.
- Stimulation electrodes must be used in pairs (source and sink), but each contact tends to function as an independent monopolar electrode unless the two contacts are positioned closer together than to the target neurons.

Electrochemistry. The rise of safe and effective neural prosthetic devices over the past 30 years is a consequence of the gradual elucidation of the electrochemical processes involved in converting electrical current from flow of electrons in a metal conductor to flow of ions in an aqueous one. In order for this to occur without cumulative deterioration of the electrodes or damage to the tissues,

it is necessary that this be accomplished by entirely reversible chemical reactions. The typical reactions of electrolysis result in irreversible breakdown of water molecules into gases and acid or alkali solutions and shifts of the neutral valence of metals into positive-valence oxides with very different electrical, chemical, and biotoxic properties (reviewed by Loeb, McHardy, and Kelliher, 1982).

The most obvious fully reversible reaction is the charging and discharging of the capacitance between the metal electrode and the body fluids. Because the irreversible reactions of electrolysis all have minimal working voltages that must be exceeded before they occur (usually about ± 0.8 VDC), stimulating current can be passed into and out of the electrode safely as long as capacitive charging never reaches these working voltages. Thus, repeated brief pulses of electrical current can be applied safely, as long as an equal and opposite amount of charge flows in the opposite direction between each stimulating pulse.

The amount of charge that can be passed by this double-layer charging depends on the capacitance of this interface, which depends on the surface area of the electrode and the thickness of the effective dielectric boundary of the interface. A metal that forms little or no surface oxide, such as platinum, has a dielectric boundary thickness that depends on the thermodynamics of molecules bouncing off its surface. Metals that form stable nonconductive oxides, such as tantalum, can be anodized to build up their oxide thickness, reducing the capacitance but providing a barrier to inadvertent electrolytic reactions (Guyton and Hambrecht, 1974) and permitting them to sustain much higher voltages during stimulation pulses. Other reversible reactions available on some metal surfaces include absorption and desorption of hydrogen and oxygen. Iridium provides the highest charge density limit (about 3 mC/cm²) of any biocompatible electrode material because it can be “activated” by growing a multilayer surface oxide that is electrically conductive and porous to ions (Robblee, Lefko, and Brummer, 1983). Iridium exhibits a range of stable positive valences from about +3 to +4.8, so that each atom in the oxide layer can absorb or release about two electrons, with concomitant release or absorption of two hydroxyl ions.

Electrochemical considerations are particularly important to attempts to extend neural prosthetic technology to provide denser multichannel interfaces with the nervous system. In order to provide more independent channels of stimulation in a given neural pathway, it is necessary to make the electrodes smaller and position them closer to their target neurons (Loeb, Peck, and Martyniuk, 1995). Such a microelectrode produces sufficient current density to activate local neurons selectively, while minimizing the spread of stimulation current to adjacent sites under the control of other microelectrodes. Unfortunately, the surface area of such electrodes tends to decrease faster than the amount of charge required to activate local neurons, pushing electrode materials to their safe charge density limits.

Recording

Electrophysiological recordings of interest to the control of neural prosthetics range from the potentials generated by large populations of cells (such as those measured in EEG and EMG) to the action potentials generated by individual neurons. All of these signals are small-amplitude AC signals (typically less than 1 mV) that must be detected against a background of interfering signals from other bioelectrical sources and various sources of noise. In most parts of the central and peripheral nervous systems, the activity of adjacent neurons is often quite distinctive, making it necessary to record and distinguish the action potentials of single units to use them as command and control signals. This has been accomplished for brief periods of time in many research applications, but techniques re-

main to be developed for stable, long-term recordings from human patients.

The action potentials generated by individual neurons are produced by action currents of 1–10 nA lasting 0.2–2 ms, depending on the size and myelination of the cell. As in the case of the currents produced by electrical stimulation, the current density and the resulting potential gradient in the surrounding tissues drop rapidly with distance from the current source (Rall, 1962). Microelectrodes usually must be within 100 μm of a neuron to record a usable action potential. In order to be positioned that close to a neuron, such a microelectrode must be physically small, with a small surface area. For a metal microelectrode, double-layer charging of the metal-electrolyte interface provides the mechanism for converting a biopotential from ion fluxes in water to electron motion in a metal conductor. The small surface area of the exposed metal surface provides only a small capacitance, resulting in a relatively high impedance in the frequency band of the action potential (typically 100–1,000 k Ω at 0.5–5 kHz). The resistivity of the saline in the immediate vicinity of the microelectrode also presents a substantial impedance. High impedance is associated with high thermal noise, which adds to and obscures any biopotentials to be recorded.

Usually microelectrodes pick up signals from several adjacent neurons, which may need to be discriminated based on small differences in their waveforms. Even relatively low noise levels may degrade the reliability of such discrimination. Small movements of the microelectrode with respect to the neurons are likely to distort the relative amplitude and shape of the single unit potentials or change the sampled population entirely.

Systems Hardware

Power and Data Management

In order to improve the sophistication and capabilities of neural prosthetic interfaces, larger numbers of stimulating and recording channels must be positioned closer to their neural targets. This raises the problem of how to transmit more data to and from arrays of small electrodes located in delicate neural tissues.

One approach is to combine many electrodes into an array that includes active electronic processing so that a large number of separate signals can be multiplexed onto one data connection (Najafi, Ji, and Wise, 1990). Stimulation pulses are usually relatively brief ($\sim 100 \mu\text{s}$) compared to their interpulse intervals (~ 10 – 100 ms), making it possible for a single stimulus channel to be multiplexed among many electrodes. Bioelectrical potentials are more difficult to multiplex because they usually have high concurrent bandwidths (1–10 kHz) on each channel, resulting in very high aggregate sampling rates.

Any active circuitry for multiplexing or demultiplexing requires DC power. Electrical leads and connectors carrying DC voltages are particularly difficult to insulate because even tiny amounts of saline leakage result in electrolytic corrosion (see below). The very low power consumption of some integrated circuit technologies has led to interest in the wireless transmission of power. Radio-frequency (RF) inductive coupling is now used routinely in cochlear implants and has been developed for injectable muscle stimulators (Troyk and Schwan, 1992). Infrared transmission and photoelectric conversion may also be possible over short distances.

Packaging

Active microelectronic circuitry is generally contained within a hermetic enclosure to protect it from moisture. This adds greatly to the physical bulk of the circuitry, particularly if large numbers of input-output channels must be routed through feedthroughs and

connectors incorporated into the package. The design of the package and the selection of hermetic materials (metals, ceramics, and glasses) are likely to be further complicated by the need to transmit RF or infrared energy to power and control the electronics. Embedding in epoxy, silicone, and other nonhermetic polymers was commonplace in the early cardiac pacemaker industry, but it is difficult to perform reliably on complex circuits. There is much interest in passivating monolithic integrated circuits so that they can be implanted directly into the nervous system with few or no attached leads.

Nonhermetic encapsulation and passivation depends on adhesion of the coating material to the substrate electronic components rather than impermeability. Water in the vapor phase tends to diffuse through all polymeric materials, but it does not cause electronic problems until it condenses onto the circuitry itself. Once condensation occurs, the water vapor forms an ionic solution by dissolving surface contaminants or the materials themselves. This solution represents an osmotic attractant, pulling additional water vapor out of the surrounding polymer and pressurizing itself so that it dissects along the surface, eventually bridging electrical conductors and resulting in corrosion and circuit failure. Condensation on hydrophilic surfaces can be prevented only if there are no voids and there is sufficient adhesive force between the encapsulant and all substrate materials. Even trace surface contaminants tend to interfere with adhesion, which usually depends on electrostatic bonding (Donaldson, 1987).

An alternative approach to chemical adhesion for certain geometries is to use hydrostatic pressure. This has proved useful in preventing electrical shorting from condensed water within connectors that must be opened and closed in the body. The encapsulant is pressurized mechanically as the connector is closed so that its hydrostatic pressure exceeds the maximal osmotic pressure of salt solutions (typically 200–250 psi) (Loeb et al., 1983).

Conclusions

Serious attempts to build functional neural prostheses have been under way for about 35 years. Progress has been limited because it depended on concurrent developments in microelectronic and biomaterial technologies as well as on advances in understanding the neurophysiology of the functions to be restored. Once these thresholds have been passed for a given application (e.g., cochlear implants for the deaf; see PROSTHETICS, SENSORY SYSTEMS), dramatic functional restoration has been achieved, although the development of these complex and highly regulated medical devices remains much slower than that of comparable consumer electronics. As the armamentarium of applicable technology and basic neurophysiology enlarges, there should be a steady acceleration in the clinical application of neural prostheses.

Road Map: Applications

Related Reading: Brain-Computer Interfaces; Brain Signal Analysis; Prosthetics, Motor Control; Prosthetics, Sensory Systems

References

- Donaldson, P. E., 1987, Twenty years of neurological prosthesis-making, *J. Biomed. Eng.*, 9:291–298. ♦
- Guyton, D. L., and Hambrecht, F. T., 1974, Theory and design of capacitor electrodes for chronic stimulation, *Med. Biol. Eng.*, 12:613–619.
- Loeb, G. E., Byers, C. L., Rebscher, S. J., Casey, D. E., Fong, M. M., Schindler, R. A., Gray, R. F., and Merzenich, M. M., 1983, The design and fabrication of an experimental cochlear prosthesis, *Med. Biol. Eng. Comput.*, 21:241–254.

- Loeb, G. E., McHardy, J., and Kelliher, E. M., 1982, Neural prosthesis, in *Biocompatibility in Clinical Practice*, vol. II (D. F. Williams, Ed.), Boca Raton: CRC Press, pp. 123–149. ♦
- Loeb, G. E., Peck, R. A., and Martyniuk, J., 1995, Toward the ultimate metal microelectrode, *J. Neurosci. Methods*, 63:175–183.
- Najafi, K., Ji, J., and Wise, K. D., 1990, Scaling limitations of silicon multichannel recording probes, *IEEE Trans. Biomed. Eng.*, 37:1–11.
- Rall, W., 1962, Electrophysiology of a dendritic neuron model, *Biophys. J.*, 2:145–167.
- Ranck, J. B., Jr., 1975, Which elements are excited in electrical stimulation of mammalian central nervous system: A review, *Brain Res.*, 98:417–440. ♦
- Robblee, L. S., Lefko, J. L., and Brummer, S. B., 1983, Activated Ir: An electrode suitable for reversible charge injection in saline solution, *J. Electrochem. Soc.*, 130:731–733.
- Troyk, P. R., and Schwan, M. A., 1992, Closed-loop class E transcutaneous power and data link for microimplants, *IEEE Trans. Biomed. Eng.*, 39:589–599.

Prosthetics, Sensory Systems

Gerald E. Loeb and Blake Wilson

Introduction

This article concerns sensory prostheses, in which information is collected by electronic sensors and delivered directly to the nervous system by electrical stimulation of pathways in or leading to the parts of the brain that normally process a given sensory modality. In principle, all of the senses could be replaced or even augmented by such technology. In practice, only some sensory modalities seem amenable to currently available approaches; the status for each sense is summarized below:

- Hearing—widespread clinical success with the use of cochlear implants over the past decade.
- Vision—long-standing goal, with a recent resurgence in preclinical research plus some pilot human experiments.
- Touch—some clinical research on peripheral restoration in conjunction with functional electrical stimulation (FES; see PROSTHETICS, MOTOR CONTROL).
- Proprioception—little research under way, despite eventual importance to FES.
- Balance—some theoretical potential and early-stage analysis of feasibility.
- Smell—some theoretical interest because of the clinical significance of anosmia in the elderly, but hampered by the complexity of the natural senses and the unavailability of prosthetic sensors.
- Taste—no research under way; little clinical interest.

The general problem in constructing and implementing sensory prostheses is to understand and emulate the relevant parts of the neural code used by normal sense organs to encode and transmit sensory information to the brain. In practice, this means identifying a surgically accessible site through which to apply a complex temporospatial pattern of electrical stimulation. The general biophysical and electronic considerations can be found in PROSTHETICS, NEURAL. This article describes current research on auditory and visual prostheses.

Cochlear Prostheses

Current Technology

Cochlear prostheses use direct electrical stimulation of auditory nerve cells to bypass absent or defective hair cells that normally transduce acoustic vibrations into neural activity. They are the most sophisticated and the most successful neural prostheses to date, and they are still evolving. The currently available devices generally use multicontact electrodes inserted into the scala tympani of the cochlea so that they can differentially activate auditory neurons that normally encode different pitches of sound (for a historical review,

see Loeb, 1990). A much smaller number of patients with bilateral degeneration of the auditory nerve have been treated with modest success by stimulation of the cochlear nucleus in the brainstem. In all currently available systems, an external, wearable control unit (Figure 1) determines a pattern of electrical stimulation in which the stimulus amplitude in each channel depends on the spectral content in the acoustic input and a previously stored map of auditory sensations that can be elicited by electrical stimulation of each channel in that patient. Many algorithms have been developed over the years, employing both analog and pulsatile electrical waveforms delivered sequentially and/or simultaneously to fixed or dynamically changing channels.

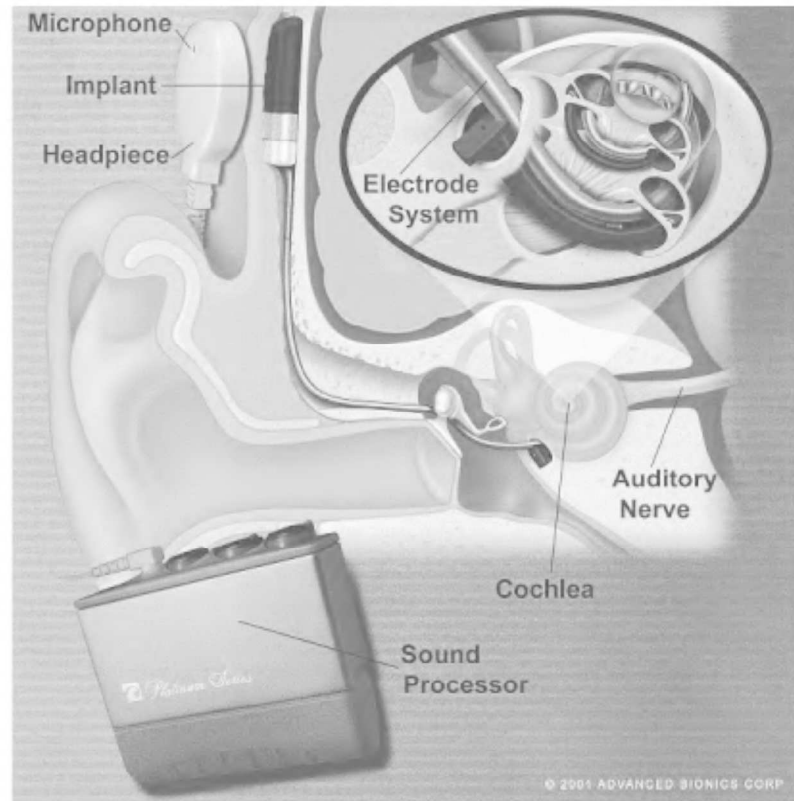
Clinical results with cochlear implants have improved steadily, to the point that they are the treatment of choice for most cases of severe to profound sensorineural hearing loss in both adults and children. Most patients with adult-onset deafness are devastated by the loss of social interactions and find it difficult to develop alternatives such as lip-reading and sign language. Cochlear implants provide essentially immediate restoration of awareness of sound in all patients and functional levels of speech recognition in the majority, suggesting that the goal of replicating the salient natural encoding of sound has been achieved in most but not all patients (see below).

For prelinguistically deafened children, the trend is to implant a cochlear prosthesis at an early age, generally under 6 and increasingly under 2 years old. Young children appear to benefit from the increased plasticity of the young nervous system and the ability to participate in conventional verbal educational programs (Svirsky et al., 2001). In general, they acquire language skills at the same rate as normal children, but they remain delayed by the preimplantation period. Individuals who have received no acoustic stimulation in the first few years of life appear to lose much of their ability to learn to process such information by adolescence (Busby et al., 1991), consistent with the notion of “critical periods” in the training of biological neural networks.

Current Research

Improved temporospatial representations of speech sounds. The replacement of 10,000 independent hair cell transducers with 8 to 20 sources of stimulation current necessarily results in distortions of both the temporal and spatial patterns of neural activity received by the brain. The spatial problem is further aggravated by the tendency of stimulus currents to spread radially from their source in the volume-conductive tissues of the cochlea, resulting in “cross-talk,” whereby a stimulus targeted to one spatial subgroup of spiral ganglion cells exerts modulatory effects on distant subgroups that are the target of another stimulation channel. The extent of this problem has been appreciated relatively recently through both clinical

Figure 1. Clarion cochlear prosthesis showing external (left foreground) and implanted (right) components. The implanted components include a 16-contact intracochlear electrode (wedged into the spiral shape of the first one-and-a-half turns of the scala tympani by the dark blue positioner; see insert) and hermetically encased electronics (labeled implant: 25 mm wide by 6 mm thick). The external sound processor contains patient-operated controls, a rechargeable battery, a microprocessor, and a digital signal processor, and connects to a headpiece with a microphone and the antenna that transmits power and data to the implant. Acoustic signals from the microphone are filtered and converted into 8 to 16 channels of stimulus waveforms; these are delivered by the electrode contacts to recruit tonotopically arranged subsets of the spiral ganglion cells that comprise the auditory nerve. (Photograph courtesy of the manufacturer, Advanced Bionics Corp., Valencia, Calif.)



cal studies (Lawson et al., 1996) and computer modeling (Frijns, Braire, and Grote, 2001). It is particularly severe for apical (low-frequency) sites. This has led to renewed development and testing of novel cochlear electrode arrays intended to position the contacts closer to the target spiral ganglion cells and reduce spread of stimulation currents.

The temporal distortions are more complex and their significance and amelioration less obvious. As the strength of an electrical stimulus is increased to represent increasing loudness, it recruits more and more auditory neurons, as would occur with an acoustic stimulus, but the electrically evoked activity tends to be much more highly synchronized. This results in a form of “beating” or “aliasing” among the repetition rates of the stimulation pulses (typically 400–800 pps), the modulation bandwidths of the acoustic information (typically 100–400 Hz), and the relative refractory periods of auditory neurons (probably 1–3 ms, corresponding to 300–1,000 Hz). The effect is most severe for low-frequency (apical) percepts, which are normally decoded from both temporal and spatial cues in the neural activity. It appears to be possible and useful to break up such beating by employing stimulation pulse rates that are far higher than those that can be followed by individual neurons (e.g., 1,500 up to 5,000 pps). This reduces beating and aliasing and results in a more randomized temporospatial representation, which many patients find to be subjectively less annoying and functionally more useful (Wilson et al., in press).

Combined electrical and acoustic stimulation in patients with residual hearing. There are many more patients with severe than with profound hearing loss, and most of these tend to have preferential preservation of low-frequency (apical) acoustic perception. This is the band in which cochlear implants tend to produce the greatest spatial and temporal distortions (see above). Clinical testing in a limited number of such subjects suggests that it is usually

possible to preserve acoustic hearing apical to an electrode array that has been inserted shallowly into the scala tympani. The simultaneous presentation of amplified low-frequency acoustic information together with a multichannel electrical representation of the higher frequencies produces substantial improvements in performance, particularly for complex tasks such as perception of speech in highly noisy environments (Wilson et al., in press).

Bilateral cochlear implants. Individuals with normal hearing use binaural cues to distinguish desirable signals from noise sources that are located at different positions. Differences in relative loudness and arrival time at the two ears are decoded in the auditory brainstem so that the cognitive centers of the cortex can focus on spectral information from a single source. At least some of the few patients who have received cochlear implants in both ears have experienced substantial improvements in speech perception in noisy environments (Wilson et al., in press). This has motivated additional research on methods to synchronize the stimulation of the corresponding channels in the two ears, which might lead to performance that would warrant the additional expense and invasiveness of two cochlear implants.

Psychophysical correlates of performance variability. The development and testing of cochlear implants has been plagued by large variability of results among patients with no distinguishing characteristics. This complicates the design and interpretation of studies comparing the performance of different devices and speech-processing strategies. It also makes it difficult to justify implantation in the many patients whose residual hearing provides function comparable to that obtained by the poorest implant recipients. Enhanced psychophysical tests enabled by more flexible stimulation systems have started to identify neurophysiological correlates of cochlear implant performance (Wilson et al., in press). This bodes

well for the development of speech-processing strategies to overcome these individual limitations and to fit them to the appropriate patients.

Fully implanted systems. Cochlear implants have been following a development track similar to that of hearing aids, their technological predecessors. Both started with relatively large and power-hungry circuitry that had to be worn on the body, including large, heavy, rechargeable batteries. Both used improvements in low-power integrated circuitry and battery technologies to miniaturize the sound-processing systems so they could be worn behind the ear (or even in the ear canal, in the case of hearing aids). Because one component of a cochlear prosthetic system must be surgically implanted, an obvious goal is to eliminate the external components entirely. This poses three major challenges that seem likely to be overcome within the next 2–3 years:

- The power consumed by the stimulation pulses themselves is substantial in a high-speed, multichannel implant, necessitating the development of high-performance batteries that can be recharged rapidly and frequently by inductive coupling of RF energy applied outside the body. More efficient electrodes closer to the spiral ganglion cells should also help.
- The microphone in present cochlear implant systems is usually located with the external headpiece that transmits power and data to the implanted electrodes. Novel technologies are in development for an implanted microphone that will function electrically and acoustically in a surgically suitable site.
- The dynamic range of electrical stimulation (from perceptual threshold to uncomfortably loud) is very narrow compared with that of acoustic hearing (6–20 dB versus 100 dB). Speech processors employ sophisticated digital algorithms for dynamic gain control and stimulus intensity mapping, but many patients find it necessary to make frequent adjustments to the manual loudness control on their externally worn speech processors. An alternative strategy is to have the implant monitor the electromyographic activity associated with the stapedius reflex and use it to make automatic adjustments of stimulation intensity. This is a protective reflex that comes on when the brain perceives the sound to be uncomfortably loud; it is intact in most cochlear implant recipients.

Visual Prostheses

As in the early days of auditory prostheses, there is not yet any general agreement on the most promising site to apply electrical stimulation to the visual pathways. Sites that have been considered include subretinal (microelectronic array of photoreceptors between the retina and the sclera), epiretinal (thin film electrode array on the vitreous surface of the retina), optic nerve (nerve cuff electrode), optic radiations (probes with multiple contacts inserted stereotactically), surface of striate cortex (arrays of contacts on the pial surface), and striate intracortical (see below). The obvious difference between auditory and visual prostheses is that auditory information requires a small number of channels with high data rates while the visual system requires a large number of channels with low data rates. This would seem to favor the two sites described below, which offer large, fairly flat surfaces on which to deploy retinotopically mapped electrode arrays.

Cortical Approach

Attempts to provide useful visual sensations in the blind by direct electrical stimulation of visual cerebral cortex began in 1966 (Brindley and Lewin, 1968). The initial devices used arrays of small electrodes (about 1 mm diameter) on the pial surface. Relatively

high stimulus currents (about 1 mA for a 200 μ s pulse) were required to produce sensations of light called phosphenes. Because of current spread by volume conduction, such stimulation presented to a single electrode presumably recruits neurons scattered over many adjacent cortical columns, but the surround inhibitory mechanisms actually result in a surprisingly small, well-formed dot of light. This seems to suggest that a complete, if coarse-grained, picture could be built up from a sufficient number of such phosphenes. The problem is that the processes responsible for the focusing operate quite slowly, so that stimulus trains presented concurrently but interleaved between even two such sites produce unpredictable, nonlinear interactions (Girvin, 1988). A useful image will require hundreds, if not thousands, of independently controllable phosphenes.

More recently, intracortical microelectrodes have been employed successfully to create similar phosphenes with stimulus currents (5–20 μ A) that would tend to recruit only a few neurons within the immediately vicinity of the electrode tip (Bak et al., 1990). When two sites spaced less than a millimeter apart are stimulated concurrently, their phosphenes seem to combine and fuse in a predictable and desirable manner. Silicon fabrication (Wise and Najafi, 1991; Normann et al., 1996) may make it feasible to build dense arrays of contacts and associated electronic circuitry that are safe to implant and operate continuously for long periods of time.

Retinal Approach

Recent improvements in low-power integrated circuitry and intraocular surgical techniques have sparked interest in the possibility of placing an array of microelectrodes on the inner retinal surface. This approach requires viable retinal ganglion cells, so it is limited to blindness caused by photoreceptor degeneration, such as retinitis pigmentosa and macular degeneration.

Human research to date has employed intraoperative probes and small electrode arrays to determine suitable stimulation parameters and the percepts that they evoke. Data are inconclusive because of the severe limitations on intraoperative experiments and because of uncertainties about the positioning of the electrodes and the condition of the retinal circuitry. The following is a tentative interpretation of the biophysics of retinal stimulation and their implications for the design of a functional visual prosthesis.

In the intact retina, photoreceptors maintain a polarization level that maximizes sensitivity to incident photons, which results in changes in the release of transmitter and in the background spontaneous activity of bipolar and ganglion cells. Even tiny transretinal currents in the μ A range can change the bias levels of these photoreceptors, resulting in perceptions of light and dark phosphenes. In the absence of photoreceptors, electrical stimulation pulses must produce sufficiently intense voltage gradients to depolarize neurons from the resting potential to the threshold for the propagation of action potentials. Retinal neurons are relatively small and unmyelinated, so they would be expected to have high thresholds and long membrane time constants. The output axons of retinal ganglion cells are the largest structures and lie on the vitreous surface of the retina, immediately under the stimulating electrodes, so they would be expected to have the lowest thresholds. However, the axons at a given location originate from a wedge-shaped sector of the retina, so their activation would be expected to produce elongated and overlapping phosphenes. Bipolar cells are more localized and tend to be preserved in most retinopathies, but their electrical thresholds are unclear. Because bipolar cells have longer membrane time constants than ganglion cells (1–2 ms versus 0.5 ms), they can be activated selectively by long-duration pulses (Greenberg et al., 1999). However, such stimulation applied to a large array may result in cross-talk between channels and unacceptable levels of power dissipation.

Information Processing

The introduction of information directly into the CNS, bypassing the natural sensory encoding, raises interesting questions about how that information will be interpreted by the CNS.

Temporal Patterning

Classical neurophysiology is grounded in the notion that the output of each individual neuron represents an independent channel in which the mean spike rate encodes unidimensional information. There have been various theories regarding the encoding and decoding of information in the fine temporal details of activity patterns in ensembles of neurons. One theory of pitch perception held that the acoustic frequency information encoded in the phase-locked activity of auditory afferents could be decoded by cross-correlation of delayed and undelayed versions of this signal. However, patients reported pitch sensations that were dominated by place of electrical stimulation rather than frequency for stimulation rates above about 500 Hz (Eddington et al., 1978). Electrical stimulation of the visual cortex would seem to offer a powerful technique to test current theories regarding the significance of widespread synchronization among neurons responding to a single object in a complex scene.

Neuronal Plasticity

There have been dramatic demonstrations of remapping of both sensory maps and motor representations in primary cortex in response to various surgical, electrical, and behavioral modifications of cortical input (Merzenich and Grajski, 1990). Abrupt or gradual loss of signals from failing sense organs is likely to induce various reorganizations of the ascending pathways, as well as the general atrophy that has been noted. The use of electrical stimulation to restore sensory information inevitably results in somewhat unphysiological temporospatial patterns of neural activity that are likely to induce further reorganizations. For example, recent evidence suggests that the tendency of cochlear implants to produce a better representation of high (basal) versus low (apical) acoustic frequencies may result in a remapping of the central representations and consequent improvements in speech perception (Svirsky et al., 2001). Because of the complex precortical processing of auditory information, the locus of such plasticity will be difficult to identify. Similar experiments in the much simpler visual system have the potential to provide important insights into cortical information processing.

Conclusions

The growing clinical application of neural prosthetics should provide a major catalyst for the expansion of basic knowledge about the nervous system. The devices themselves provide unique opportunities for psychophysical testing of current theories of neural computing and immediate incentives for improving those theories when their limitations are revealed. The technology that is being

developed to build these prostheses has considerable spin-off potential as neurophysiological research tools. Conversely, the nervous system embodies tried and proven solutions to computational problems that have resisted conventional algorithmic approaches of robotics and artificial intelligence. It is difficult to imagine a more appropriate application of electronic neural networks than in the repair of the biological systems that have inspired them.

Road Maps: Applications; Other Sensory Systems

Related Reading: Prosthetics, Motor Control; Prosthetics, Neural

References

- Bak, M., Girvin, J. P., Hambrecht, F. T., Kufta, C. V., Loeb, G. E., and Schmidt, E. M., 1990, Visual sensations produced by intracortical microstimulation of the human occipital cortex, *Med. Biol. Eng. Comput.*, 28:257–259.
- Brindley, G. S., and Lewin, W. S., 1968, The sensations produced by electrical stimulation of the visual cortex, *J. Physiol.*, 196:479–493.
- Busby, P. A., Roberts, S. A., Tong, Y. C., and Clark, G. M., 1991, Results of speech perception and speech production training for three prelingually deaf patients using a multiple-electrode cochlear implant, *Br. J. Audiol.*, 25:291–302.
- Eddington, D. K., Dobelle, W. H., Brackmann, D. E., Mladejovsky, M. G., and Parkin, J., 1978, Place and periodicity pitch by stimulation of multiple scala tympani electrodes in deaf volunteers, *Trans. Am. Soc. Artif. Intern. Organs*, 24:1–5.
- Frijns, J. H. M., Briare, J. J., and Grote, J. J., 2001, The importance of human cochlear anatomy for the results of modiolus-hugging multichannel cochlear implants, *Otol. Neurotol.*, 22:340–349. ♦
- Girvin, J. P., 1988, Current status of artificial vision by electrocortical stimulation, *Neuroscience*, 15:58–62.
- Greenberg, R. J., Velte, T. J., Humayun, M. S., Scarlatis, G. N., and de Juan, E., Jr., 1999, A computational model of electrical stimulation of the retinal ganglion cell, *IEEE Trans. Biomed. Eng.*, 46:505–514.
- Lawson, D. T., Wilson B. S., Zerbi, M., and Finley, C. C., 1996, *Speech Processors for Auditory Prostheses: 22 Electrode Percutaneous Study. Results for the First Five Subjects*, Quarterly Progress Report 3, NIH project N01-DC-5-2103, Neural Prosthesis Program.
- Loeb, G. E., 1990, Cochlear prosthetics, *Annu. Rev. Neurosci.*, 13:357–371. ♦
- Merzenich, M. M., and Grajski, K., 1990, Cortical network changes underlying representational plasticity, *Cold Spring Harbor Symp. Quant. Biol.*, 55:873–887.
- Normann, R. A., Maynard, E. M., Guillory, K. S., and Warren, D. J., 1996, Cortical implants for the blind, *IEEE Spectrum*, 54–59. ♦
- Svirsky, M. A., Silveira, A., Suarez, H., Neuburger, H., Lai, T. T., and Simmons, P. M., 2001, Auditory learning and adaptation after cochlear implantation: A preliminary study of discrimination and labeling of vowel sounds by cochlear implant users, *Acta Otolaryngol.*, 121:262–265.
- Wilson, B. S., Brill, S. M., Cartee, L. A., Cox, J. H., Lawson, D. T., Schatzer, R., Wolford, R. D., Muller, J. M., Schon, F., Tyler, R. S., Kiefer, J., Pfennigdorff, T., and Gstottner, W. (in press), From the 2001 Conference on Implantable Auditory Prostheses: Some likely next steps in the further development of cochlear implants, *Ear Hearing*. ♦
- Wise, K. D., and Najafi, K., 1991, Microfabrication techniques for integrated sensors and microsystems, *Science*, 254:1335–1342.

Pursuit Eye Movements

Richard J. Krauzlis and Leland S. Stone

Introduction

When viewing objects, primates use a combination of saccadic and pursuit eye movements to stabilize the retinal image of the object

of regard within the high-acuity region near the fovea. Although these movements involve widespread regions of the nervous system, they mix seamlessly in normal behavior. Saccades are discrete movements that quickly direct the eyes toward a visual target,

thereby translating the image of the target from an eccentric retinal location to the fovea. In contrast, pursuit is a continuous movement that slowly rotates the eyes to compensate for the motion of the visual target, minimizing blur that can compromise visual acuity. Whereas other mammalian species can generate smooth optokinetic eye movements—which track the motion of the entire visual surround—only primates can smoothly pursue a single small element within a complex visual scene, regardless of any extraneous motion on the retina. This difference likely reflects the greater ability of primates to segment the visual scene, to identify individual visual objects, and to select targets of interest.

Basic Features of Pursuit Behavior

The basic features of pursuit can be illustrated by considering the *ramp paradigm*, in which a target initially at rest moves at a constant speed (Figures 1A and 1B). Pursuit is often interrupted by initial “catch-up” saccades, because the delay in the eye movement response makes the eye lag behind the target. However, if the onset of target motion is accompanied by a position step in the opposite direction, pursuit can be elicited without any catch-up saccades (Rashbass, 1961). The eye movement records obtained with this paradigm can be divided roughly into four phases (Figure 1B). During the latent phase (1), the target is moving, but the eyes have not yet begun to move. During the initiation of pursuit (2), eye speed increases at a nearly constant rate related to the constant image motion experienced during the latent phase. This is followed by a transition phase (3), as eye speed continues to increase and often overshoots target speed slightly. During sustained pursuit (4), eye speed settles to a steady-state value that often oscillates around a value near target speed.

The ramp paradigm illustrates several features of the pursuit system. The first feature is that, as Rashbass (1961) demonstrated, the pursuit response is dominated by target motion; pursuit rotates the eyes in the direction of target motion, even if this is away from the current position of the target. Although subordinate to motion, position offsets can also contribute to the visual drive for pursuit (Pola and Wyatt, 1979).

A second feature is that, because the retina is part of the eye, there is a reciprocal relationship between the motion of the target's retinal image and the motion of the eyes. During the latent phase, the retinal *image speed* (the difference between target and eye speeds) is equal to target speed. Afterward, image speed decreases and then oscillates near zero during sustained pursuit. Pursuit therefore acts like a negative feedback system; its eye-movement output tends to reduce its visual motion input.

A third feature is the relatively long delay (~100 ms) associated with sensory and motor processing. Combined with negative feedback, this delay tends to make the system unstable; in fact, under certain conditions, pursuit can exhibit large-amplitude oscillations. To compensate for this inherent problem, pursuit uses predictive mechanisms. For example, pursuit can maintain a constant speed in the absence of visual motion, perhaps by retaining an *eye velocity memory*. Visual motion is therefore an indicator of how eye speed should change and is correlated with future *eye acceleration* (compare Figures 1C and 1D; see also Lisberger et al., 1981).

A fourth feature is that pursuit provides a steadily changing *muscle force* to produce a constant-speed eye movement (Figure 1E). The required changes in muscle force are a function of eye position in the orbit and can be approximated by taking the mathematical integral of eye speed. This integration process, believed to be common to all eye movements, is accomplished by an *oculomotor integrator* contained within the brainstem.

A fifth feature is that pursuit largely compensates for the mechanical effects on movement dynamics caused by the eye “plant”: the collective term for the inertial mass of the eye and

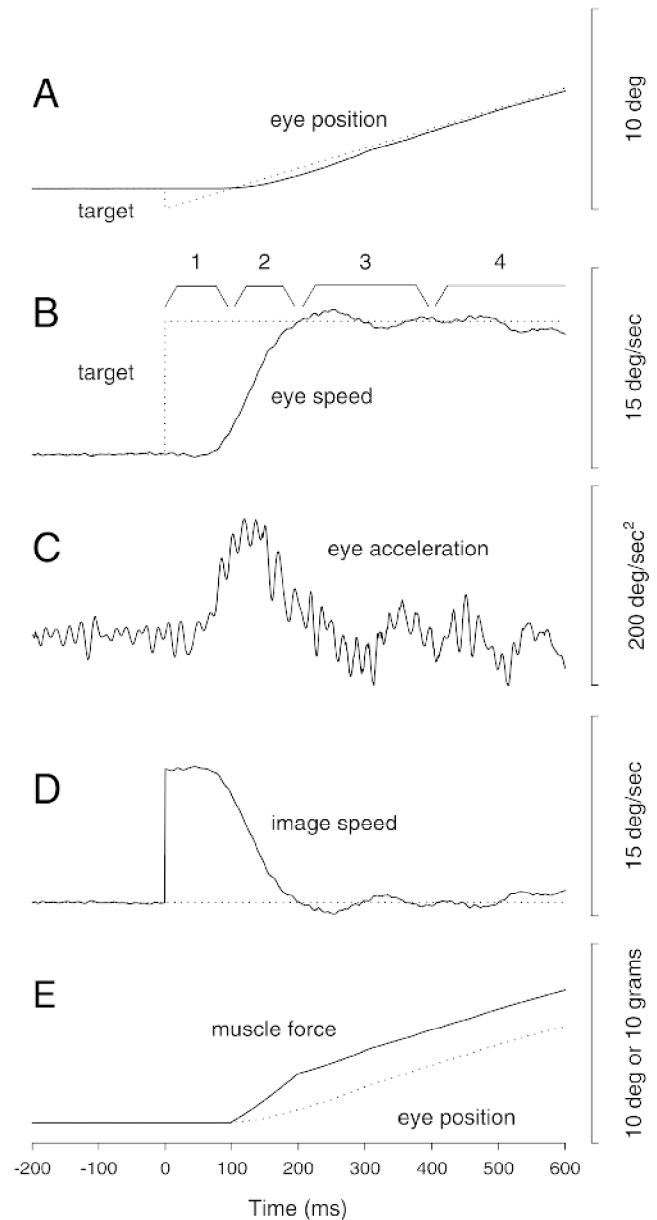


Figure 1. Basic features of pursuit are illustrated with the ramp paradigm. The target jumps to an eccentric position (step) and moves at a constant speed of 10°/s (ramp) that is matched by the human's smooth eye movement after a few hundred milliseconds.

the viscoelastic properties of the eye muscles. As indicated by the initial offset between muscle force and eye position, the applied force begins with an additional boost to overcome the sluggish dynamics of the eye (Robinson, 1965). Without this initial extra force, it would take three to four times longer for pursuit eye speed to match target speed. Pursuit therefore provides an eye movement command that is neurally filtered to compensate for the dynamics of the eye plant.

Pursuit as a Negative Feedback System

Pursuit was originally viewed as a negative feedback velocity-servo system, driven primarily by retinal image motion error signals and

sustained by an internal positive feedback loop to enhance performance (Figure 2A). The contribution of several neural sites to pursuit can be understood within this framework. Areas within the extrastriate cortex that are specialized for processing visual motion, such as the middle temporal area (MT) and the medial superior temporal area (MST), are the major source of the visual-motion information used to guide pursuit (Dursteler and Wurtz, 1988). These cortical regions provide outputs to motor regions in the brainstem and cerebellum that form the motor commands for pursuit. In these subcortical pathways for pursuit, there are reciprocal connections between the ventral paraflocculus of the cerebellum and its target nuclei in the brainstem. This anatomical loop has been suggested to form an eye-command feedback circuit that implements the *velocity memory* for pursuit (Stone and Lisberger, 1990). Purkinje cells in the ventral paraflocculus discharge during pursuit even when there is no image motion on the retina. This discharge, if updated by visual error information from extrastriate cortical areas ($\Delta \dot{E}_d$), could continuously provide a command to change the current eye speed (\dot{E}_c). Although appealing because of its simplicity, there is now abundant evidence that the simple control strategy outlined in Figure 2A cannot account for some of the known physiological and behavioral features of the pursuit system.

Gain Control in the Pursuit System

Contradicting the essentially linear control strategy outlined previously, there is abundant evidence that the pursuit system displays

major nonlinear behaviors. One such nonlinearity is demonstrated by the dramatic changes in efficacy of visual stimuli with behavioral context. For example, it has been shown that rapid displacements of a target can cause a smooth eye acceleration if they are imposed during pursuit of a moving target, but not if they are imposed during fixation of a stationary target. Models of pursuit have simulated these nonlinear effects by including a variable gain element (Krauzlis and Lisberger, 1994). The observation that steady-state eye speed can exceed target speed requires that the proposed variable gain element affect not only the sensitivity to the visual inputs driving the initiation of pursuit, but also the signals maintaining steady-state eye speed.

The behavioral effects of altering activity at different neural sites (Table 1) have helped identify the location of a gain element in the pursuit pathways. Electrical stimulation applied within area MST or the pons can produce changes in smooth eye speed, but only if applied when the monkey is already engaged in pursuit, suggesting that these structures probably lie upstream of a variable gain element. In contrast, stimulation of the ventral paraflocculus elicits smooth eye movements even during fixation, consistent with the proximity of this structure to the final motor pathways for pursuit and its likely placement downstream of the gain element. Finally, microstimulation of the Superior Colliculus (SC) or the pursuit area within the Frontal Eye Fields (FEF) modifies pursuit speed, suggesting that these areas might directly influence a variable gain element (Basso, Krauzlis, and Wurtz, 2000; Tanaka and Lisberger, 2001).

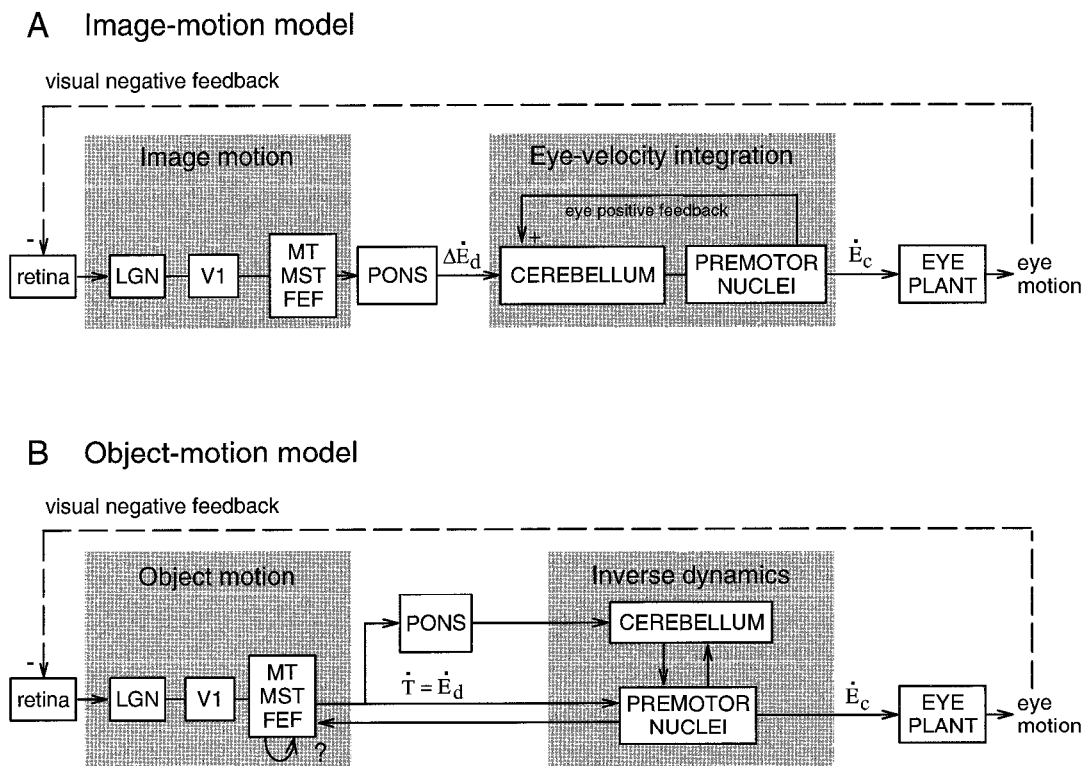


Figure 2. Two models of pursuit that relate system functions to physiology. (A) The “image-motion” assumes that the cortex provides an image-motion signal encoding pursuit error. The subcortical pathways integrate the descending visual error signals or correction commands ($\Delta \dot{E}_d$) to generate a command for smooth eye movement (\dot{E}_c). (B) The “object-motion

model” assumes that the cortex provides a combined visual and extra-retinal signal encoding the target object’s motion. The cerebellar loop transforms this object-motion signal (\dot{T}) or desired eye motion signal (\dot{E}_d) into a plant compensated eye-velocity command (\dot{E}_c), whereas the brainstem provides the required neural integration to generate the eye-position command.

Table 1. Summary of Physiological Studies

Structure	A. Lesions	B. Microstimulation	C. Neuronal recording
1. V1	Retinotopic deficits in saccades and pursuit		
2. Extrafoveal MT	Retinotopic deficits in the initiation of pursuit		Visual responses tuned for direction/speed of small stimuli
3. Foveal MT	Retinotopic deficits in initiating pursuit; directional deficits for ipsiversive sustained pursuit	Ipsiversive eye acceleration if applied during sustained pursuit	Visual responses tuned for direction/speed of small stimuli
4. MST	Deficits in initiating pursuit; directional deficits for ipsiversive sustained pursuit	Ipsiversive eye acceleration if applied during sustained pursuit	Visual responses to small- and large-field motion; extraretinal responses during sustained pursuit
5. 7a, VIP			Visual responses to stimulus motion; extraretinal responses
6. FEF	Deficits in sustained and predictive or anticipatory pursuit	Eye acceleration, often ipsiversive	Visual responses to stimulus motion; responses during pursuit
7. Rostral SC		Contraversive saccades, contraversive eye acceleration if applied during pursuit	Visual responses to stimulus position; responses during fixation, pursuit and saccades
8. DLPN	Deficits in initiating pursuit; deficits for ipsiversive sustained pursuit	Ipsiversive eye acceleration if applied during sustained pursuit	Visual responses best for moving large stimuli; extraretinal responses
9. DMPN, NRTP	Deficits in pursuit		Visual responses to large-field motion
10. NOT	Deficits in ipsiversive pursuit	Ipsiversive eye acceleration	Visual responses to large-field motion
11. LTN			Visual responses to large-field motion
12. Ventral paraflocculus	Deficits in pursuit	Ipsiversive eye acceleration	Responses to eye and head velocity; transient responses during pursuit initiation and changes in eye speed
13. Oculomotor vermis	Deficits in pursuit	Ipsiversive saccades if applied during fixation, ipsiversive eye acceleration or saccades if applied during pursuit	Responses to eye and head velocity; passive visual responses
14. VN, FN, NPH	Deficits in pursuit and saccades		

Abbreviations: V1, primary visual cortex; MT, middle temporal area; MST, medial superior temporal area; VIP, ventral intraparietal area; FEF, frontal eye fields; DLPN, dorsolateral pontine nucleus; DMPN, dorsomedial pontine nucleus; NRTP, nucleus reticularis tegmenti pontis; NOT, nucleus of the optic tract; LTN, lateral terminal nucleus; VN, vestibular nucleus; FN, fastigial nucleus; NPH, nucleus prepositus hypoglossi; SC, superior colliculus.

The Role of the Cerebellum: Velocity Memory or Plant Compensation?

The command signal provided by the central nervous system compensates for the lagging dynamics of the eye plant (Figure 1); this compensation appears to be included in the output signal of the ventral paraflocculus of the cerebellum. One line of evidence has shown that the time-varying profiles of individual Purkinje cell firing rates can be replicated by a weighted average of eye position, eye velocity, and eye acceleration (Shidara et al., 1993). The fits provided by this model suggest that the output of the ventral paraflocculus could represent an inverse dynamic signal. In a more direct test, it has been shown that when the average Purkinje cell output is provided as the input to a model of eye mechanics, the output matches the observed time course of eye velocity (Krauzlis, 2000). These results demonstrate that neural circuits through the ventral paraflocculus are capable of providing the necessary dynamic compensation for the mechanical properties of the eyeball and surrounding orbital tissues.

The evidence in favor of plant compensation suggests an alternative functional role for the brainstem-cerebellar loop. Rather than forming a velocity memory for pursuit, this circuit may be responsible for ensuring that the movement of the eyes matches that of the target as specified by descending signals (i.e., by converting \dot{E}_d

to \dot{E}_c in Figure 2B). This interpretation is consistent with evidence that information about image and eye motion already appears to be combined within cortical areas such as MST, thereby obviating the need to combine them downstream. In fact, the sustained output of MST neurons during perfect steady-state pursuit in the absence of visual motion (Newsome, Wurtz, and Komatsu, 1988) casts doubt on the original interpretation of the similar finding within the ventral paraflocculus (Stone and Lisberger, 1990), which reinforced the view of pursuit shown in Figure 2A. More explicitly, if the cerebellar input from MST is not reduced to zero during perfect steady-state pursuit, then the sustained activity observed at the level of the cerebellum may simply reflect this sustained descending input, rather than the presence of eye-velocity feedback from brainstem structures.

Additional advantages of the plant-compensation view are that it generalizes to other motor systems and simplifies the overall control strategy. All motor systems must contend with the mechanical properties of the body part they control. Rather than attempting to tailor individual sensory signals appropriately for each body part or movement type, the strategy of plant compensation makes it possible for the same input signal to control multiple body parts synergistically—the same object-motion signal could be used in several different ways, allowing one's eyes or one's pointed finger to simultaneously track the same object. This generalization

is consistent with the fact that local circuits throughout the cerebellum are quite uniform, suggesting that they are indeed performing a single consistent process across the body for motor control in general.

The Role of the Cerebral Cortex: Retinal Motion or Object Motion?

The pursuit behaviors that led to the development of the view in Figure 2A were mostly studied with small spots moving over a featureless background. However, one of the distinguishing features of pursuit, as compared to phylogenetically older smooth eye-movement systems, is that, by performing a global analysis of the visual scene, it can track complex objects over textured backgrounds—even when those objects are only partially visible. Experiments using more complex visual stimuli suggest that the descending cortical signal driving pursuit is not a 2D *image motion* signal that relays ongoing eye-movement errors ($\Delta\dot{E}_d$ in Figure 2A), but rather a 3D *object motion* signal that relays the current estimate of the target's trajectory (\dot{T} in Figure 2B), or in motor terms, the desired eye trajectory (\dot{E}_d in Figure 2B). For example, more than two decades ago, Steinbach (1976) provided qualitative evidence that humans can pursue the horizontal motion of a wagon wheel defined only by the cycloidal motion of a few illuminated points along its circumference. More recently, it has been shown that humans can pursue the motion of partially occluded line-figure objects (Stone, Beutter, and Lorenceau, 2000). These stimuli were designed such that the object's motion could only be recovered by selectively grouping its local component line segments (i.e., deciding which pieces belong together) and performing a global motion-integration (i.e., combining the disparate local motions to compute a single object-motion vector). Furthermore, when the static luminance of such line-figure stimuli is altered to induce a percept of independent line segments, rather than of a single moving object, pursuit can no longer accurately follow the object despite the identical image motion. These parallel effects on both motion perception and pursuit cannot be accounted for by any linear system, so future models of both perception and pursuit must contend with this inherently nonlinear processing. They will also have to incorporate the fact that the visual signals driving pursuit are time-varying, initially reflecting image motion and converging towards object motion only after a few hundred milliseconds (Pack and Born, 2001). Finally, the close relationship between smooth eye movement and 3D perception is further supported by the findings that humans can generate smooth vergence eye movements in response to the kinetic depth effect, i.e., perceived motion in depth from global motion integration in the absence of disparity (Ringach, Hawken, and Shapley, 1996). These results imply that the sustained steady-state pursuit previously attributed to an *eye-velocity memory* within a brainstem-cerebellar loop, might actually result from an *object-motion memory* within the cortex, most probably in area MST, derived either from local cortical circuits or ascending feedback from brainstem oculomotor structures (question mark in Figure 2B). Modulation of this object motion signal by nonvisual factors could account for the frequently observed effects of attention and cognitive expectations on pursuit movements (e.g., Kowler, 1989; Barnes, 1993).

Another component of the analysis required for pursuit in real world conditions is the selection of the visual target from within the visible scene. The presence of a moving distractor stimulus can alter the latency or direction of pursuit made to a target stimulus. Although similar effects have been observed with microstimulation of area MT, single-unit studies of MT neurons in selection tasks have produced variable and often only small effects (Ferrera and Lisberger, 1997), suggesting that the process of target selection occurs elsewhere. The role of target selection in pursuit has a close

affinity to the idea of a variable gain element described previously—both putative mechanisms regulate how sensory information accesses the motor pathways for pursuit. Given that the SC and the FEF are involved in saccade target selection, it is tempting to postulate that these areas play a similar role in pursuit (Krauzlis and Dill, 2002).

Pursuit and Saccades: Separate Motor Systems or Coordinated Motor Outputs?

Pursuit and saccades have been viewed as functionally and anatomically distinct eye movement systems, but recent studies have begun to question this assumption. The pursuit-related areas of the cerebral cortex are not restricted to those processing visual motion, but can also be found adjoining each of the saccade-related eye fields; these pursuit and saccade areas have overlapping connections with several subcortical structures. Accordingly, several regions in the brainstem that have been traditionally considered part of the saccadic system now appear to be involved in pursuit as well (Krauzlis and Stone, 1999). Recent single-unit and microstimulation studies show that the SC is involved in the programming of pursuit, in addition to its well-known role in the control of saccades, perhaps by processing target signals that are common to the two types of movements (Krauzlis and Dill, 2002; Basso, Krauzlis, and Wurtz, 2000). The firing rate of some saccade-related “burst” neurons is related to eye speed during pursuit, as well as during saccades (Missal et al., 2000). These new findings suggest the presence of direct pathways through the brainstem for the control of pursuit, in addition to the established pathways through the cerebellum. As we learn more about these alternate pathways, the organization of the pursuit system may more nearly resemble that of the saccade system, consisting of direct pathways from the cerebral cortex to the brainstem and pre-motor nuclei, with a critical but less direct pathway involving the cerebellum.

Discussion

This brief review has outlined in broad strokes old and new frameworks for understanding the sensorimotor processing and control strategy of the pursuit eye-movement system. Control theory models and experiments using small-spot stimuli have helped frame some of the basic organizational principles of the pursuit system. However, the fundamentally linear models that have resulted from this approach cannot account for more recent behavioral and physiological data obtained under more realistic visual stimulus conditions. The growing evidence that inherently nonlinear processes such as image segmentation, selective motion integration, target identification and selection, prediction, and even cognitive and attentional factors play essential roles in pursuit, highlights the critical need for new pursuit models to transcend traditional linear or quasi-linear system control theory. Moreover, the growing evidence of interactions between the pursuit and saccade systems raise basic questions about the overall control strategy employed during tracking eye movements. The operation of pursuit is clearly not limited to the narrow goal of minimizing retinal image motion, but—together with saccadic eye movements—has the broader goal of acquiring and using visual information about real objects in the 3D world to guide motor behavior.

Road Maps: Mammalian Motor Control; Vision

Related Reading: Collicular Visuomotor Transformations for Gaze Control; Eye-Hand Coordination in Reaching Movements; Vestibulo-Ocular Reflex

References

- Barnes, G. R., 1993, Visual-vestibular interaction in the control of head and eye movement: The role of visual feedback and predictive mechanisms, *Prog. Neurobiol.*, 41:435–472.

- Basso, M. A., Krauzlis, R. J., and Wurtz, R. H., 2000, Activation and inactivation of rostral superior colliculus neurons during smooth-pursuit eye movements in monkeys, *J. Neurophysiol.*, 84:892–908.
- Dursteler, M. R., and Wurtz, R. H., 1988, Pursuit and optokinetic deficits following chemical lesions of cortical areas MT and MST, *J. Neurophysiol.*, 60:940–965.
- Ferrera, V. P., and Lisberger, S. G., 1997, Neuronal responses in visual areas MT and MST during smooth pursuit target selection, *J. Neurophysiol.*, 78(3):1433–1446.
- Kowler, E., 1989, Cognitive expectations, not habits control anticipatory smooth oculomotor pursuit, *Vision Res.*, 29:1057–1094.
- Krauzlis, R. J., 2000, Population coding of movement dynamics by cerebellar Purkinje cells, *NeuroReport*, 11:1045–1050.
- Krauzlis, R. J., and Dill, N., 2002, Neural correlates of target choice for pursuit and saccades in the primate superior colliculus, *Neuron*, 35:355–363.
- Krauzlis, R. J., and Lisberger, S. G., 1994, A model of visually-guided smooth pursuit eye movements based on behavioral observations, *J. Comp. Neurosci.*, 1:265–283.
- Krauzlis, R. J., and Stone, L. S., 1999, Tracking with the mind's eye, *Trends Neurosci.*, 22:544–550. ♦
- Lisberger, S. G., Evinger, C., Johanson, W., and Fuchs, A. F., 1981, Relationship between eye acceleration and retinal image velocity during foveal smooth pursuit in man and monkey, *J. Neurophysiol.*, 46:229–249.
- Missal, M., De Brouwer, S., Lefevre, P., and Olivier, E., 2000, Activity of mesencephalic vertical burst neurons during saccades and smooth pursuit, *J. Neurophysiol.*, 83:2080–2092.
- Newsome, W. T., Wurtz, R. H., and Komatsu, H., 1988, Relation of cortical areas MT and MST to pursuit eye movements. II. Differentiation of retinal from extraretinal inputs, *J. Neurophysiol.*, 60:604–620.
- Pack, C. C., and Born, R. T., 2001, Temporal dynamics of a neural solution to the aperture problem in visual area MT of macaque brain, *Nature*, 409:1040–1042.
- Pola, J., and Wyatt, H. J., 1979, Target position and velocity: The stimuli for smooth for pursuit eye movements, *Vision Res.*, 20:523–534.
- Rashbass, C., 1961, The relationship between saccadic and smooth tracking eye movements, *J. Physiol. Lond.*, 159:326–338.
- Ringach, D. L., Hawken, M. J., and Shapley, R., 1996, Binocular eye movements caused by the perception of three-dimensional structure from motion, *Vision Res.*, 36:1479–1492.
- Robinson, D. A., 1965, The mechanics of human smooth pursuit eye movement, *J. Physiol. Lond.*, 180:569–591.
- Shidara, M., Kawano, K., Gomi, H., and Kawato, M., 1993, Inverse-dynamics model eye movement control by Purkinje cells in the cerebellum, *Nature*, 365:50–52.
- Steinbach, M., 1976, Pursuing the perceptual rather than the retinal stimulus, *Vision Res.*, 16:1371–1376.
- Stone, L. S., Beutter, B. R., and Lorenceau, J., 2000, Visual motion integration for perception and pursuit, *Perception*, 29:771–787.
- Stone, L. S., and Lisberger, S. G., 1990, Visual responses of Purkinje cells in the cerebellar flocculus during smooth pursuit eye movements in monkeys. I. Simple spikes, *J. Neurophysiol.*, 63:1241–1261.
- Tanaka, M., and Lisberger, S. G., 2001, Regulation of the gain of visually guided smooth-pursuit eye movements by frontal cortex, *Nature*, 409:191–194.

Q-Learning for Robots

Claude F. Touzet

Introduction

Robot learning is a challenging—and somewhat unique—research domain. If a robot behavior is defined as a mapping between situations that occurred in the real world and actions to be accomplished, then the supervised learning of a robot behavior requires a set of *representative* examples (situation, desired action). In order to be able to gather such a learning base, the human operator must have a deep understanding of the robot-world interaction (i.e., a model). However, in many application domains, such models cannot be obtained, either because detailed knowledge of the robot's world is unavailable (e.g., spatial or underwater exploration, nuclear or toxic waste management), or because it would be too costly. In this context, the *automatic* synthesis of a representative learning base is an important issue. It can be sought using reinforcement learning techniques—in particular, Q-learning, which does not require a model of the robot-world interaction. Compared to supervised learning, Q-learning examples are triplets (situation, action, Q value), where the Q value is the *utility* of executing the action in the situation. A supervised learning base is obtained through the selection by the human operator of the triplets with the highest utility. Robot Learning avoids human operator involvement: an important step toward automatic learning.

Because it allows the synthesis of behaviors despite the absence of a robot-world interaction model, Q-learning (Watkins and Dayan, 1992) has become one of the most used learning algorithms for autonomous robotics. Although the convergence theorem does not apply to the robotics domain (due to the limited number of situation-action pairs that can be explored during the lifetime of robot batteries), heuristically adapted Q-learning has proved successful in applications such as obstacle avoidance, wall following, go-to-the-nest, etc. This is mostly due to *neural-based* implemen-

tations, such as multilayer perceptrons trained with backpropagation, or self-organizing maps. Such implementations provide an efficient generalization, i.e., fast learning, and designate the critic—the reinforcement function definition—as the real issue. The articles REINFORCEMENT LEARNING and REINFORCEMENT LEARNING IN MOTOR CONTROL provide background information on reinforcement learning. Kaelbling, Littman, and Moore (1996), Sutton and Barto (1998) and Wiering, Salustowicz, and Schmidhuber (1999) are three other sources of information. For more detailed treatments, the reader should consult Touzet (1997).

Q-Learning

Figure 1 shows a functional decomposition of Q-learning. Three different functions are involved: *evaluation*, *memorization*, and *updating*. Using the information stored in the robot memory, the current situation is evaluated to select the *best* action to accomplish (i.e., the most reward-promising action). This proposition is modified to allow exploration of the situation-action space. The new situation, entered as a consequence of the execution of the action, is qualified by the reinforcement function. Its qualitative criterion (reinforcement) is used by the updating algorithm to adjust the Q values in the following way:

$$Q(s, a)_{\text{new}} = Q(s, a)_{\text{old}} + \beta(r + \gamma \cdot \text{Max}(Q(s', a)) - Q(s, a)_{\text{old}}) \quad (1)$$

where s is the situation, a is the action, r is the reinforcement, and s' represents all situations that can be reached from s . β and γ are positive coefficients less than 1.

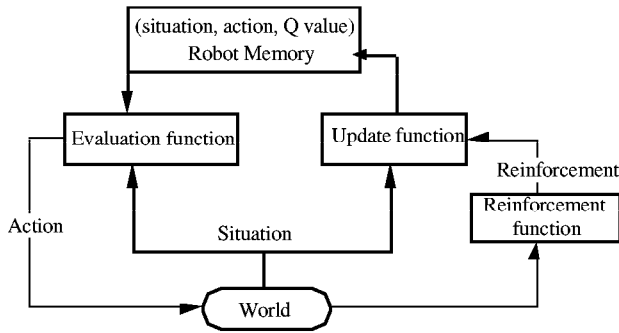


Figure 1. Q-learning method functional decomposition. In response to the present situation, an action is proposed by the robot memory. This action is the one that has the best probability of reward (relatively to the robot memory). However, this proposition may be modified by the evaluation function to allow an extensive exploration of the situation-action space. After the execution of the action by the robot in the real world, a reinforcement function provides a reinforcement value. This value—a simple qualitative criterion (e.g., +1, 1, or 0)—is used by the updating algorithm to adjust the utility value associated with the situation-action pair.

Convergence

Recent developments in the theory of reinforcement learning have allowed proof of asymptotic convergence. These proofs rely on several assumptions that do *not* apply to robots facing real-world tasks. In particular, the asymptotic convergence requires a discrete coding of the situation-action pairs (look-up table storage) and requires attempting every action for every situation an infinite number of times. A robot is a mechanical device that needs at least a few hundred microseconds to execute any action. Therefore, because robot battery life typically lasts less than 10 hours, only a few thousand situation-actions can be visited during a given experiment. This is an extremely small number when compared to the potential number of situation-action pairs (e.g., 10^{26} for the Khepera miniature mobile robot (basic module with eight IR sensors and two wheels), today the most common research robot). Thus, generalization between similar situation-action pairs is *mandatory*.

Generalization

Improvements emphasizing generalization have been proposed by Mahadevan and Connell (1992) who used weighted Hamming distance to generalize between similar situations. This simple method is limited to *syntactic* situation criteria (i.e., it is dependent on the coding of the situations). A second method, proposed by the same authors, adds the action into the syntactic criteria, using clusters to generalize across similar situation-action sets. One of the problems is that the clusters must be handpicked.

Neural Q-Learning

Neural implementations offer a *compact* representation (i.e., limited memory requirement) and good generalization performance (as demonstrated by numerous connectionist applications). The memorization function uses the weight set of the neural network: the memory size required by the system to store the knowledge is defined, a priori, by the number of connections in the network. It is independent of the number of explored situation-action pairs. The proposed action is the processing result of the situation by the network, plus the addition of a random component for the exploration. The update function uses the weight modification algorithm to store the utility values computed by the Q-learning rule (1).

The ideal neural implementation would provide, in a given situation, the best action to undertake and its associated Q value. However, training of such a network requires the definition of an error on the output layer, i.e., knowledge of the best action to undertake in every situation. Such knowledge can be inferred if there are only two different possible actions for the robot, as in the cart pole balancing problem. However, in the general case, the number of actions is larger. Lin (1993)—who proposed the first multilayer perceptron implementation of the Q-learning (Q-Con)—uses as many perceptrons as there are actions, each network output coding for the utility of accomplishing this action in the current situation. Therefore, only one Q-Con network is updated at every time step, and generalization between networks (i.e., actions) is impossible. Other multilayer perceptron implementations have been proposed (Ackley and Littman, 1991; Touzet, 1997), but they do not yet solve the output error definition problem.

Q-Kohon

Unsupervised learning models—such as SELF-ORGANIZING FEATURE MAPS (q.v.), RADIAL BASIS FUNCTION NETWORKS (q.v.), and ADAPTIVE RESONANCE THEORY (ART; q.v.)—do not require an error definition for updating their weight values. Q-Kohon, a Kohonen map implementation of the Q-learning, is a method of state grouping involving syntactic similarity and locality (McCallum, 1995). Each neuron codes a particular triplet (situation, action, Q value); therefore the number of neurons equals the number of stored associations. The neighborhood property of the self-organizing map accounts for the generalization across similar situation-action pairs.

Q-Kohon uses the self-organizing map as an *associative memory*. This associative memory stores triplets. Part of a triplet is used to probe the self-organizing map in search of the corresponding information. Here, situation and Q value are used to find the action: the best action to undertake in a world situation is given by the neuron that has the minimal distance to the input situation and to a Q value of value +1. The selected neuron corresponds to a triplet (situation, action, Q). It is this particular action that should offer the best reward in the world situation. To update the Q value, equation (1) requires the maximum Q value of the new entered situation. This is easily obtained by probing the map with the new situation and a Q value of value +1. The selected neuron Q value will be the maximum possible value.

A nice side effect of using clustering techniques to implement the Q-learning is that the learned behavior can be interpreted by looking (see SELF-ORGANIZING FEATURE MAPS) at the network weights (something extremely difficult with multilayer implementations). Also, because the neurons of the self-organizing map approximate the probability density function of the inputs, one can predict that if a correct behavior is learned, all neurons will code positive Q values. This is most useful to determine when a correct behavior has been learned. This last fact results in the *optimization* of the stored knowledge.

Comparisons

Experiments aimed at comparing various implementations of the Q-learning in a task of synthesizing an *obstacle avoidance behavior* for the miniature robot Khepera (Touzet, 1997) demonstrate that neural Q-learning implementations require a lot less memory and less learning examples, and learn faster (Table 1). The Q-Kohon implementation also exhibits the best behavior after learning, i.e., less negative reinforcements received than all the other implementations.

Table 1. Comparison of Various Implementation.

	Q-learning	+ Hamming	+ Clustering	Q-Comp	Q-Kohon
Time length	55 mn	25 mn	30 mn	8 mn	2 mn
# iterations	7500	3500	4000	2000	500
Memory size	6400	6400	1.6 10 ⁶	56	176

The learning time is the time in seconds needed to synthesize an obstacle avoidance behavior. It reflects the number of real world experiments required. The number of learning iterations is the number of updates to the memory (look-up table or neural network). The memory size is the number of floats required to store the information.

Reinforcement Function Design

The reinforcement function quality is intrinsically limited by the expert's abilities. When a reinforcement learning experiment does not converge, it is impossible to know if this is due to the fact that the experiment was too short and more examples are needed, or if the intrinsic nature of the reinforcement function forbids convergence. Today, reinforcement learning researchers use a slow and painful *trial-and-error* approach to define the reinforcement function. In the meantime, efforts have been devoted to find ways to automatically define such functions. Santos and Touzet (1999) have proposed an Update Parameter Algorithm (UPA) to automatically adjust the threshold values: θ_+ and θ_- within a particular definition of the reinforcement function:

$$RF(s_1, \dots, s_n) = \begin{cases} +1 & \text{if } g_1(s_1, \dots, s_n) > \theta_+ \\ -1 & \text{if } g_2(s_1, \dots, s_n) < \theta_- \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where (s_1, \dots, s_n) is the output readings of the sensors, $g_1()$ and $g_2()$ are any functions linking the sensor data to the rewards.

The resulting effect is to *optimize* the exploration part of the learning phase by achieving and maintaining predefined ratios of positive and negative rewards. If there is no positive reward, the evaluation function built during the learning phase will have "0" as a maximum value and the policy cannot select effective actions. If there is no negative reward, the robot can remain in a dead-end situation forever. If there is no null reward, the evaluation function will be noncontinuous at the frontier between positive and negative situation-action pairs.

A dynamic version of UPA (Santos and Touzet, 1999) updates the threshold values during the learning phase—exploration and exploitation (to take into account the improvement of the robot policy). It allows behavior performance improvements without the need of some sort of external supervisor, capable of ranking situations by difficulty and of choosing tasks of increasing difficulty. (Dorigo and Colombetti, 1998). Santos et al. have been able to synthesize a wall-following behavior by using reinforcement learning (Figure 2), demonstrating support for reinforcement function design techniques.

Discussion

Q-learning is one of the most used (reinforcement) learning technique for behavior-based robots (see REACTIVE ROBOTIC SYSTEMS). Neural-based Q-learning implementations provide compactness and generalizability. Clustering-based neural methods, such as Q-Kohon, allow drastic reduction of the learning time and number of examples required. Their efficiency puts forward the definition of the reinforcement function as a major issue.

Another major issue involves the ability to overcome the exponentially growing number of required learning examples that comes with target behaviors of greater complexity. Battery lifetime seems

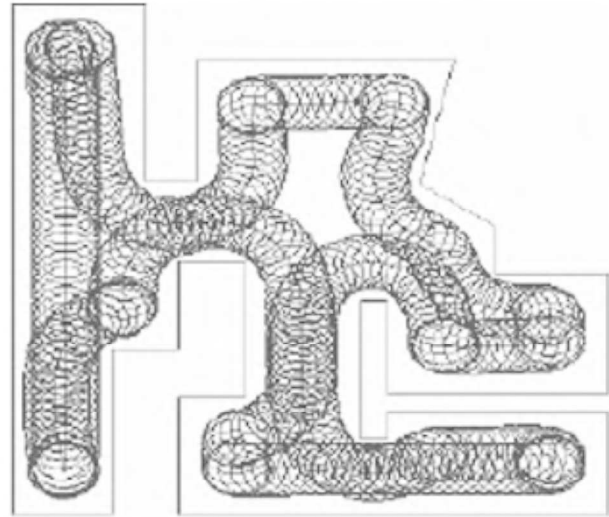


Figure 2. The trace of the miniature Khepera robot after the synthesis of a wall following behavior using a RBF implementation of the Q-learning and Dynamic-UPA in a new environment. Only about 2000 learning iterations are needed.

to impose a definite limit, and researchers tend to promote knowledge incorporation as a speed-up mechanism. The goal is to bias the exploration toward "rewarding" part of the search space—at the expense of *tabula rasa* methods. The drawback is that new—unforeseen—solutions *cannot* be discovered.

Lazy learning (Aha, 1997), also called instance-based learning, provides a way to add samples without implying bias. In a lazy learning approach, computation of the inputs is delayed until the necessity arises. Lazy learning samples the situation-action space, storing the succession of events in memory and, when needed, probes the associative memory for the best move. The sampling process stores the successive situation-action pairs generated by a random action selection policy. The exploration phase is done only once, stored, and used later by all future experiments. The probing of the memory involves complicated computations: clustering, pattern matching, and so forth.

By storing situation-action pairs, a lazy memory builds a non-explicit model of the situation transition function, that is used as a bias to leverage the model-free following learning phase (i.e., Q-learning). Sheppard and Salzberg (1997) propose to mix lazy learning and reinforcement learning, probing the memory with the reinforcement function. Their objective is to provide a method for predicting the rewards for some state-action pairs without explicitly generating them. They call their algorithm *lazy Q-learning*. For the current real-world situation, a situation matcher locates all the states in the memory that are within a given distance. If the situation matcher has failed to find any nearby situations, the action comparator selects an action at random. Otherwise, the action comparator examines the expected rewards associated with each of these situations and selects the action with the highest expected reward. This action is then executed, resulting in a new situation. There is a fixed probability of generating a random action, regardless of the outcome of the situation matcher. New situation-action pairs are added to the memory, along with their Q values computed in the classical way. Among similar situation-action pairs in the memory, an update of the stored Q values is made. There is a limit to the generalizability of this lazy memory because the Q values associated with the situation-action pairs only apply for a particular application.

Learning is not restricted to single robots. Learning in *cooperative robotics* is intriguing: this would be a way to program a set of robots without having to explicitly model their interactions with the world—including the other team members—to achieve cooperation. To achieve this goal, mechanisms that relay the *unique* information associated with the team behavior (reinforcement value) to the individual robots have to be found (see Parker, Touzet, and Fernandez, 2001). Results from the multi-agent research community cannot be applied since they are usually symbolic methods, where robot Q-learning requires a sub-symbolic approach.

Despite all the efforts and success around Q-learning, several drawbacks are associated with supervised and reinforcement learning when it comes to real applications. First, the time needed to achieve the synthesis of any behavior is *prohibitive*, and determining good initial approximations that reduce wasted exploration is not recommended, since it may forbid the finding of unsuspected solutions. Second, the robot behavior during the learning phase is—by definition—bad, and it may even be *dangerous*. To put constraints that preclude dangerous moves implies that there exists a complete model of the robot-world interactions (which by definition does not exist). Third, except within the lazy learning approach, a new behavior implies a *new* learning phase. What is needed is a learning that instantaneously synthesizes any behavior, and which leads to improvement in performance resulting from the mere repetition of this behavior (for a first step in this direction see Touzet, 1999).

Road Map: Robotics and Control Theory

Related Reading: Reinforcement Learning in Motor Control; Robot Arm Control; Robot Learning

References

- Ackley, D., and Littman, M., 1991, Interactions between learning and evolution, in *Artificial Life II*, SFI Studies Sc. Complexity, vol. X (C. G. Langton, C. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, Eds.), Reading, MA: Addison-Wesley, pp. 487–509.
- Aha, D., Ed., 1997, *Lazy Learning*, Dordrecht, Netherlands: Kluwer Academic Publishers (reprinted from *Artif. Intell. Rev.*, 11:1–5). ♦
- Dorigo, M., and Colombetti, M., 1998, *Robot Shaping: An Experiment in Behavior Engineering*, Cambridge, MA: MIT Press. ♦
- Kaelbling L., Littman, M., and Moore, A., 1996, Reinforcement learning: A survey, *J. Artif. Intell. Res.*, 4:237–285. ♦
- Lin, I.-J., 1993, *Reinforcement Learning for Robots Using Neural Networks*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, CMU-CS-93-103.
- Mahadevan, S., and Connell, J., 1992, Automatic programming of behavior-based robots using reinforcement learning, *Artif. Intell.*, 55(2–3):311–365.
- McCallum, R. A., 1995, Instance-based state identification for reinforcement learning, in *Advances in Neural Information Processing Systems 7* (G. Tesauro, D. Touretzky, and T. Leen, Eds.), Cambridge, MA: MIT Press.
- Parker, L. E., Touzet, C., and Fernandez, F., 2001, Techniques for learning in multi-robot teams, in *Robot Teams: From Diversity to Polymorphism* (T. Balch and L. E. Parker, Eds.), Natick, MA: A. K. Peters. ♦
- Santos, J. M., and Touzet, C., 1999, Dynamic update of the reinforcement function during learning, *Connection Science*, special issue on adaptive robots (C. Torras, Ed.), 11(3–4):267–290.
- Sheppard, J. W., and Salzberg, S. L., 1997, A teaching strategy for memory-based control, in *Lazy Learning* (D. Aha, Ed.), Dordrecht, Netherlands: Kluwer Academic Publishers, pp. 343–370.
- Sutton, R., and Barto, A., 1998, *Reinforcement Learning*, Cambridge, MA: MIT Press. ♦
- Touzet, C., 1997, Neural reinforcement learning for behaviour synthesis, *Robotics and Autonomous Systems*, special issue on learning robots: The new wave (N. Sharkey, Ed.), 22(3–4):251–281.
- Touzet, C., 1999, Programming robots with associative memories, in *Proceedings of International Joint Conference on Neural Networks*, Washington, DC, July 10–16.
- Watkins, C. J. C. H., and Dayan, P., 1992, Technical note: Q-learning, *Machine Learning*, 8(3–4):279–292.
- Wiering, M., Salustowicz, R., Schmidhuber, J., 1999, Reinforcement learning soccer teams with incomplete world models, *J. Autonomous Robots*, 7(1):77–88.

Radial Basis Function Networks

David Lowe

Introduction

The radial basis function (RBF) network is conceptually a very simple yet intrinsically powerful network structure. Recent reviews of the RBF approach from a mathematical perspective of function interpolation can be found in Buhmann (2000), from a perspective of regularization and connections to support vectors machines in Evgeniou, Pontil, and Poggio (2000; see also SUPPORT VECTOR MACHINES), and in relation to ideas in statistical analysis in Lowe (1999).

We can motivate the basic idea of the RBF network and reveal its difference from the multilayer perceptron by considering Figure 1. This figure conceptually illustrates a simple classification example in which the distribution of data points exhibits a simple clustering. There are primarily two ways to separate these clusters. One is by segregating the space into polygonal cells. The straight lines in the figure illustrate this decomposition of the pattern space into regions, as would be obtained by a simple multilayer perceptron, where the lines represent the class boundaries. An alternative is to describe the clusters of data themselves as if they were generated according to an underlying probability density function, modeled here in the figure by elliptical distributions. Thus, one

method concentrates on class boundaries and the other focuses on regions where the data density is highest. These are complementary approaches, with respective disadvantages and advantages. The latter alternative is the RBF approach.

The RBF constructs global approximations to functions using combinations of basis functions centered on weight vectors (Figure 2), whereas a multilayer perceptron constructs an architecture out of separating hyperplanes. These weight vectors could be actual data points in the training set, as they are in the basic interpolation model of RBF networks, or they could be chosen according to a utility function, such as a clustering criterion, in density modeling, or a separation criterion, as in support vector machines and regularization. An extra distinction is that the RBF employs a distance function to convert the vector input pattern into a scalar at the hidden layer, as opposed to a vector dot product such as is used in the multilayer perceptron. The network's strength derives from a rich interpretational foundation because it lies at the confluence of a variety of established scientific disciplines. Thus, although the original motivation of this particular network structure was in terms of function approximation techniques (Broomhead and Lowe, 1988; Powell, 1992), the network may be derived on the basis of statistical pattern processing theory (Lowe, 1999), regression and

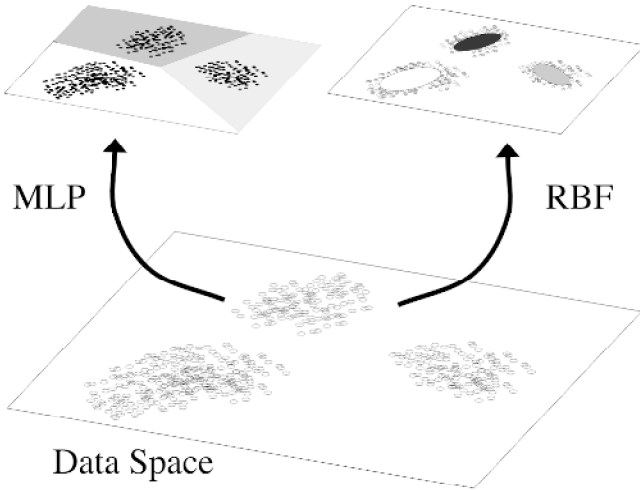


Figure 1. Dissection of pattern space by clusters and hyperplanes. The multilayer perceptron (MLP) exploits the logistic nonlinearity to create combinations of hyperplanes to dissect pattern space into separable regions. Subsequent layers combine these regions to allow the formation of non-convex class boundaries. The radial basis function (RBF) dissects pattern space by modeling *clusters* of data directly and so is more concerned with data distributions.

regularization (Evgeniou et al., 2000), biological pattern formation (Logotheis, Pauls, and Poggio, 1995), mapping in the presence of noisy data and, more recently, in terms of kriging approximations (Wan and Bone, 1997) and kernel methods, particularly support vector machines (see SUPPORT VECTOR MACHINES). However, in addition to exhibiting a range of useful theoretical properties, the RBF network structure is above all a practical construct, as it may be applied efficiently to problem domains in discrimination (such as speaker verification), time series prediction (such as economic modeling), and feature extraction or even topographic mapping problem domains (such as encoding the sensory space of an artificial nose in chemical vapor analysis). Some studies have even suggested that RBF topology developed in nature as an evolutionary functional counterpart to object views for vision, where the

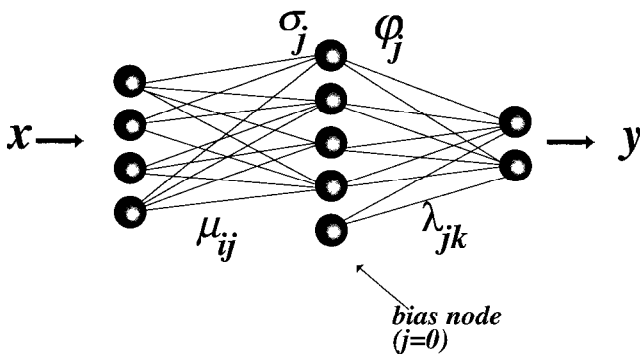


Figure 2. The basic radial basis function structure. A nonlinear basis function $\phi_j(\dots)$ is centered on each hidden node weight vector μ , which also has a (possibly) adaptable “range of influence” σ_j . The output of the hidden node j , h_j is given as a *radial* function of the distance between each pattern vector and each hidden node weight vector, $h_j = \phi_j(\|x - \mu_j\|/\sigma_j)$. This is the main difference from a multilayer perceptron. The network outputs are evaluated by a traditional scalar product between the vector of hidden node outputs and the weight vector attached to output node k , as $o_k = h \cdot \lambda_k$.

basis functions correspond to cell aggregates tuned to particular views of familiar objects (Logotheis et al., 1995). This idea supports Edelman’s “chorus of prototypes” and is reflected in the approach adopted in Moody and Darken (1989) using locally tuned processing units, although understanding of the biology in this area is presently insufficient to support generalizations.

The Basic RBF Structure

The RBF is a single-hidden-layer feedforward network with linear output transfer functions and nonlinear transfer functions, $\phi_j(\dots)$, on the hidden layer nodes. Many types of nonlinearities may be used. There is also typically a bias on each output node. The primary adjustable parameters (see Figure 2) are the final layer weights, $\{\lambda_{jk}\}$, connecting the j th hidden node to the k th output node. There are also weights $\{\mu_{ij}\}$ connecting the i th input node with the j th hidden node, and occasionally a “smoothing” factor matrix, $\{\Sigma_j\}$. It often helps to think of the weights $\{\mu_{ij}\}$ as representing “prototypes” of patterns in the input space, either as cluster centers or as pattern exemplars that are in some sense representative of the distribution of input patterns. Similarly, the weights $\{\Sigma_j\}$ that govern the regularization of the network (after the model order complexity of the number of basis functions) are sometimes considered to represent the range of influence of these prototypes (though this intuition can break down for some of the nonlocal basis functions that can be employed).

The mathematical embodiment of the RBF takes the following form. The k th component of the output vector y_p corresponding to the p th input pattern x_p is expressed as

$$[y(x_p)]_k = \sum_{j=0}^h \lambda_{jk} \phi_j(\|x_p - \mu_j\|; \Sigma_j) \quad (1)$$

where $\phi_j(\dots)$ denotes the nonlinear transfer function of hidden node j , ($\phi_0(\dots) \equiv 1$ is the bias node), and the possible dependence on a smoothing matrix is left explicit. The most common example of the smoothing factor is in the use of a general Gaussian transfer function, i.e., $\phi(z) \approx \exp[-z^T \Sigma^{-1} z]$. Since the general expression is an analytic function of the variables corresponding to the basis function positions and smoothing factors, it is possible to adapt them by a full nonlinear least squares process if required (Moody and Darken, 1989). This is usually not necessary. As can be seen from Equation 1, the main difference from a multilayer perceptron is that the output of the hidden node j , h_j is given as a *radial* function of the distance between each pattern vector and each hidden node weight vector (or prototype), $h_j^{RBF} = \phi_j(\|x - \mu_j\|)$, rather than as a scalar product, $h_j^{MLP} = \phi_j(x \cdot \mu_j)$.

One of the advantages of the RBF is that the first-layer weights $\{\mu_{ij}, \Sigma_j; j = 1, \dots, h\}$ may often be determined or specified by a judicious use of prior knowledge, or adapted by simple techniques. Early work (Broomhead and Lowe, 1988) found it sufficient to position the basis functions at data points sampled randomly according to the distribution of the data. This ensured that network resources were concentrated in regions of higher data density. Another early technique (Moody and Darken, 1989) was to position the centers of the basis functions according to a K -means clustering process on the data points, and then set the smoothing parameters of the assumed Gaussian basis functions to be the average distance between cluster centers. Over the past decade, many variations on these themes have been introduced that employ some form of prior knowledge expressed as a utility function, which the choice of prototypes should try to optimize. Therefore, once the weights associated with the first layer have been specified, the major problem in training an RBF network is focused on determining the final layer weights. Since the RBF network is typically employed to perform a *supervised* discrimination or prediction task, such as time

series forecasting, this training usually takes the form of optimizing a cost function requiring the outputs of the network to somehow closely approximate a set of known target values. It is common to attempt to minimize a standard residual sum-of-squares cost function, although other cost functions may be employed. Because this is a linear optimization process (the parameters $\{\lambda_{jk}\}$ occur linearly when minimizing the residual sum-squared-error measure), the RBF is computationally more attractive in applications than a multilayer perceptron, even though they are both computationally universal architectures (Park and Sandberg, 1991).

RBFs for Classification

One of the more common uses for an RBF network is as a semi-parametric classification model capable of modeling nonlinear boundaries between clusters. This problem can be analyzed statistically and may be motivated from the perspective of kernel-based density estimation (Tr  v  n, 1991). Here we outline how the RBF architecture emerges from this statistical perspective, although historically it first emerged from interpolation theory.

In classification, we are primarily interested in the posterior, $p(c|\mathbf{x})$, the probability that class c is present given the observation \mathbf{x} . However, it is easier to model other related aspects of the data, such as the unconditional distribution of the data, $p(\mathbf{x})$, and the likelihood of the data, $p(\mathbf{x}|c)$, which is the probability that the data were generated given that the data came from a specific class c . We can then recreate the posterior from these quantities according to Bayes's theorem, $p(c|\mathbf{x}) = p(c)p(\mathbf{x}|c)/p(\mathbf{x})$. The distribution of the data is modeled as if it were generated by a mixture distribution, i.e., a linear combination of parameterized states or of basis functions such as Gaussians. Since individual data clusters for each class are not likely to be approximated by a single Gaussian distribution, we need several basis functions per cluster. We assume that the likelihood and the unconditional distribution can both be modeled by the same set of distributions, $q(\mathbf{x}|s)$, but with different mixing coefficients, i.e., $p(\mathbf{x}) = \sum_s \hat{p}(s)q(\mathbf{x}|s)$ and $p(\mathbf{x}|c) = \sum_s p(s; i)q(\mathbf{x}|s)$. Then the quantity we are interested in $p(c|\mathbf{x}) = p(c)p(\mathbf{x}|c)/p(\mathbf{x})$ is given by

$$p(c|\mathbf{x}) = \frac{\sum_s p(c)p(s; i) \hat{p}(s)q(\mathbf{x}|s)}{\sum_s \hat{p}(s)q(\mathbf{x}|s)} = \sum_j \lambda_{ij}\phi(\mathbf{x}|\mathbf{j})$$

where $\lambda_{ij} = p(c)p(j; i)/\hat{p}(j)$ relates the overall significance of state j to class i , and $\phi(\mathbf{x}|\mathbf{j})$ is a normalized basis function, $\hat{p}(j)q(\mathbf{x}|\mathbf{j})/\sum_s \hat{p}(s)q(\mathbf{x}|s)$.

This gives an RBF architecture. For a total of h functions used to approximate the likelihood and the unconditional density, there are h hidden nodes corresponding to the normalized basis functions, and the final layer weights relate the significance of the hidden nodes to the c output class nodes, providing the class-conditional information. Of course, the positions and possibly also the ranges of influence of each of these basis functions need to be specified or adapted to allow an adequate model of each data cluster. This can be achieved by unsupervised clustering techniques.

In this manner, the RBF is an ideal network for use in classification problems. Note that the architecture of RBF networks for density estimation is more general than was indicated in the discussion of motivation. In particular, it is not essential that each basis function itself should be a probability density function.

RBFs for Prediction

In the previous section, the RBF was motivated by a statistical interpretation of data distributions. In that case, the underlying generator of the data (the probability density function) was sampled

stochastically. However, the original formulation of the RBF network was developed in order to produce a deterministic mapping of data by exploiting links with traditional function approximation. This approach attempted to introduce the notion that the training of neural networks could be described as curve fitting. Hence, "generalization" consequently has a natural interpretation as interpolating along this fitting surface.

The basic idea was as follows. Assume that we have a set of input/output pairs of input/target patterns representing data from an unknown but smooth surface in $\mathbb{R}^n \times \mathbb{R}^c$. As a simple example, consider a set of (x, y) pairs generated according to $y = x^2$. In this approach, the problem is to choose a function $\mathbf{y} : \mathbb{R}^n \rightarrow \mathbb{R}^c$, which satisfies the interpolation conditions $\mathbf{y}(\mathbf{x}_p) = \mathbf{t}_p$, $p = 1, 2, \dots, P$. This is *strict* interpolation, in which the function is constrained to pass through all the known data points. The strategy in interpolation theory was to construct a linear function space spanned by a set of nonorthogonal basis functions that depended on the positions of the known data points. The radial basis function expansion mapping to one dimension was originally expressed as $\mathbf{y}(\mathbf{x}) = \sum_{j=1}^P \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|)$. By using the interpolation conditions, the fitting parameters λ may be determined by matrix inverse methods. The approach was generalized to higher-dimensional mappings, to incorporate bias terms, and to account for the fact that strict interpolation is not a good strategy for real-world noisy data, leading to the feedforward neural network topology already discussed.

In the case of the simple $y = x^2$ example mentioned above, the inputs are x values, the targets are specific y values, and the RBF network is constructed so as to produce a fitting surface to the parabola $y = x^2$, which is a surface in $\mathbb{R}^1 \times \mathbb{R}^1 = \mathbb{R}^2$. Note that this is curve fitting to the *parabola* in the product space, not the data samples themselves. This parabola may be interpreted as the *generator* of the observed data, as locations on the parabola "produce" or "generate" the (x, y) input/output pairs. As long as the generator of the data is smooth and nonlinear, we should be able to arbitrarily closely approximate it with an RBF network. This explains why networks can be successful in predicting deterministically chaotic time series: although the time series themselves may exhibit apparent randomness, the underlying map that produces the samples is itself usually very smooth.

Miscellaneous Topics

Many topics related to RBF networks cannot be addressed in detail here. Among them are the issues of how many centers to use, what types of nonlinearities may be employed, Bayesian approaches to choosing the smoothing parameters, how to optimize the various weights, how to assist generalization through regularization, and how to determine confidence intervals. A few of these topics are briefly discussed below.

Choice of Kernel Function

The theory of statistical density estimation produces many recommended bounded kernels that may or may not be density functions themselves. Examples of $\phi(z)$ are the Epanechnikov $(3/4)[1 - z^2/5]/\sqrt{5}$ for $|z| < \sqrt{5}$, and 0 otherwise), the triangular $(1 - |z|)$ for $|z| < 1$ and 0 otherwise), and of course the Gaussian. From interpolation theory the following choices are common: cubic (z^3), thin-plate spline ($z^2 \log z$), inverse multiquadric $([z^2 + c^2]^{-1/2})$, multiquadric $([z^2 + c^2]^{1/2})$, and again the Gaussian. Note that these functions do not have finite support, and indeed, some of the choices are *unbounded* functions, contrary to intuition and common belief that network basis functions are localized. However, despite this unbounded nature, it is correct that the parameters of the network may be chosen such that $\mathbf{y}(\mathbf{x}) \rightarrow 0$ as $\mathbf{x} \rightarrow \infty$ so that the network

as a whole achieves a localized response, even if the individual basis functions do not.

Regularization

Various schemes have emerged to discourage overfitting of the training data points. Such options include (1) choosing a small set of initial centers and adapting the positions and spreads to best describe the data in some sense; (2) regularization—having a center located at each data point, but adding a smoothing term that is an effective constraint on the possible weight values, the magnitude of this extra term governing the amount of smoothing applied to the fitting surface (see GENERALIZATION AND REGULARIZATION IN NONLINEAR LEARNING SYSTEMS); (3) selecting centers over a subset of the data points in the training set incrementally to maximize the descriptive power of the data variance obtained by adding each new basis function; and (4) Bayesian approaches of using the evidence to estimate the hyperparameters and weights. Because of its importance, we outline the regularization approach. Extensive and useful reviews of this approach, along with further references, can be found in Haykin (1999) and Evgeniou et al. (2000).

In regularization we wish to interpolate a finite data set using an approximation $y(\mathbf{x})$. Of all possible approximators, we wish to choose the one that minimizes the augmented functional

$$H[y] = \sum_{p=1}^P (t_p - y(\mathbf{x}_p))^2 + \eta \|\hat{O}y\|^2$$

where \hat{O} is an operator such as $\partial/\partial\mathbf{x}$, which is a mathematical embodiment of our prior knowledge on desired smoothness constraints. For example, this particular choice of operator gives an extra component to the overall cost function that represents curvature of the resulting map (it is an approximation to the expected Hessian or matrix of second derivatives of the map). So, high curvature solutions will incur a higher penalty than lower curvature solutions. Overall, then, there will be an interplay between the desire to produce a very-low-curvature RBF mapping and the desire to produce a surface that exactly passes through each data point. This latter objective usually requires very high curvature surfaces, since there will be random noise on the data points. η is the regularization parameter that embodies the degree to which the constraint should dominate the data. Interestingly (Lowe, 1999), the functional that formally minimizes $H[y]$ takes the form of an RBF, i.e., $y(\mathbf{x}) = \sum_{p=1}^P \lambda_p G^\dagger(\mathbf{x}, \mathbf{x}_p)$. Here, $G^\dagger(\mathbf{x}, \mathbf{x}_p)$ denotes a Green function that is a solution of an equation determined by the regularization operator. If the operator $\hat{O}^\dagger \hat{O}$ is rotationally and translationally invariant, then the Green function is only a function of the radial inferences of its arguments, i.e., $G(\|\mathbf{x} - \mathbf{x}_p\|)$. Thus, once again the form of the RBF may be derived, but this time from the perspective of preventing overfitting by regularization. In this case the form of the basis function is determined by the type of smoothness constraint we have imposed. As in the previous interpolation case, the weighting coefficients, λ_p , may be determined by the solution of a linear equation. Again, strictly speaking, this approach requires a center or basis function located at each data point, and overfitting is avoided by imposing the smoothing constraint. However, in practice the number of centers may also be chosen to vary, in which case these precise mathematical relationships no longer hold.

Discussion

This article has discussed the motivation and application of the radial basis function network from a variety of perspectives. We have chosen to concentrate on contrasting a statistical pattern processing perspective with a function approximation perspective. However, both perspectives have the common philosophical basis that the aim of the network is to approximate the underlying structure that generated the observed data, rather than the data itself. This is also how a multilayer perceptron operates. However, the RBF was introduced to make this link with curve fitting and interpolation explicit.

The RBF may be employed in classification tasks, time series prediction, and both unordered and topographic feature extraction. Because of its computational tractability, the RBF has been applied to many diverse real-world problems, and there is some evidence that its structure may have similarities to biological aspects of vision processing. But above all, its strength and utility derive from its simplicity and from a close relationship with other areas of signal and pattern processing and other neural network architectures. These connections and interpretations are still being uncovered, as recent work on support vector machines (see SUPPORT VECTOR MACHINES), Gaussian processes, and kriging has demonstrated.

Road Maps: Grounding Models of Networks; Learning in Artificial Networks

Related Reading: Bayesian Methods and Neural Networks; Support Vector Machines

References

- Broomhead, D. S., and Lowe, D., 1988, Multivariable functional interpolation and adaptive networks, *Complex Syst.*, 2:321–355.
- Buhmann, M. D., 2000, Radial basis functions, *Acta Numerica* (A. Dseres, Ed.), 9:1–38.
- Evgeniou, T., Pontil, M., and Poggio, T., 2000, Regularization networks and support vector machines, *Adv. Computat. Math.*, 13:1–50.
- Haykin, S., 1999, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Englewood Cliffs, NJ: Prentice Hall, chap. 5. ♦
- Logothetis, N. K., Pauls, J., and Poggio, T., 1995, Shape recognition in the inferior temporal cortex of monkeys, *Curr. Biol.*, 5:552–563.
- Lowe, D., 1999, Radial basis function networks and statistics, in *Statistics and Neural Networks: Advances at the Interface* (J. W. Kay and D. M. Titterton, Eds.), New York: Oxford University Press, pp. 65–95.
- Moody, J., and Darken, C., 1989, Fast learning in networks of locally tuned processing units, *Neural Computat.*, 1:281–294.
- Park, J., and Sandberg, I. W., 1991, Universal approximation using radial basis function networks, *Neural Computat.*, 3:246–257.
- Powell, M. J. D., 1992, The theory of radial basis function approximation in 1990, in *Advances in Numerical Analysis*, vol. 2, *Wavelets, Subdivision Algorithms and Radial Basis Functions* (W. A. Light, Ed.), Oxford, Engl.: Oxford University Press, pp. 105–210.
- Tråvén, H. G. C., 1991, A neural network approach to statistical pattern classification by “semiparametric” estimation of probability density functions, *IEEE Trans. Neural Netw.*, 2:366–377.
- Wan, E., and Bone, D., 1997, Interpolating earth science data using RBF networks and mixtures of experts, *Adv. Neural Inf. Process. Syst.*, 8:988–994.

Rate Coding and Signal Processing

Fabrizio Gabbiani

Introduction

In the peripheral and central nervous system, many neurons encode information and pass it on to other neurons by generating irregular sequences of short voltage pulses, typically less than 1 ms in duration, called action potentials. The shape of these action potentials, or spikes, is usually quite stereotyped over the course of time. The sequence of spike occurrence times generated by the cell, often called the *spike train*, is therefore thought to carry most of the information that a neuron communicates to its targets. When studying how sensory information might be encoded in neuronal spike trains, one is faced with the fact that spike trains are often quite variable under seemingly identical stimulation conditions (Figure 1). Is this variability simply noise, perhaps due to uncontrolled changes in the state of some internal variable, or does it carry information about the stimulus? Answering this question in a particular case would require a thorough knowledge of the biophysical mechanisms of spike generation and of stimulus coding—knowledge that is out of reach at present.

Although no universal definition exists, the term *rate coding* is applied in situations where the precise timing of spikes is not thought to play a significant role in carrying sensory information. Rate codes have been identified in many sensory systems and are probably the best understood means by which neurons encode information. In many cases, rate codes have been shown to play an important role in determining behavioral responses of animals.

Rate coding comes in two flavors: mean firing rate codes and instantaneous firing rate codes. The sensory information conveyed by these two types of codes can be studied rigorously by applying classical methods of statistical signal processing borrowed from the engineering literature. In the next two sections, we will show how these methods can be carried over to the analysis of neuronal spike trains. Before turning to more general examples in the third section, we will illustrate them in the case of electrical field amplitude-sensitive neurons of weakly electric fishes. These animals possess an unusual sense for the electrical properties of their environment that is favorable to computational investigations (see ELECTROLOCATION).

Rate coding is not the only mean by which neurons convey information. In weakly electric fishes and in other auditory-like sensory systems, the role played by spike timing information is well documented (see ELECTROLOCATION and ECHOLOCATION: COCHLEOTOPIC AND COMPUTATIONAL MAPS). Two articles address the

issue of spike timing in cortical circuits (SYNFIRE CHAINS; SYNCHRONIZATION, BINDING AND EXPECTANCY). Finally, the role of rate coding in the context of neuronal populations is examined in POPULATION CODES (q.v.) and MOTOR CORTEX (q.v.), CODING AND DECODING OF DIRECTIONAL OPERATIONS (q.v.).

Mean Firing Rate Coding

An increase in firing rate is typically the most conspicuous change recorded from sensory neurons in response to external stimuli. It is therefore natural to ask how well the spike count observed in a single trial from such a cell can predict the presence of the stimulus. Let us take the example of a neuron having a mean spontaneous rate $\bar{\lambda}_0 = 30$ spk/s that fires at a rate of $\bar{\lambda}_s = 50$ spk/s under stimulus presentation (Figure 1B). We will first assume for simplicity that spikes are generated completely independently of each other (i.e., following a Poisson process) and thus do not carry any additional information beyond their mean rate of occurrence.

Figure 2A illustrates the distribution of spike counts observed during a 200 ms window in the baseline and stimulus condition for this model neuron. The overlap between these two distributions indicates that guessing the presence of the stimulus from the spike count observed in a single trial will lead to a significant fraction of errors. A simple method to decide between the two alternatives “stimulus present” or “no stimulus” consists in choosing a threshold number of spikes, k_{thres} , and classifying the observed responses, n , as baseline activity or stimulus-induced activity according to whether the threshold is exceeded or not:

$$\begin{aligned} n < k_{\text{thres}} &\Rightarrow \text{baseline activity} \\ n > k_{\text{thres}} &\Rightarrow \text{stimulus present} \end{aligned} \quad (1)$$

This decision strategy leads to two types of errors. On the one hand, we might call for the stimulus to be present in a trial during which spontaneous activity was unusually high. This type of error is called a false alarm. On the other hand, we might confuse an unusually low stimulus response with spontaneous activity, an error called a false miss. Clearly, the proportion of false alarms to false misses depends on the choice of the threshold k_{thres} : high (low) threshold values correspond to low (high) probabilities of false alarms with higher (lower) fractions of false misses. It is customary to characterize the performance of this detection algorithm by varying the threshold from low to high values and plotting the probability of correct detection, p_D (i.e., 1 minus the probability of false

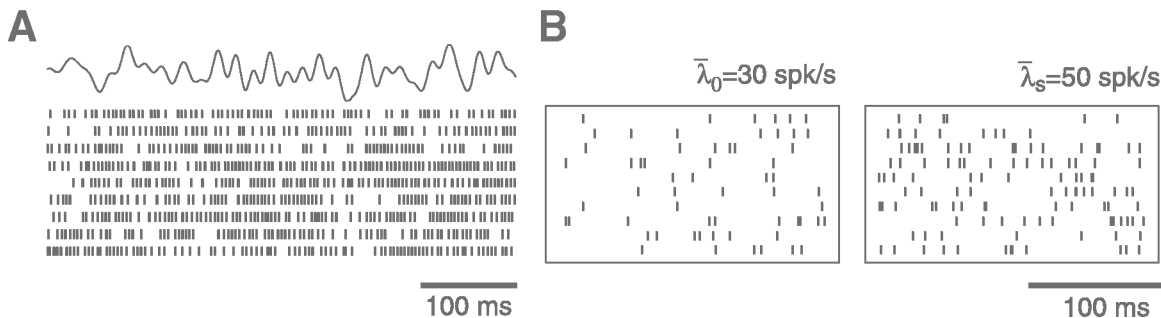


Figure 1. A, Nine spike trains recorded from an amplitude-sensitive afferent in the weakly electric fish *Eigenmannia* in response to repeated presentations of the same random electrical field amplitude modulation (shown on top). (Adapted from Kreiman et al., 2000.) B, Ten spike trains (200 ms

long) obtained from a Poisson process with mean firing rate $\bar{\lambda}_0 = 30$ spk/s (spontaneous rate, left) and $\bar{\lambda}_s = 50$ spk/s (stimulus-induced rate, right).

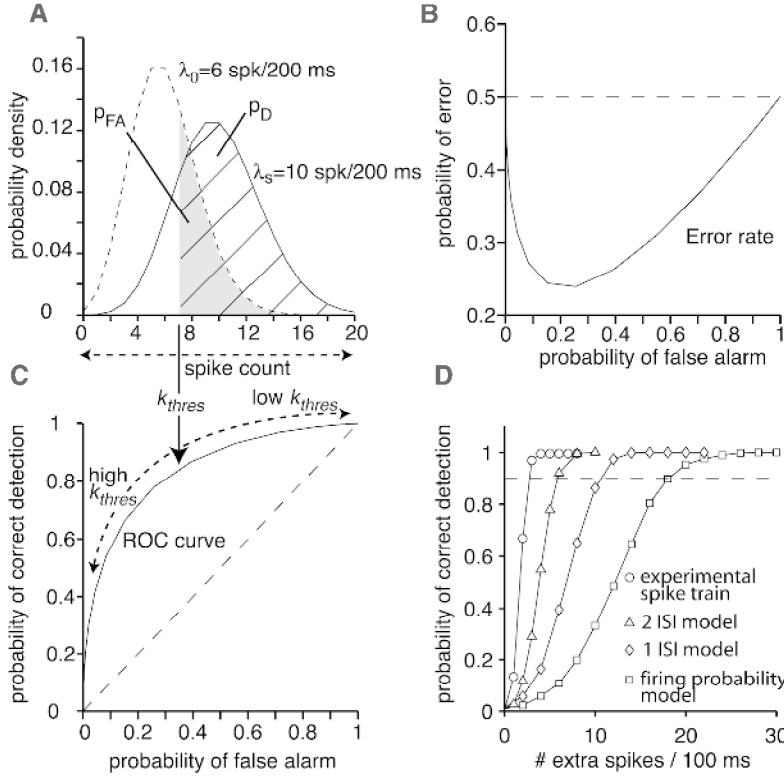


Figure 2. A, Probability distributions of the spike count observed in a 200 ms window for the two Poisson processes illustrated in Figure 1B. Choosing a threshold number of spikes (k_{thres}) to discriminate between the presence or absence of the stimulus leads to errors because of the overlap of the distributions. The probability of false alarm (p_{FA}) and of correct detection (p_D) are illustrated by the gray and hatched areas, respectively. B, Plot of p_D versus p_{FA} , called an ROC curve. Different thresholds will correspond to different values of p_D and p_{FA} (dashed double arrows in A and B). C, Overall probability of error (see Equation 2), computed from the ROC curve in B. D, Probability of correct detection obtained from the spike trains of an amplitude-sensitive afferent neuron as a function of the number of spikes above spontaneous activity generated by the cell (circles). Note that the cell can discriminate with more than 90% accuracy increases of three spikes or more, corresponding to a 1% increase in firing rate. Different models that take into account only the mean firing rate (squares), the mean firing rate and the interspike interval distribution (diamonds), or in addition the joint properties of successive intervals (triangles) are unable to match the experimental performance. (Adapted from Ratnam and Nelson, 2000.)

misses), as a function of the probability of false alarms, p_{FA} (i.e., 1 minus the probability of correct rejections; Figure 2B).

This curve is called the receiver operating characteristic (ROC) curve of the detection algorithm (a term originating from early applications to radar observations). The dashed diagonal line $p_D = p_{FA}$ corresponds to chance performance (i.e., independent of the threshold, k_{thres} , the probability, p_D , of correctly detecting the stimulus is as good as the probability, p_{FA} , of incorrectly mistaking spontaneous activity with stimulus-induced activity). Thus, the higher the ROC curve lies above the diagonal, the better the performance of our detection algorithm and, in the limit of perfect performance, $p_D = 1$ over the entire interval $0 \leq p_{FA} \leq 1$.

From the ROC curve, one can obtain the values of p_{FA} and $p_D(p_{FA})$ that minimize the overall probability of stimulus detection error (comprising both false alarms and false misses). If the stimulus is presented on average in one-half of the trials, the error rate is given, for a fixed value of p_{FA} , by

$$\varepsilon(p_{FA}) = \frac{1}{2} p_{FA} + \frac{1}{2} (1 - p_D(p_{FA})) \quad (2)$$

The minimum of $\varepsilon(p_{FA})$ as a function of p_{FA} can be easily found by numerical methods (see Figure 2C). The corresponding threshold k_{thres} may then be obtained from $p_{FA}(k_{\text{thres}})$. Thus, in our example the minimal error rate $\varepsilon = 0.24$ is achieved for $p_{FA} = 0.26$, corresponding to a detection threshold k_{thres} of 8.5 spk/s.

One important question remains: Given the simplicity of this algorithm, could it be outperformed by a more sophisticated one? Remarkably, this is not the case: under fairly general assumptions, the threshold condition of Equation 1 is equivalent to a similar condition on the likelihood ratio, $l(n) = p_s(n)/p_0(n)$, where $p_s(\cdot)$ and $p_0(\cdot)$ are the probability distributions of the spike count in the presence and absence of the stimulus, respectively (Figure 2A). The likelihood ratio is a quantity central to signal detection theory, and this equivalence shows that, for a fixed value of p_{FA} , no other al-

gorithm taking into account only the probability distributions of Figure 2A can outperform the threshold test. Thus, the ROC curve defines the performance of an ideal observer of the mean rate code, having complete access to $p_0(\cdot)$ and $p_s(\cdot)$. Whether neurons or neuronal networks in the brain adopt similar algorithms to read out information about the external world remains an open question.

The performance of the ideal observer algorithm will be affected by at least two additional factors, the first being the length of the time window over which spikes are registered. Longer windows typically lead to better performance by averaging out the noise component of the spike rate that causes deviations from the mean. In the case of the Poisson process discussed above, for a fixed mean firing rate $\bar{\lambda}$, the mean number of spikes observed in a time window T is given by $\bar{n} = \bar{\lambda}T$, whereas the standard deviation is $\sigma = \sqrt{\bar{\lambda}T}$. Thus, $\bar{n}/\sigma \propto \sqrt{T}$, and the signal grows as \sqrt{T} with respect to noise over the course of time. Currently, the time interval that is relevant for behavioral responses often is only weakly constrained by experimental observations.

The second factor is the regularity of the spike train or, in other words, the amount of noise that is present to start with. While many neurons in cortical areas have highly variable responses resembling those obtained from Poisson processes, other neurons can be much more regular. In the weakly electric fish *Apteronotus*, for example, the spike trains of primary sensory afferent neurons sensitive to amplitude modulations of the electrical field have a variability that is almost an order of magnitude smaller than that expected from a Poisson spike train on behaviorally relevant time scales (Ratnam and Nelson, 2000). These neurons are thought to encode information necessary for the detection of small prey, such as the water fleas on which the fish feeds. Computer simulations, behavioral observations, and electrophysiological recordings suggest that the firing rate of these cells will increase by only a few spikes per second during the 200 ms needed to detect the prey. An ROC analysis reveals that increases of 2–3 spk/s above baseline activity

can be detected with greater than 90% accuracy even if the probability of a false alarm is very low, 0.1% (Figure 2D). Such low false alarm rates ($p_{FA} \leq 0.001$) are constrained from behavioral observations showing that fishes almost never strike a nonexistent prey. As illustrated in Figure 2D, three spike trains of models designed to reproduce the short-term variability of the experimental spike trains cannot reproduce these results. The first model (squares) reproduces only the mean firing rate of the afferents, while the second and third models also reproduce the interspike interval distribution (diamonds) or the joint statistical distribution of two successive interspike intervals (triangles), respectively. The regularity and statistical structure of the spike trains over at least three firing cycles is therefore responsible for this unusually low detection threshold.

Instantaneous Rate Coding

Stimuli that vary on a fast time scale—comparable to the 200 ms observation window introduced in the last section—cannot be encoded by the mean spike count alone. Such stimuli are ubiquitous in the sensory environment of many animals. Motion of an object or self-motion, for example, result in rapid changes in light intensity across the visual field. Sound stimuli used for communication or localization correspond to rapidly varying changes in air pressure. In the case of weakly electric fishes considered in the last section, time-varying electrical field amplitude modulations occur as the fish moves through an electrically dense environment in water.

Such time-varying stimuli could be encoded by time-varying changes in the instantaneous firing rate of a neuron, even if the precise timing of spikes does not play an essential role in the process (Gabbiani and Koch, 1999). Consider, for example, the Poisson spike train model of the previous section with a spontaneous rate $\bar{\lambda} = 30$ spk/s. We assume that changes in the instantaneous firing rate from its mean value, $\bar{\lambda}$, are caused by changes of the stimulus, $s(t)$, from its mean value, \bar{s} ,

$$\lambda(t) = \alpha(s(t) - \bar{s}) + \bar{\lambda} \quad (3)$$

The constant α is a conversion factor between stimulus and firing rate, and $\lambda(t)$ is assumed to be positive. In the following discussion the stimulus will usually be assumed to have zero mean, i.e., $\bar{s} = 0$.

How much information does such a spike train convey about the stimulus? Using an approach analogous to that introduced in the last section, this question can be addressed by presenting a random stimulus $s(t)$ and estimating it from the spike train (Figure 3A). Because $s(t)$ varies randomly in time, the estimate $s_{\text{est}}(t)$ will also have to vary in time to track $s(t)$. Thus, this estimation problem is more complex than the detection problem considered in the last section. It is customary to minimize the root mean square error between the stimulus and its estimate,

$$\varepsilon = \langle (s(t) - s_{\text{est}}(t))^2 \rangle^{1/2} \quad (4)$$

where the average is taken over the stimulus presentation interval (Figure 3A). It is much more difficult to find an optimal estimate $s_{\text{est}}(t)$ from the spike train than it is to find an optimal classification strategy based on the spike count. A simplification is therefore

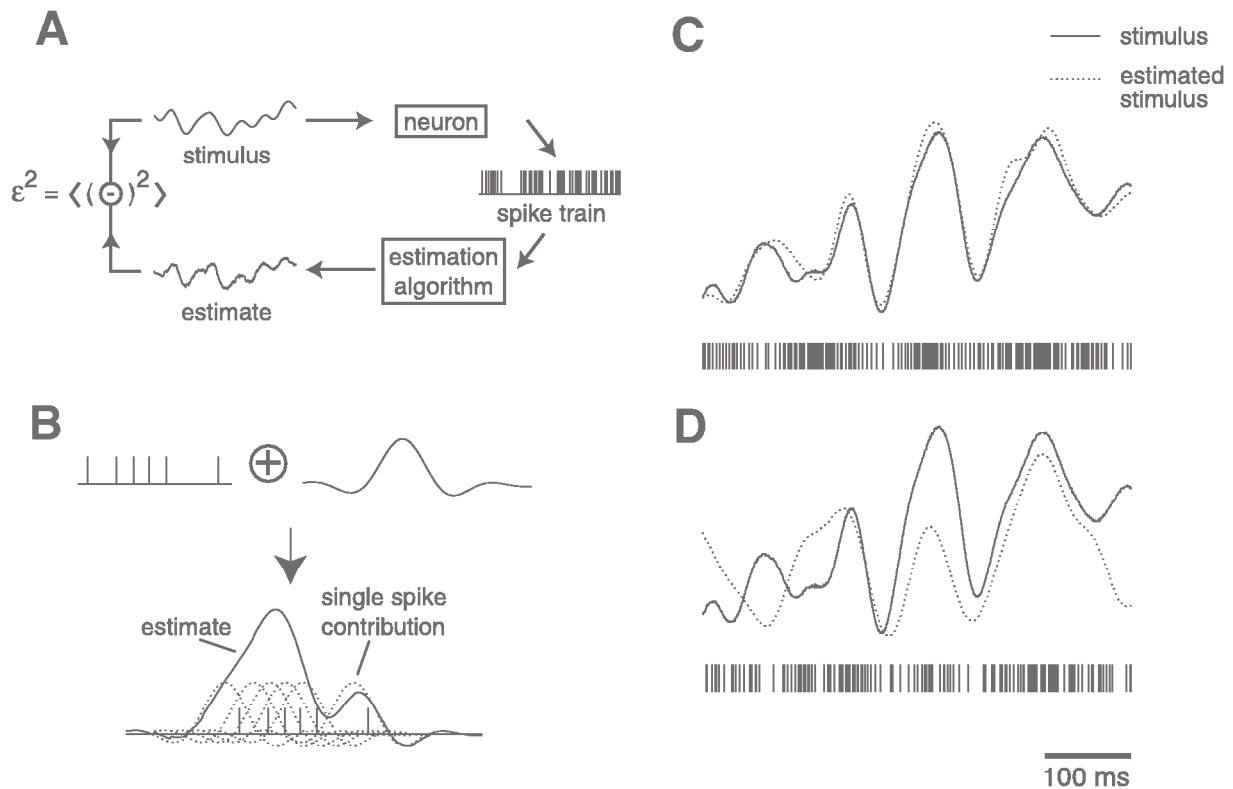


Figure 3. A, Stimulus estimation is performed using a linear algorithm (see B) that is based on a comparison of the stimulus and its estimate aimed at minimizing the mean square error between the two (Equation 4). B, The linear algorithm consists in taking a spike train (left) and placing a waveform (right) around each spike. Linear superposition of these waveforms (bottom) yields the estimate. The waveform is chosen to minimize the mean

square error between stimulus and estimate (see A). C, Estimation of a random amplitude modulation from the spike train of an amplitude sensitive afferent neuron in *Eigenmannia* (mean firing rate 314 spk/s). D, Same stimulus estimated from a Poisson spike train encoding the stimulus according to Equation 3 at the same mean firing rate as in C.

made by looking only at estimates obtained from linear superpositions of a waveform, $h(t)$, around each spike. If $r(t) = \sum_i \delta(t - t_i)$ is a sum of delta functions representing the sequence of spikes at times $\{t_i\}$, the estimated stimulus is assumed to be of the form

$$\begin{aligned} s_{\text{est}}(h, t) &= \int h(t - t_0) r(t_0) dt_0 - \bar{r} \int h(t_0) dt_0 \\ &= \sum_i h(t - t_i) - \bar{r} \int h(t_0) dt_0 \end{aligned} \quad (5)$$

where \bar{r} is the mean firing rate of the cell and the second term ensures that $s_{\text{est}}(t)$ has zero mean value, as was assumed for $s(t)$ (Figure 3B). Under this assumption, the optimal waveform, $h(t)$, minimizing the root mean square error in Equation 4 can be obtained by standard statistical and Fourier transform techniques. The minimum value obtained for the root mean square error, ε , is usually normalized by the stimulus standard deviation, σ , which corresponds to chance guessing (i.e., to the error obtained in Equation 4 when $s_{\text{est}}(t) = \bar{s} = 0$). Figure 3C illustrates the result of this estimation procedure using the spike train of an amplitude-sensitive afferent obtained in response to a random electrical field amplitude modulation in a second species of weakly electric fishes, *Eigenmannia*. From the spike train, the amplitude modulation could be estimated with an error $\varepsilon/\sigma = 0.17$. Equivalently, our ideal linear observer could reproduce 83% of the stimulus standard deviation using a single spike train.

In contrast, estimation of the same stimulus using a spike train generated using a Poisson process and Equation 3 is considerably less accurate (only 25% of the stimulus standard deviation is recovered; Figure 3D), because a Poisson process is more variable than the spiking of *Eigenmannia* afferents (Kreiman et al., 2000). Thus, as in the signal detection case, spike train variability plays an important role in stimulus estimation. Other factors that play a significant role in the encoding capacity of the instantaneous firing rate are the contrast of the stimulus (or its standard deviation σ ; typically, higher contrasts result in larger firing rate modulations and thus better encoding), the mean firing rate of the cell (higher mean firing rates lead to a finer temporal sampling of the stimulus), and the cutoff frequency of the stimulus (accurate encoding is possible only when temporal stimulus frequencies are well below the mean firing rate of the cell).

The assumption relating spike train and stimulus estimate by a linear transform embodied in Equation 5 works very well in practice when the encoding of the stimulus by the spike train can be described by equations analogous to Equation 3. This result can be justified theoretically (Gabbiani and Koch, 1999). In contrast, no systematic studies have been carried out on the effect of nonlinear relations between stimulus and firing rate; only a few scattered examples have been examined (Gabbiani and Koch, 1999; Haag and Borst, 1998).

Rate Coding in Neural Systems

Instantaneous and mean firing rate codes have been extensively characterized in a variety of different neuronal systems. In the following discussion, we will highlight a few directions of investigation and some open questions relevant to the subject.

Starting in the late 1940s, signal detection methods have been applied to characterize the information conveyed by neuronal spike trains, along the lines prescribed earlier in this article (see Parker and Newsome, 1998). The investigation of neuronal signals carried by optic nerve fibers of the horseshoe crab *Limulus* was one of the earliest examples of work on this topic (see Parker and Newsome, 1998). Over the next 30 years, signal detection methods were extended to the activity of sensory neurons in the early auditory, somatosensory, and visual pathways of vertebrates. The variability of retinal ganglion cell spike trains, for example, has been exten-

sively investigated in an attempt to explain its impact on the encoding reliability of visual signals (Parker and Newsome, 1998).

More recently, signal detection methods have been applied to neurons in cerebral cortical areas (visual and somatosensory, for instance) of monkeys trained to perform discrimination tasks. In some cases, the reliability of neuronal firing could be compared to the behavioral accuracy of the animal performing the task. These results, together with analyses of variability and correlation across cells, have led to neural models of signal encoding that can account for the animal's behavior (see Parker and Newsome, 1998). The neural mechanisms underlying behavioral selection in those discrimination tasks, however, remain difficult to test experimentally.

In the cockroach, directional escape responses to wind stimuli are thought to rely on the mean firing rate of 14 giant interneurons (GIs). Several models that could in principle explain escape behaviors on the basis of the mean firing rate of GIs have been tested by directly manipulating them through current injections (Levi and Camhi, 2000). The results of these experiments suggest that a directional average of the GIs' mean firing rate is the most accurate description of the behavior. Mean firing rate codes across population of neurons have also been shown to play similar roles in vertebrate neurons, in the generation of visual saccades in the superior colliculus of monkeys, and in the generation of limb movements in motor cortical areas (Sparks, Kristan, and Shaw, 1997).

Given that in the engineering literature signal estimation is usually considered a close relative of signal detection (Poor, 1994), it is perhaps surprising that it has been applied to neural spike trains only within the past 10 years. Estimation of time-varying stimuli along the lines developed earlier in this article has shown that single spike trains of sensory neurons can accurately convey information about time-varying stimuli, although the results are usually less spectacular than those shown in Figure 3B (Borst and Theunissen, 1999). At present, these methods have been applied mainly to invertebrate and lower vertebrate preparations. Mechanisms of encoding across multiple neurons and their relation to behavior have received little attention so far (but see Stanley, Li, and Dan, 1999).

In contrast, instantaneous firing rate codes have been extensively studied by characterizing how stimulus attributes are encoded in the instantaneous firing rate of neurons through generalizations of Equation 3. Such models are particularly well developed for the early visual pathways of mammals, from the retina to early visual cortical areas (Dayan and Abbott, 2001).

Discussion

Mean and instantaneous firing rate codes are undoubtedly the best documented and best understood way by which neurons transmit information. Several other codes have also been studied, among them the mechanisms of coincidence detection in auditory processing (Pena and Konishi, 2001). More elaborate coding schemes are likely to be found, particularly across populations of neurons, although the highly sophisticated codes at the heart of information theory seem unlikely to find a place in describing the signaling repertoire of sensory and motor neurons.

One question that has long intrigued neuroscientists is whether the spike train variability usually observed in neurons using rate coding also carries further sensory information (Bullock, 1970). This question is difficult to answer rigorously. In the case of the cockroach, the pattern of spikes in GIs does not appear to play a role in determining escape behaviors (Liebenthal, Uhlmann, and Camhi, 1994). On the other hand, it has been suggested that in electric fishes, coincidence detection could be used to integrate the information conveyed by the amplitude-sensitive receptors described in this article and in Berman and Maler (1999). Thus, neurons might use a combination of different codes simultaneously at different levels of a neuronal circuit.

Road Map: Neural Coding

Related Reading: Population Codes; Sensory Coding and Information Transmission

References

- Borst, A., and Theunissen, F. E., 1999, Information theory and neural coding, *Nature Neurosci.*, 2:947–957. ♦
- Berman, N. J., and Maler, L., 1999, Neural architecture of the electrosensory lateral line lobe: Adaptations for coincidence detection, a sensory searchlight and frequency-dependent adaptive filtering, *J. Exp. Biol.*, 202:1243–1253.
- Bullock, T. H., 1970, The reliability of neurons, *J. Gen. Physiol.*, 55:565–584. ♦
- Dayan, P., and Abbott, L. F., 2001, *Theoretical Neuroscience*, Cambridge, MA: MIT Press. ♦
- Gabbiani, F., and Koch, C., 1999, Principles of spike train analysis, in *Methods in Neuronal Modeling: From Synapses to Networks*, 2nd ed. (C. Koch and I. Segev, Eds.), Cambridge, MA: MIT Press, pp. 313–360. ♦
- Haag, J., and Borst, A., 1998, Active membrane properties and signal encoding in graded potential neurons, *J. Neurosci.*, 18:7972–7986.
- Kreiman, G., Krahe, R., Metzner, W., Koch, C., and Gabbiani, F., 2000, Robustness and variability of neuronal coding by amplitude-sensitive afferents in the weakly electric fish *Eigenmannia*, *J. Neurophysiol.*, 84:189–204.
- Levi, R., and Camhi, J. M., 2000, Population vector coding by the giant interneurons of the cockroach, *J. Neurosci.*, 20:3822–3829.
- Liebenthal E., Uhlmann, O., and Camhi, J. M., 1994, Critical parameters of the spike trains in a cell assembly: Coding of turn direction by the giant interneurons of the cockroach, *J. Comp. Physiol. A*, 174:281–296.
- Parker, A. J., and Newsome, W. T., 1998, Sense and the single neuron: Probing the physiology of perception, *Annu. Rev. Neurosci.*, 21:227–277. ♦
- Pena, J. L., and Konishi, M., 2001, Auditory spatial receptive fields created by multiplication, *Science*, 292:249–252.
- Poor, H. V., 1994, *An Introduction to Signal Detection and Estimation*, New York: Springer-Verlag. ♦
- Ratnam, R., and Nelson, M. E., 2000, Nonrenewal statistics of electrosensory spike trains: Implications for the detection of weak sensory signals, *J. Neurosci.*, 20:6672–6683.
- Sparks, D. L., Kristan, W. B., and Shaw, B. K., 1997, The role of population coding in the control of movement, in *Neurons, Networks, and Motor Behavior* (P. S. G. Stein, S. Grillner, A. I. Selverston, and D. G. Stuart, Eds.), Cambridge, MA: MIT Press, pp. 21–32. ♦
- Stanley, G. B., Li, F. F., and Dan, Y., 1999, Reconstruction of natural scenes from ensemble responses in the lateral geniculate nucleus, *J. Neurosci.*, 19:8036–8042.

Reaching Movements: Implications for Computational Models

Paul Cisek and John F. Kalaska

Introduction

Computational models are playing an increasingly important role in the study of biological motor control. In the first edition of this *Handbook*, the article on reaching movements (Kalaska, 1995) reviewed a range of computational models of visually guided movements and discussed their implications for cerebral cortical mechanisms of motor control. Here, we take the opposite approach. We discuss a number of issues that are emerging from neurophysiological studies of motor control and their implications for model development. We present these issues and implications as a set of challenges for computational models that hope to meet the demands of both functional competence and biological plausibility.

Planning Movement

Much of the theoretical background for computational models of the motor control system comes from engineering. Engineering practice usually solves a problem by breaking it down into a set of subproblems, each solved by a dedicated and distinct module. For example, a central distinction that motor control models inherit from engineering is that between planning and execution (Figure 1A). However, while this distinction may appear obvious from a robotics perspective because it is implied by the statement of the problem of control, it is not necessarily the most appropriate description for the organization of the biological motor system.

Neurophysiological evidence does not support a rigid separation between planning and execution at either the single-cell level or the population level (Kalaska, Sergio, and Cisek, 1998). Neurons that become active during movement preparation are distributed throughout the premotor and motor regions as well as in parietal regions, and those same cortical areas exhibit movement-related activity during execution. Motor imaging studies show that many of the same cortical areas are activated whether the subject is ac-

tually performing a movement or merely imagining it. Furthermore, even during execution of the movement itself, extensive representations of higher-order movement parameters and “early” sensorimotor transformations can be seen, especially in areas outside of primary motor cortex (Shen and Alexander, 1997a, b; Wise et al., 1997). Finally, correlates of both motor planning and execution processes can often be found in the activity of single cells, whose association with motor output changes in time from more abstract aspects of the task to more limb movement-related parameters (Crammond and Kalaska, 2000; Shen & Alexander, 1997a, b), as if single cells tended to shift functionally from the planning to the execution boxes of traditional models.

The functional distinctions most useful in understanding the organization of the biological motor control system may not be those that have proved most useful for traditional engineering methods. The general organization of the motor control system may in fact not resemble the serial input-to-output hierarchy of traditional models (Figure 1A) but instead may consist of parallel systems for *action specification* and *action selection* (Kalaska et al., 1998) (Figure 1B). Action specification includes all mechanisms involved in the computation of the parameters of motor actions. For a reaching movement, this begins with the processing of sensory information defining parameters such as distance to target and required grip size, and continues even during movement execution with on-line modification of the hand trajectory and joint torques. Because it continues even during movement itself, action specification incorporates processes that are often separated into the planning and execution stages of many computational models. Action selection includes all the mechanisms that choose between the various actions that are possible at a given moment. For a reaching movement, it encompasses such processes as attentional mechanisms that orient the eyes toward potential targets and select the ones most relevant at a given moment, cognitive mechanisms which decide the most appropriate response based on prior reinforcement, and

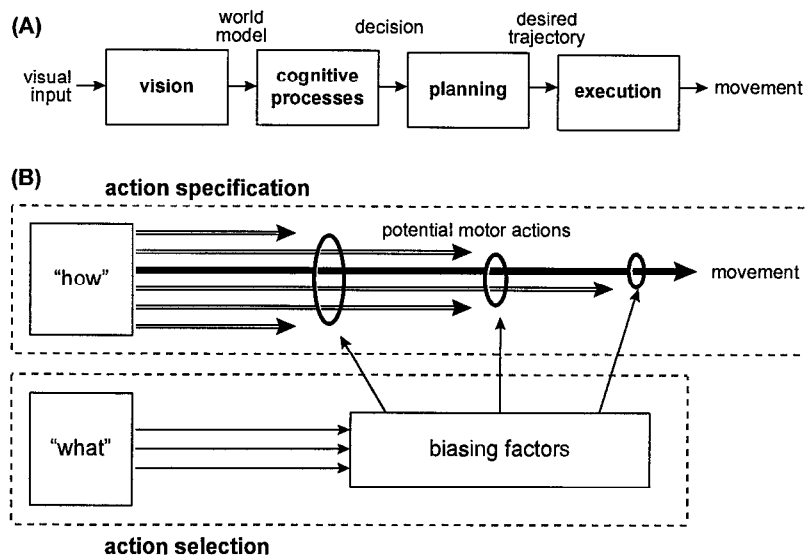


Figure 1. A, The traditional “sequential processing” model of visually guided behavior. In this model, visual input is used to construct a model of the world that is used to make decisions. After decisions are made, a desired trajectory is generated and executed. B, Schematic representation of the “specification and selection” architecture for visually guided behavior. Under this view, visual information has two different roles: specifying the parameters of potential motor actions, and defining criteria that bias competition among those potential actions until a single action is selected. These biasing factors include attention, behavioral relevance, prior reinforcement, required effort, behavioral context, learned associations, motivations, long-term behavioral objectives, desired outcomes, and any other factor that influences action selection. The processes of specification and selection occur in parallel and continue even during overt movement. A striking feature of this architecture is the absence of a central model of the visual world.

even mechanisms that abort or switch ongoing actions if the need arises.

The processes of selection and specification need not occur in a serial order but can instead both operate in parallel. Several potential motor actions may begin to be specified by the available sensory information through parallel sensorimotor transformations, allowing multiple responses to be “primed” for action (Cisek and Kalaska, 2002). At the same time, selection mechanisms using information on object identity and the organism’s objectives influence attentional and decision-making mechanisms, which select out the action most appropriate at the time. While the selected action is performed, other alternatives may not be discarded but instead may be maintained and updated in case the need arises to switch the course of action. Such a parallel architecture allows the flexibility of behavior required by the changing demands of a natural environment. This perspective argues against computational models with a strict sequential hierarchical structure and supports models that emphasize dynamic interactions between different brain regions.

Because the tasks of action specification and action selection impose different demands on the processing of sensory information, they likely involve at least partially independent neuronal systems. Action specification requires information about the spatial relationships between body segments and objects in the environment with which the organism is interacting. In contrast, action selection emphasizes information about the nature and identity of external objects in order to evaluate the possible consequences of acting on these objects. These differing demands correspond well to the differing properties of the dorsal and ventral visual streams (Milner and Goodale, 1995), suggesting that these sensory systems evolved initially to serve the needs of the motor system, not to generate an internal representation of the world.

The general architecture of action specification and selection has much in common with Arbib’s *SCHEMA THEORY* (q.v.). According to schema theory, visual information activates perceptual mechanisms (“perceptual schemas”), such as object localization and size recognition, that provide the information required to prepare specific action-oriented mechanisms (“motor schemas”), such as hand transport and hand preshaping, which are selected and released for execution according to learned contextual clues. Instead of a serial decomposition of action into planning and execution modules, schema theory suggests that behavior consists of the interplay be-

tween different parallel perception-action schemas that the organism applies when interacting with its environment.

Trajectory Generation

Another consequence of assuming separate planning and execution systems is the assumption that there must exist a representation of the motor plan that links them together. Different theories subscribe to this assumption to different degrees. For example, some theories propose that the motor plan takes the form of a representation of the complete time sequence of states that the execution system will pass through, a “desired trajectory” that is prespecified before movement begins. This is particularly important for explicit optimization models, which must know about the final states of movement before adjusting parameters of the intermediate states (see *OPTIMIZATION PRINCIPLES IN MOTOR CONTROL*).

To date, however, there has been no compelling neurophysiological evidence for an explicit representation of desired trajectories prior to the execution phase of motor tasks. One simple way to assess trajectory preplanning is to compare activity between instructed-delay tasks in which complete information about the metrics of an upcoming movement is presented prior to the instruction to initiate movement (“Go signal”), and reaction-time tasks in which the metrics are specified at the same time as the Go signal. One expects that during instructed-delay tasks, preplanning can take place as soon as movement metrics are specified, and need not be recapitulated after the Go signal. Although some of the predicted “neural savings” can be seen, especially in premotor cortex, most of the movement-related activity in premotor and primary motor cortex is relatively unaltered by the prior information, suggesting that most of the spatiotemporal details of the trajectory are generated dynamically as the movement unfolds (Crammond and Kalaska, 2000).

This is supported by another study (Shen and Alexander, 1997a, b), which dissociated the direction of limb movement and the direction of visual feedback guiding action (cursor motion on a screen). It was found that most of the activity in premotor and primary motor cortex during the delay period reflected the direction of cursor motion, and that a representation of the direction of limb motion became prominent only after the Go signal. This implies that only a relatively abstract representation of motor output is preplanned, even during well-practiced behaviors, whereas limb-

specific signals are expressed at the time of movement. This implication does not contradict the possibility that certain high-level aspects of complex movements such as via points or sequence elements can be preplanned (see *SEQUENCE LEARNING*).

One might also question the concept of preplanned trajectories from an evolutionary perspective. It is unlikely that very primitive creatures preplanned and optimized the details of their actions before executing them. Instead, they generated movement on-line, adjusting movement details based on information fed back in real time. This kind of solution can produce acceptable results for many kinds of movements without requiring complex internal models of the dynamics of the controlled object. With a closed-loop system, the dynamics of the controlled object directly participate in the modification of the time course of the control signals. It is likely that such a simple strategy set the ancestral foundation for motor control.

Indeed, recent evidence suggests that movements are fine-tuned and adjusted using on-line sensory information during the movement itself. When a target of a reaching movement unexpectedly jumps during the course of the reach, subjects adjust automatically, even when the jump is not consciously perceived (Desmurget et al., 1999). Such on-line correction processes appear to involve the parietal cortex (Milner and Goodale, 1995; Desmurget et al., 1999) and are presumed to operate at all times during natural movements. In fact, without on-line correction, most normal activity would be very inaccurate because the world around us is always changing. For example, one could never catch a baseball by preplanning one's position and glove placement only when the ball is first hit or thrown. Instead, constant adjustments are necessary as the ball approaches, with the subject using sensory information about the ball's motion as well as feedback about the subject's own movements.

Nevertheless, the ability of the motor system to adjust movements on-line does not imply that it operates purely in the closed-loop manner of a standard feedback controller. That would not be possible with the conduction delays inherent in the system. To compensate for such delays, the motor system is able to learn specific movement contexts and to predict the state of the system when performing in each context (see *SENSORIMOTOR LEARNING*). For example, the system may learn an internal "forward model" that predicts, for a given movement context, what the outcome of a particular set of motor commands will be. With such a forward model, compensation for an expected perturbation can occur even before the perturbation causes any overt movement errors. As the system becomes more and more familiar with a given movement context, its dynamics will converge on the production of those commands that minimize the errors most relevant for the given task. However, although the resulting trajectory may be described as optimal (with respect to some criterion such as end-point error), this optimization arises slowly over a series of repeated action-perception cycles. It does not occur before movement begins through the explicit optimization of a desired trajectory. Instead, trajectories are produced during movement through the interplay of dynamics involving overt feedback and forward models, and it is the set of parameters implicitly defining these dynamics that is optimized during learning over many repeated movements.

Temporal Features of Cortical Activity

Many motor control models are described in terms of abstract computational elements whose activity over time does not clearly correspond to any of the neural activity profiles observed in the nervous system. However, the temporal features of cell activities in movement-related cortical areas should be very informative about the evaluation and modification of motor control models.

Neural activities in primary motor cortex exhibit a variety of complex shapes that appear to be importantly related to the kinematic and kinetic requirements of the task at hand (Kalaska et al., 1989; Fetz, 1992; Sergio and Kalaska, 1998). Since even the cells that project directly to spinal motor neurons exhibit complex temporal response profiles that do not explicitly code the ensuing EMG, it is clear that the descending command is not simple. These studies demonstrate that many details of the time-varying aspects of movement are already evident at the cortical level and thus are not all computed at the spinal cord, despite the elegance of proposed schemes for doing so (see *EQUILIBRIUM POINT HYPOTHESIS*).

These various temporal response profiles must certainly be informative for computational models. For example, area 4 cells exhibit several different kinds of response profiles, including phasic, tonic, and phasic-tonic responses (Kalaska et al., 1989; Fetz, 1992). The phasic-tonic cells are the most load sensitive of these and are most often found in the deeper layers from which the pyramidal tract projection originates. They form the largest proportion of corticomotor-neuronal (CM) cells that directly project to spinal motor neurons (Fetz, 1992). Their temporal response pattern is clearly related to the dynamical requirements of different tasks (Sergio and Kalaska, 1998). In contrast, phasic cells tend to show much less load sensitivity, are more often found in superficial layers, and are almost never CM cells. There are also important differences between the activities of cortical neurons in different regions and different layers, as well as during different movement contexts. For example, cells in parietal area 5 exhibit much less sensitivity to loads than do cells in primary motor cortex, especially at the population level (Kalaska et al., 1990).

Such observations led Bullock, Cisek, and Grossberg (1998) to outline a circuit model that incorporates neural elements whose activity resembles these observed temporal patterns. According to this model, the load-sensitive phasic-tonic cells in area 4 assemble a descending command by integrating a directional signal from phasic cells and combining it with launching and braking pulses. Area 5 tonic cells combine a load-sensitive corollary discharge signal from area 4 and a load-sensitive feedback from stretch receptors to yield a load-insensitive position representation. In the model, interactions between area 4 and area 5 cells result in the on-line generation of reaching trajectories. By assigning specific functional roles to observed cell profiles, such models make specific predictions that can be tested in future neurophysiological experiments.

Overlapping Polymodal Gradients

Motor control models usually consist of discrete functional modules, with specific computational roles assigned to specific elements. However, a striking feature of cortical neurophysiology, at least in the motor system, is the absence of well-delineated borders separating populations of cells with different functional properties. Instead, one observes gradual transitions in cell properties as one moves across the cortical surface.

Within a local area, cells do not neatly partition into separate classes or "types," but rather form a complex continuum exhibiting different mixtures of properties (Caminiti, Ferraina, and Battaglia-Mayer, 1998). As we move along the cortical surface, the mixture of cell properties changes gradually. Moving medially or laterally in the motor cortex, we find cells whose activity relates to different body parts. Moving rostrally from the central sulcus over the precentral gyrus, we find progressively more phasic and less load-sensitive cells and more correlation with abstract task information than with the details of movement kinematics or dynamics. In the postcentral gyrus, there is a reciprocal gradient of progressively less movement-related and more preparation-related activity as we move caudally along area 5 and into the deep intraparietal sulcus. These opposing gradients in pre- and postcentral areas are paral-

leed by a connectivity pattern, with neurophysiologically similar regions across the central sulcus being reciprocally connected. As already mentioned, gradual transitions are also observed in the time domain, with cell properties changing during the course of a movement trial.

In addition to smooth gradients of changing cell properties, some movement-related cortical areas also exhibit a great deal of polymodality. For example, in premotor and posterior parietal areas, cells are found that are sensitive to a variety of sensory and motor information. Cells respond to different degrees to salient retinal inputs, especially to objects and motion in the region of space near the monkey, and are modulated by direction of gaze, direction of attention, direction of intended movement, limb configuration, and cutaneous contact, among other factors (Caminiti et al., 1998). Modeling studies suggest that these combinations of signals converging on single cells are appropriate to effect a sensorimotor coordinate transformation. However, neurophysiological studies routinely fail to find a significant population of cells whose activity explicitly encodes the output of that transformation in a unique coordinate system. Instead, the output may be implicitly embedded in the distributed pattern of activity across the population, and extracted by appropriate decoding mechanisms (Pouget and Sejnowski, 1997). This diversity of polymodal properties indicates that computational models based on strict engineering principles and homogeneous coordinate systems (for instance, inverse differential kinematics from instantaneous hand velocity to instantaneous rates of change of muscle lengths) may have some heuristic value but do not capture the true nature of cerebral cortical motor control processes.

A model being developed by Yves Burnod and colleagues (Burnod et al., 1999) takes on the challenge of addressing the distributed nature of movement-related information in the cerebral cortex. In their framework, learned associations between combinations of sensory information (such as target position, gaze direction, and current limb configuration) and motor commands (such as gaze shifts or arm movements) are retrieved by “match” units, and the combination defining the movement that is most appropriate to the given task is selected by “condition” units on the basis of prior reinforcement. These match and condition units are distributed throughout the cortex in a continuum reflecting the possible combinations of information necessary to guide movement in various contexts. Thus, the model suggests that the overlapping gradients of polymodal activities in frontal and parietal regions are not merely a biological nuisance masking the true functional decomposition of motor control but are instead the basis of the motor system’s strategy for flexibly integrating information for the demands of different tasks.

Discussion

Models that attempt both to solve interesting computational problems and to explain biological data face many challenges. These challenges come from diverse directions, including constraints imposed by the laws of physics, neurophysiological data, psychophysics, the evolution and development of the nervous system, and the impressive flexibility and adaptability of movement. The best way for progress to be made in such an endeavor is through a combined modeling and empirical approach. Models should be

viewed as stepping stones in such a process, complementary to the experiments that they help to guide and that in turn help to modify and refine the models.

Road Map: Mammalian Motor Control

Related Reading: Arm and Hand Movement Control; Cerebellum and Motor Control; Eye-Hand Coordination in Reaching Movements; Limb Geometry, Neural Control; Robot Arm Control

References

- Bullock, D., Cisek, P., and Grossberg, S., 1998, Cortical networks for control of voluntary arm movements under variable force conditions, *Cereb. Cortex*, 8:48–62.
- Burnod, Y., Baraduc, P., Battaglia-Mayer, A., Guigon, E., Koechlin, E., Ferraina, S., Lacquaniti, F., and Caminiti, R., 1999, Parieto-frontal coding of reaching: An integrated framework, *Exp. Brain Res.*, 129:325–346.
- Caminiti, R., Ferraina, S., and Battaglia-Mayer, A., 1998, Visuomotor transformations: Early cortical mechanisms of reaching, *Curr. Opin. Neurobiol.*, 8:753–761. ♦
- Cisek, P., and Kalaska, J. F., 2002, Simultaneous encoding of multiple potential reach directions in dorsal premotor cortex, *J. Neurophysiol.*, 87:1149–1154.
- Crammond, D. J., and Kalaska, J. F., 2000, Prior information in motor and premotor cortex: Activity during the delay period and effect on pre-movement activity, *J. Neurophysiol.*, 84:986–1005.
- Desmurget, M., Epstein, C. M., Turner, R. S., Prablanc, C., Alexander, G. E., and Grafton, S. T., 1999, Role of the posterior parietal cortex in updating reaching movements to a visual target, *Nature Neurosci.*, 2:563–567.
- Fetz, E. E., 1992, Are movement parameters recognizably coded in the activity of single neurons? *Behav. Brain Sci.*, 15:679–690. ♦
- Kalaska, J. F., 1995, Reaching movements: Implications of connectionist models, in *The Handbook of Brain Theory and Neural Networks* (M. A. Arbib, Ed.), Cambridge, MA: The MIT Press, pp. 788–793. ♦
- Kalaska, J. F., Cohen, D. A. D., Hyde, M. L., and Prud’homme, M. J., 1989, A comparison of movement direction-related versus load direction-related activity in primate motor cortex, using a two-dimensional reaching task, *J. Neurosci.*, 9:2080–2102.
- Kalaska, J. F., Cohen, D. A. D., Prud’homme, M. J., and Hyde, M. L., 1990, Parietal area 5 neuronal activity encodes movement kinematics, not movement dynamics, *Exp. Brain Res.*, 80:351–364.
- Kalaska, J. F., Sergio, L. E., and Cisek, P., 1998, Cortical control of whole-arm motor tasks, in *Sensory Guidance of Movement: Novartis Foundation Symposium No. 218* (M. Glickstein, Ed.), Chichester, UK: Wiley, pp. 176–201. ♦
- Milner, A. D., and Goodale, M. A., 1995, *The Visual Brain in Action*, London: Oxford University Press. ♦
- Pouget, A., and Sejnowski, T. J., 1997, Spatial transformations in the parietal cortex using basis functions, *J. Cogn. Neurosci.*, 9:222–237.
- Sergio, L. E., and Kalaska, J. F., 1998, Changes in the temporal pattern of primary motor cortex activity in a directional isometric force versus limb movement task, *J. Neurophysiol.*, 80:1577–1583.
- Shen, L., and Alexander, G. E., 1997a, Neural correlates of a spatial sensory-to-motor transformation in primary motor cortex, *J. Neurophysiol.*, 77:1171–1194.
- Shen, L., and Alexander, G. E., 1997b, Preferential representation of instructed target location versus limb trajectory in dorsal premotor area, *J. Neurophysiol.*, 77:1195–1212.
- Wise, S. P., Boussaoud, D., Johnson, P. B., and Caminiti, R., 1997, Premotor and parietal cortex: Corticocortical connectivity and combinatorial computations, *Annu. Rev. Neurosci.*, 20:25–42. ♦

Reactive Robotic Systems

Ronald C. Arkin

Introduction

Reactive systems are a relatively recent development in robotics that has redirected artificial intelligence (AI) research. This new approach grew out of a dissatisfaction with existing methods for producing intelligent robotic response and a growing awareness of the importance of studying biological systems as a basis for constructing intelligent behavior. Reactive robots are often referred to as behavior-based robots; they are instructed to perform through the activation of a collection of low-level primitive behaviors. Complex physical behavior emerges through the interaction of the behavioral set and the complexities of the environment in which the robot finds itself, resulting in more rapid and flexible responses than are attainable through traditional methods of robotic control.

Some of the hallmark characteristics of purely reactive robotic systems include (Arkin, 1998):

1. *Behaviors are the basic building blocks.* A behavior in these systems is often a simple sensorimotor circuit, where sensory activity consists of providing necessary information to support low-level reactive motor response, such as avoiding obstacles, escaping from predators, being attracted to goals, etc.
2. *Abstract representational knowledge is avoided.* Creating and maintaining accurate representations of the world is a time-consuming error-prone process. Purely reactive systems do not maintain world models, instead reacting directly to the stimuli the world presents. This is particularly useful in highly dynamic and hazardous worlds, where the environment is unpredictable and potentially hostile.
3. *Animal models of behavior are often used as a basis for these systems.* Models from neuroscience, cognitive psychology, and ethology are used to capture the nature of the behaviors that are necessary for a robot's safe interaction with a hostile world.
4. *Demonstrable robotic results have been achieved.* These techniques have been applied to a wide range of robots, including six-legged walking robots, pipe-crawling robots, military robots, entertainment robots such as Sony's AIBO, mobile manipulators, dextrous hands, and entire herds of mobile robots. Because these systems are highly modular, they can be constructed incrementally from the bottom up by adding new behaviors to an existing repertoire. From an engineering perspective, this property is quite desirable, as it facilitates the growth and application of existing software and hardware systems to new domains.

Even more recently, hybrid reactive/deliberative robotic architectures have emerged that combine aspects of more traditional AI symbolic methods and their use of abstract representational knowledge with the responsiveness, robustness, and flexibility of purely reactive systems. Both purely reactive and hybrid architectures are discussed in this article.

Biological Basis for Reactive Robotic Systems

Many of the designers of reactive systems look to biology as a source of models for use in robots. Although these efforts are quite diverse, ranging from traditionally engineered systems to those dedicated to faithfully replicating biological behavior, this article reports on a few exemplars that have affected reactive and hybrid system design.

Action-oriented perception. Neuroscientists and psychologists, especially in the cognitive and ecological communities, have pro-

vided models for the relationships between perceptual activities and behaviors required for a particular task. One excellent example is presented in Arbib (1972). His model of action-oriented perception shows that what an agent needs to perceive is based on its need to act. This is a primary guiding principle in the design of reactive robots. The traditional computer vision community often views perception as a disembodied perceiver that interprets images without consideration for what the knowing agent needs to do. In contrast, the strong coupling between action and perception is one of the hallmarks of purely reactive robotic systems. Neisser has further developed these ideas in the context of cognitive psychology (see Arkin, 1998, for a review of those aspects relevant to robotic systems).

Ethological studies. A pressing question for reactive robotic system designers is just what behaviors are necessary or sufficient for a particular task and environment. Many of these researchers have turned to ethological studies as a source for behaviors that are relevant in certain circumstances. Specific models used in reactive robotic systems have been quite varied, including bird flocking, ant foraging, fish schooling, and cockroach escape, among others. One example involving toad detour behavior (Arbib and House, 1987) provided motivation and justification for the use of vector fields in reactive schema-based robot navigation (Arkin, 1998).

Coexistence of parallel planning and execution systems (hybrid systems). Norman and Shallice (1986) have modeled the coexistence of two distinct systems concerned with controlling human behavior. One system models "automatic" behavior and is closely aligned with reactive systems. This system handles automatic action execution without awareness, starts without attention, and consists of independent parallel activity threads (schemas). The second system controls "willed" behavior and expresses an interface between deliberate conscious control and the automatic system.

While purely reactive robotic systems are compatible with the modeled automatic system (e.g., Brooks, 1986), most hybrid robotic systems (e.g., Arkin, 1990; Gat, 1992) incorporate both willed (deliberative) and automatic (reactive) components in a manner somewhat consistent with the above model.

One problem confronting the reactive robotic systems designer is that much of the data reported by biological scientists is often presented statistically. Although this approach may be useful within the context of their home disciplines, it is important for process models to be constructed whenever possible to facilitate the adoption of this work into intelligent robotic systems (see NEUROETHOLOGY, COMPUTATIONAL).

Purely Reactive Robotic Systems

Reactive robotic systems originated in the cybernetic movement of the 1940s. Grey Walter (1953) developed an electronic "tortoise" capable of moving about the world, avoiding perceived threats and attracted to certain goals. Of special interest was the inclusion of changing goals regarding the robot's recharging station. When power was low, the tortoise was attracted to and docked with the recharger. When sufficient energy was acquired, it lost its "appetite" (charger attraction) and was repelled by it. There was no use of abstract representational constructs as later found in traditional AI; perception directly controlled motor action. Simple behaviors were created: head toward weak light, back away from strong light, and turn-and-push to avoid obstacles.

Braitenberg (1984) revived interest in this class of creatures. Using simple analog circuitry, he demonstrated that “creatures” could be built that manifested behaviors comparable to those found in animals, such as cowardice, aggression, love, exploration, and logic. These thought experiments in “synthetic psychology” showed that seemingly complex behavior could result from a collection of simple sensorimotor transformations.

Brooks (1986) was an early leader of the purely reactive robotic paradigm. His group pursued this approach with the development of subsumption architecture. He articulated the departure from classical AI and broke away from the sense-plan-act paradigm that dominated AI in the 1970–1980s as typified by robots like Shakey that used resolution theorem proving as its primary reasoning mechanism. This new position brought into question the role of representational knowledge in AI altogether. The subsumption architecture was biologically motivated only in the behaviorist sense, as it produced overt results that resembled the behaviors of certain insect systems but was unconcerned for the underlying biological mechanisms that produced them.

At about the same time that subsumption architecture appeared, other researchers were interested in pursuing parallels in biological and mechanical systems. A sort of cybernetics revival occurred. Studies produced by ethologists, neuroscientists, and others provided models that were used within reactive robotic systems. These researchers’ goals varied. For example, Arkin (1990) exploited these models for the purpose of constructing intelligent robotic systems, using interacting schemas as a basis for reactive robotic control systems design (see SCHEMA THEORY). Beer, alternatively, used robotic systems to demonstrate the fidelity of neuroscientific models (see LOCOMOTION, INVERTEBRATE). Significant conferences now exist dedicated to animal and computational systems relationships; an example is the conference whose proceedings are regularly published as *Simulation of Adaptive Behavior: From Animals to Animals*, by MIT press.

Figure 1 presents a simple reactive control system example. A robot controlled by this system wanders around avoiding collisions until it finds a path, which it then follows until it locates its goal. It consists of four behaviors: *avoid-obstacle* prevents the robot from colliding with anything; *wander-about* ensures movement in the absence of goal or path attraction; *stay-on-path* guides the robot down a hall or road to find the goal near the path’s end; and *move-to-goal* attracts the robot to the final goal. The perceptual strategies for each behavior are also depicted. The behavior coordination mechanism can be of several forms. Arbitration or action-selection mechanisms are typically found in subsumption-style architectures where only one behavior is active at any given time. This action-selection mechanism can be complex, involving extensive connections between behaviors for inhibition/suppression. The schematic representation of this mechanism is greatly simplified in this figure. Other coordinators may involve blending, as in schema-based reactive control systems, where all active behaviors contribute somewhat to the overall coordinated motion. Even combinations of different coordination mechanisms can be used to compose intelligent robotic behavior.

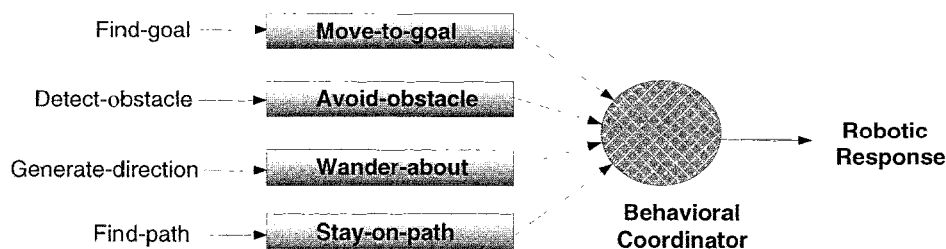


Figure 1. Example of a reactive control system.

Hybrid Reactive/Deliberative Robotic Systems

Hybrid architectures permit reconfiguration of reactive control systems based on available world knowledge, adding considerable flexibility over purely reactive systems. Dynamically reconfiguring the control system based on deliberation (reasoning over world models) is an important addition to the overall competence of general-purpose robots.

It should be recognized that purely reactive robotic systems are not appropriate for every robotic application. In situations where the world can be accurately modeled, where there is restricted uncertainty, and where there exists some guarantee of virtually no change in the world during execution (such as an engineered assembly workcell), deliberative methods are often preferred, since a plan can most likely be effectively carried out. In the real world, in which biological agents function, these prerequisites for purely deliberative planners do not exist. If roboticists hope to have their machines functioning in the same environments that we do, methods like reactive control are required. Many feel that hybrid systems capable of incorporating both deliberative reasoning and reactive execution are needed to deliver the full potential of robotic systems.

Arkin was among the first to advocate the use of both deliberative (hierarchical) and reactive (schema-based) control systems within the Autonomous Robot Architecture. Incorporating a traditional planner that could reason over a flexible and modular reactive control system, specific robotic configurations could be constructed that integrated behavioral, perceptual, and a priori environmental knowledge (Arkin, 1990). This system was tested on a wide range of applications, both indoors and outdoors.

Gat (1992) proposed a three-level hybrid system (Atlantis) incorporating a Lisp-based deliberator, a sequencer that handled failures of the reactive system, and a reactive controller. This system was fielded and tested successfully on Mars rover prototypes.

Perception and Reactivity

A fundamental guiding principle for purely reactive systems is that perceptual activities should always be viewed on the basis of motor needs (i.e., a *need-to-know basis*). A large body of mainstream computer vision research is concerned with the abstract task of image understanding, which usually is considered independently of a particular agent’s needs. Proponents of purely reactive control advocate that perception serves motor action, and that image interpretation algorithms must take this into account. Sensing strategies should be constructed that take advantage of the knowledge of underlying behavioral requirements. This eliminates the need to construct global representations of the world, an activity avoided in purely reactive robotic systems. By creating perceptual algorithms that extract only relevant information and that exploit expectations of what is necessary and sufficient to be perceived, efficient sensor processing is a natural consequence.

Hybrid approaches, nonetheless, are more consistent with the views of neuroscientists (e.g., Mishkin, Ungerleider, and Macko

1983) on *what* + *where* visual systems that account for the maintenance of spatial relationships in a more than purely reactive manner (see VISUAL SCENE PERCEPTION).

There are three ways in which reactive systems can utilize perceptual information: perceptual channeling (sensor fission), action-oriented sensor fusion, and perceptual sequencing. Perceptual channeling is straightforward: a motor behavior requires a particular stimulus for it to be invoked, so a single sensor system is created. A simple sensorimotor circuit results. There are numerous examples (e.g., Maes, 1990; Brooks, 1991).

Action-oriented sensor fusion (Arkin, 1993) permits the construction of representations (percepts) that are local to individual behaviors. Restricting the representation to the requirements of a particular behavior provides the benefits of reactive control while permitting more than one sensor to provide input, resulting in increased robustness.

Sometimes fixed action patterns require varying stimuli to support them over separations in time and space. As a behavioral response unfolds, it may be modulated by different sensors or different views of the world. Perceptual sequencing supports the coordination of multiple perceptual algorithms over time in support of a single behavioral activity. Perceptual algorithms are phased in and out, based on the needs of the agent and the environmental context in which it is situated.

Discussion

Space prevents an extensive survey of the wide range of reactive robotic systems; the reader is referred to Maes (1990), Brooks (1991), Mataric (1992), Effen and Shaw (1993), and Arkin (1998) for additional information and alternative perspectives. These methods have gained dramatically in popularity and utility since the mid-1980s and are being applied to robotic systems throughout the world.

Hybrid reactive/deliberative architectures have been created to address several of the potential shortcomings of purely reactive systems. They permit the incorporation of world knowledge and the construction of global representations, yet preserve the strength of reactive execution and responsiveness to environmental change.

Reading

John G. Holden and Guy C. Van Orden

Introduction

A skilled reader can recognize many thousands of printed words, each in a fraction of a second, with no noticeable effort. A child who is developmentally dyslexic does not easily acquire this skill. For a dyslexic child, recognizing a printed word as a particular word can be effortful to the point of frustration. Dyslexia may plague an otherwise bright and articulate child, and the fact of dyslexia illustrates how recognition of printed words as words is the crux of reading. Reading is not exclusively the recognition of printed words, but it is word recognition that sets reading apart from natural language. In effect, to become a reader is to master this special skill (Perfetti, 1985). A vast empirical literature concerns word recognition, and most "neural" networks of reading are models of word recognition.

One tool with which to diagnose dyslexia is a naming task that presents individual *pseudoword spellings*, such as "glurp," to be read aloud. A dyslexic child may have great difficulty with this

Road Map: Robotics and Control Theory

Related Reading: Potential Fields and Neural Networks; Visuomotor Coordination in Frog and Toad

References

- Arbib, M. A., 1972, *The Metaphorical Brain: An Introduction to Cybernetics as Artificial Intelligence and Brain Theory*, New York: Wiley.
- Arbib, M., and House, D., 1987, Depth and detours: An essay on visually guided behavior, in *Vision, Brain, and Cooperative Computation* (M. Arbib and A. Hanson, Eds.), Cambridge, MA: MIT Press, pp. 139–163.
- Arkin, R. C., 1990, Integrating behavioral, perceptual, and world knowledge in reactive navigation, *Robotics and Autonomous Systems*, 6:105–122.
- Arkin, R. C., 1993, Modeling neural function at the schema level: Implications and results for robotic control, in *Biological Neural Networks in Invertebrate Neuroethology and Robotics* (R. Beer, R. Ritzmann, and T. McKenna, Eds.), San Diego: Academic Press, pp. 383–410.
- Arkin, R. C., 1998, *Behavior-Based Robotics*, Cambridge, MA: MIT Press. ♦
- Braitenberg, V., 1984, *Vehicles: Experiments in Synthetic Psychology*, Cambridge, MA: MIT Press.
- Brooks, R., 1986, A robust layered control system for a mobile robot, *IEEE J. Robot. Automat.*, 2:14–23.
- Brooks, R., 1991, New approaches to robotics, *Science*, 13 Sept., pp. 1227–1232. ♦
- Effen, J., and Shaw, R., 1993, Ecological perspectives on the new artificial intelligence, *Ecol. Psychol.*, 4:247–270. ♦
- Gat, E., 1992, Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots, *Proc. AAAI-92*, pp. 809–815.
- Maes, P., Ed., 1990, *Designing Autonomous Agents*, Cambridge, MA: MIT Press/Elsevier, 1990. ♦
- Mataric, M., 1992, Integration of representation into goal-driven behavior-based robots, *IEEE Trans. Robot. Automat.*, 8:304–312.
- Mishkin, M., Ungerleider, L. G., and Macko, K. A., 1983, Object vision and spatial vision: Two cortical pathways, *Trends Neurosci.*, 6:414–417.
- Norman, D., and Shallice, T., 1986, Attention to action: Willed and automatic control of behavior, in *Consciousness and Self-Regulation: Advances in Research and Theory*, vol. 4 (R. Davidson, G. Schwartz, and D. Shapiro, Eds.), New York: Plenum Press, pp. 1–17.
- Walter, W. G., 1953, *The Living Brain*, New York: Norton.

task, and success in this nonword task is generally correlated with word reading skill. Trouble decoding the pronunciations of letter strings is a symptom of the most common form of dyslexia (Pennington, 1991). In this form, dyslexia is a specific problem in translating words' spellings into their phonology (roughly, their sounds or pronunciations). And the key to word recognition would seem to lie in the derivation of phonology from words' printed forms.

All written languages have systematic relationships between words' printed and spoken forms (Mattingly, 1992), so perhaps word recognition includes an analytic letter-by-letter process that translates spelling into phonology. This possibility has preoccupied reading scientists for over 100 years. Nevertheless, word recognition is not simply analytic. Evidence that supports an analytic hypothesis has always existed side by side with evidence that word recognition is synthetic (or holistic). The consequent synthetic/analytic debate defined reading theory throughout the twentieth century, and it provides the organizing theme for this article.

Early Reading Research

Nineteenth century studies introduced almost all topics of current reading research (Rayner and Pollatsek, 1989). A key early finding from eye-movement studies was that readers' eyes make a series of jumps in moving across a line of text, pausing for about a quarter of a second when fixated. Discrete eye movements implied the *fixation* (recognition) of individual printed words in natural reading. Another key development was the invention of the t-scope (or tachistoscope). A t-scope can flash individual words for a few milliseconds at a time, well within the range of fixation times observed in eye movements.

More recent t-scope studies, in the second half of the twentieth century, perpetuated the synthetic/analytic debate. Ulric Neisser's seminal book *Cognitive Psychology* (1967) includes a review of this debate concerning word recognition. For example, pseudoword spellings, such as "glurp," that obey the letter-to-sound pronounceability patterns of English are more easily recognized and recalled than are random strings of letters. The advantage for pronounceable letter strings suggests an analytic process. A more contemporary finding, however, indicates that word recognition could be synthetic. Words that are flashed for a fraction of a second (and then replaced by a "pattern mask" of letter features) are more accurately reported than are their component letters presented individually under the same extreme conditions, a phenomenon known as the *word superiority effect*.

Dual-Route Theories

Dual-route theories emerged in the 1970s with the advent of cognitive psychology. As the name implies, traditional dual-route theories were an ad hoc resolution of the synthetic/analytic debate. Both options were included as separate processing modules (mechanisms). Early dual-route theories were important in this regard because they moved past a contentious theoretical debate that could not be resolved empirically.

The two modules of dual-route theories accomplish word recognition in two different ways. Skilled readers reading frequently encountered words rely on a fast, synthetic, lexical module to translate a visual representation into an entry in the lexicon, the mind's dictionary. The speed of access to dictionary entries in the lexical module is determined by the relative frequency with which a word appears in print. Word frequency is estimated by counting the occurrence of each word (per million words) in large samples of text. Higher-frequency words are more readily available in the lexicon. (Different dual-route theories propose different frequency-sensitive mechanisms.)

By contrast, novice readers and skilled readers who encounter a novel word recognize words via a slow, analytic, sublexical module. At the heart of the sublexical module are rules to translate minimum units of spelling (graphemes) into minimum units of phonology (phonemes). A combination of phonemes may guide assembly of pronunciation for completely novel letter strings, such as "glurp," or may achieve lexical access when the unfamiliar spelling translates into the phonology of a familiar word. Access to a lexical entry allows access, in turn, to lexical representations of words' pronunciations and conventional meanings, as one would find in a dictionary. (Different dual-route theories propose different representations and translation procedures.)

Word Naming

Word naming experiments measure the time required to pronounce individually presented words. At one time, dual-process theories provided a sufficient account of results from naming experiments, with skilled readers as participants. Low-frequency regular words, such as "mint," that obey sublexical grapheme-phoneme corre-

spondence rules are named faster than low-frequency exception words, such as "pint," that entail exceptions to the rules. No regularity effect was found to high-frequency words. Hence, naming of low-frequency words is accomplished by the sublexical module, but naming of all other words is accomplished by the lexical module. Eventually, however, this categorical regular/exception distinction was contradicted. New studies found graded effects of semiregular relationships between spelling and phonology, not simply the regular versus exception distinction predicted by traditional dual-route theories. For instance, although both "wave" and "wade" obey the grapheme-phoneme rules, the existence of "have," an exception "neighbor" to "wave," induces slower naming times to "wave" itself.

Lexical Decision

Word recognition is also studied by using the *lexical decision task*. Lexical decision experiments measure the time required to indicate that an individually presented word is a word (with catch trials that present pseudoword spellings). Previously, lexical decision time did not appear to be affected by regularity, only by relative frequency. High-frequency words are recognized faster than low-frequency words. Regularity effects arise in the sublexical module, and frequency effects arise in the lexical module. Hence, recognition for lexical decisions appeared to rely on the lexical module, exclusively. New studies contradicted this hypothesis, however. Key findings were subcategories of exception words, such as "weird" or "choir," called *strange words*, that produced slow and error-prone performance, reliably worse performance than that to ordinary exception words such as "pint." These findings, like the graded-regularity effects in naming experiments, contradict the categorical regularity distinction of dual-route theories.

Patient Studies

At one time, dual-process theories also provided a reasonable account of neuropsychological findings. Lexical and sublexical modules were corroborated in the patterns of naming errors, described in case studies of brain-damaged individuals. For example, surface dyslexics incorrectly regularize exception words ("pint" is pronounced to rhyme with "mint") but correctly name regular words. Regularized pronunciations of exception words dissociate the sublexical module (the source of regularization errors) from the damaged or absent lexical module (the source of correct pronunciations). Alternatively, deep dyslexics produce visual errors ("bush" is pronounced as "brush") and semantic errors ("bush" is pronounced as "tree"). These errors were attributed to a dissociated but damaged lexical module.

Evidence from patient studies proved to be problematic, however. No general agreement emerged concerning which patients' deficits actually counted as dissociated components of word recognition. All of the patient case studies that concerned reading were challenged by competing theorists, who claimed that they did not actually dissociate synthetic versus analytic components or that they simply did not pertain to reading (Van Orden, Pennington, and Stone, 2001). More recent brain-imaging studies appear to have arrived at a similar impasse. No general agreement has emerged in the brain-imaging literature that uniformly implicates specific brain regions in a large sample of reading tasks. Indeed, small differences in reading tasks and experimental methods appear to implicate different brain regions in what appear, intuitively, to be very similar reading acts.

Additive Factors Method

As we have noted, dual-process theories emerged in the 1970s, when cognitive psychology was predominantly concerned with in-

formation processing. Within that framework, the mind was conceived as an information-processing device that could be described in a way much like a flowchart of information processing in a computer program. Simon (1973) described how complex systems, such as cognitive systems, could be partly decomposed if the system's components interact approximately linearly. If interactions (exchanges) between components are linear in their effect, then the components can be identified even if their internal dynamics are nonlinear. Cognitive components thus described work as a chain of single causes—a metaphorical extension of domino causality. Push the first domino in a chain of standing dominoes, and each will fall in its turn.

The previous theoretical rationale coincided with a methodological tool to individuate cognitive components: the *additive factors method*, proposed by Sternberg (1969). Factorial experiments allow simultaneous manipulations of candidate variables that may influence distinct hypothetical components. The main effects of two or more manipulations are additive if they add up to the total behavioral effect. In this idealization, separate experimental manipulations selectively influence (e.g., slow) the falling times of separate “dominoes.” Alternatively, when nonadditive interaction effects are observed, manipulations do not satisfy the assumption of selective influence and influence (at least) one component in common. Thus, to Sternberg's lasting credit, his method included an empirical failure point: ubiquitous nonadditive interaction effects.

Additive main effects are rarely observed in reading experiments. It is not possible to manipulate all factors simultaneously in one experiment, but it is possible to trace chains of nonadditive interactions across published experiments that preclude the assignment of any factors to distinct components. For example, factorial manipulations of word frequency and regularity yield nonadditive interaction effects. This raises the question of whether the respective effects actually arise from separate synthetic and analytic processes.

Parallel Distributed Processing Models

Parallel distributed processing (PDP) models allowed a new position in the debate, because they did not require distinct synthetic and analytic processes (Seidenberg, 1995). PDP models are connectionist models in which constraints (connection weights) determine the activation values of nodes. Nodes represent words' spellings, pronunciations, or meanings, and patterns of response times are simulated in a model's *error term* (the difference between a model's pronunciation, for example, and an ideal correct pattern of pronunciation-node activation). A learning algorithm shapes a matrix of connection weights to reflect statistical relationships among node representations. This is referred to as statistical or *covariant learning*. PDP models introduced covariant learning algorithms to a broad audience of cognitive scientists. PDP models were equally important as existence proofs, which advanced the synthetic/analytic question. Covariant learning may reflect, in a single process, both subword (analytic) and whole-word (synthetic) covariation, as we illustrate next.

Distinctions in the relationships among English spellings and pronunciations, at a variety of scales, may all be construed as statistical relationships. Covariant learning tracks all scales of covariation, simultaneously, in the connection weights of a PDP model (Plaut et al., 1996). For example, consonant spellings and pronunciations are more reliably correlated than vowel spellings and pronunciations, and in both cases, there are statistically dominant and subordinate relationships. “Regular” words, comprising dominant relationships, are named more quickly than “exception” words that include subordinate relationships. Likewise, spelling bodies (e.g., the spelling pattern “-uck” in “duck”) and pronunciation rimes (e.g., pronunciation /uk/ in /duk/) are invariantly correlated; but some other body-rime relationships are dominant though less

strongly correlated (“-int” pronounced as in “mint”); and still other body-rimes are subordinate and only weakly correlated (“-int” pronounced as in “pint”). This rank order is corroborated in naming times; words like “duck” are named faster than words like “mint,” and words like “mint” are named faster than words like “pint” (all other things being equal). Finally, a word's relative frequency estimates the strength of the relationship between the word's (whole-word) spelling and pronunciation; high-frequency words are named faster than low-frequency words.

As the examples illustrate, the outcome of covariant learning will be determined by the pattern of statistical relationships in a *training set*—the sample of words used to train the model. Each training set entails a sample of constraints (relationships between spelling, phonology, and meaning) from the body of constraints in a literate culture, and covariant learning attunes the network to the sampled constraints. Thus, implicit in the training set is a description of the cultural artifact—a particular language's pattern of relationships—that is crucial for cognitive theory. Jared (1997) used this implication of PDP models to derive a nonintuitive empirical test. Careful attention to statistical relationships among the spellings and pronunciations of high-frequency words predicted a statistical advantage for high-frequency words with invariant body-rime relationships. Jared subsequently corroborated this prediction—a previously unobserved “regularity” effect in naming for high-frequency words. However, no such effect was observed in a lexical decision experiment.

Hybrid, partially recurrent, connectionist models moved the PDP approach further in the direction of fully recurrent, iterative, “neural” network models. In a hybrid model, the output of nonrecurrent (strictly feedforward) portions of a PDP network sets the initial conditions in a recurrent subnetwork that includes feedback connections. The recurrent portion behaves as an attractor network, tuned to fixed points that correspond to word pronunciations (for example), and pronunciation times are simulated in the number of iterations before the network reaches a “stable” attractor.

Partly “damaged” hybrid networks simulated the bizarre semantic and visual errors of deep dyslexic patients as well as the regularization errors of surface dyslexics. Simulated lesions have been implemented in several ways, including (a) random cutting of some connections between nodes, (b) random changes in connection weights, and (c) random selection of nodes whose values are fixed at zero. The various patterns of dyslexic patient's naming errors have been mimicked by using one or combinations of these simulated lesions.

Iterative Network Models

Hybrid feedforward PDP models were actually proposed as a first step toward fully recurrent, iterative networks. Iterative networks are attractor networks simulated as nonlinear iterative maps. A nonlinear iterative map may approximate solutions of a system of nonlinear differential equations. Thus, iterative network models, as dynamical systems, invoke the most sophisticated mathematical framework available to scientists (Farmer, 1990). An iterative map takes its output at one time step as input on the next time step until a stable pattern of node activity emerges—an attractor state. A stable attractor state corresponds to an iterative model's naming response.

Iterative network models may include covariant learning algorithms that reflect relationships that map from patterns of spelling to pronunciations and from pronunciations to spelling patterns (and meanings). Invariant, bidirectional relationships correspond to stable attractors in the network, which extends the previous view of statistical relationships among spellings and pronunciations. Some consonants have invariant bidirectional relationships with their pronunciations. For example, the grapheme B at the beginning of a word is always pronounced /b/, and the /b/ pronunciation is always

spelled B. Likewise, some spelling-body pronunciation-rime relationships are invariant (e.g., “-uck” and /uk/, as in “duck,” always co-occur). And most words have a bidirectional invariant relationship between their particular whole-word spelling and their particular pronunciation. As we noted, invariant bidirectional relationships correspond to stable attractor states in an iterative network. By contrast, ambiguous spelling-pronunciation relationships correspond to multiple, mutually inconsistent *multistable*, or more precisely *metastable*, attractor states in an iterative network.

Empirical studies of nonlinear systems typically focus on their less stable behavioral regimes, because very stable regimes reveal less of system dynamics. Pioneering studies have examined how ambiguous relationships between spelling and pronunciation affect empirical patterns in naming performance. By definition, the relationship between spelling and phonology is ambiguous if more than one reliable pronunciation is elicited by the same spelling. For example, a homograph, such as “wind,” has two legitimate pronunciations and is thus an ambiguous spelling. In the case of “wind,” its “regular” pronunciation (to rhyme with “pinned”) is produced faster than its “exception” pronunciation (to rhyme with “find”). Thus, the dynamics of word naming must unfold in a way that respects this ordering.

Kawamoto and Zemplidze (1992) simulated homograph naming using an iterative network. Relationships (connections) among letter, phoneme, and semantic node families were recurrent (including both feedforward and feedback connections) and excitatory, but within each node family, recurrent connections were (predominantly) inhibitory. The multistable unfolding of homograph pronunciations was simulated as a transcritical bifurcation. In a transcritical bifurcation, for example, the two possible pronunciations of “wind” exchange stability at a bifurcation point. That is, initial dynamics unfold in favor of one potential solution, but over successive iterations, additional constraints (which may unfold on a slower time scale) begin to favor an alternative solution. In the case of “wind,” the faster “regular” pronunciation reflected a strong local attractor between letter and phoneme nodes. However, coherent interactive activation among phoneme and semantic nodes slowly emerged to favor the “exception” pronunciation. The bifurcation point occurred when the balance of constraints switched to favor the “exception” pronunciation. At the bifurcation point, the “regular” pronunciation (attractor) exchanged stability with the “exception” pronunciation.

Discussion

The previous simulation of ambiguous homograph pronunciations as bifurcation phenomena illustrates how nonlinear dynamical systems theory has been applied to reading performance. However, empirical methods that are appropriate to nonlinear analysis are not widely applied. In large part, connectionism has inherited the empirical methods of information-processing psychology. However, statistical analyses that assume the general linear model and theories to be implemented as strongly nonlinear dynamical systems may be incompatible because they entail different notions of causality.

Previously, we discussed how the factorial logic of additive factors method assumes a one-way, domino-effect notion of causality. In contrast, bifurcation phenomena entail *circular causality*. Circular causality requires a strategic (not morphological) reduction of system behavior, due to emergent properties. In a strategic reduction, generic emergent properties may be found at multiple levels of systems, but emergent properties at “higher” levels do not reduce to causal properties of “lower” levels.

Plausible nonlinear models allow that it may not be productive, for scientific purposes, to view word recognition (or reading) as a component process, but linear methods are directed at the discovery of component processes. Moreover, the results of a vast linear anal-

ysis actually raise the question of whether a distinct process of word recognition may be distinguished. All reading tasks would seem to include word recognition, but they do not yield any converging pattern of word recognition effects.

Consider the word frequency effect in the lexical decision task, for example. The same set of words that produce a large word frequency effect in the lexical decision task may produce a reduced, or statistically unreliable, word frequency effect in naming (or other tasks). Within the lexical decision task itself, it is possible to modulate the word frequency effect by making the non-words more or less word-like (and, in turn, modulate nonadditive interaction effects among word frequency, regularity, and other variables). Across languages, Hebrew produces a larger word frequency (familiarity) effect than English, and English produces a larger effect than Serbo-Croatian (which tracks the analytic transparency of their print-to-sound relationships—less transparent equals larger frequency effect).

All empirical phenomena of word recognition appear to be conditioned by task, task demands, and reference language (Frost and Katz, 1992; Lukatela and Turvey, 1998). These nonadditive interactions allow the question of whether “word recognition effects” may be attributed to a distinct process of word recognition. Consider the previous examples together, within the guidelines of additive factors logic. Word recognition factors cannot be individuated from each other, and they cannot be individuated from the context of their occurrence (task, task demands, and language). Because additivity of task effects or language effects is never observed, we lack evidence that may individuate word recognition.

Despite these problems, most theorists, including connectionist theorists, share the intuition that a distinct separable component of word recognition may yet exist. We speculate, however, that the intuition persists because most cognitive theorists were trained exclusively in linear methods. If we are correct, then rigorous tests of iterative network models await a reliable logic of nonlinear analysis that is consistent with nonlinear dynamical systems theory and appropriate to reading performance. Thus, the historical question of analytic versus synthetic processes with which we began is replaced by the question of linear versus nonlinear dynamics—a question motivated in part by the success of nonlinear iterative network models.

Road Map: Linguistics and Speech Processing

Related Reading: Constituency and Recursion in Language; Developmental Disorders; Motor Theories of Perception

References

- Farmer, J. D., 1990, A Rosetta Stone for connectionism, *Phys. D*, 42:153–187.
- Frost, R., and Katz, L. (Eds.), 1992, *Orthography, Phonology, Morphology, and Meaning*, Amsterdam: North Holland. ♦
- Jared, D., 1997, Spelling-sound consistency affects the naming of high-frequency words, *J. Mem. Lang.*, 36:505–529.
- Kawamoto, A. H., and Zemplidze, J. H., 1992, Pronunciation of homographs, *J. Mem. Lang.*, 31:394–374.
- Lukatela, G., and Turvey, M. T., 1998, Reading in two alphabets, *Am. Psychol.*, 53:1057–1072. ♦
- Mattingly, I. G., 1992, Linguistic awareness and orthographic form, in *Orthography, Phonology, Morphology, and Meaning* (R. Frost and L. Katz, Eds.), Amsterdam: North-Holland, pp. 11–26.
- Neisser, U., 1967, *Cognitive Psychology*, Englewood Cliffs, NJ: Prentice Hall. ♦
- Perfetti, C. A., 1985, *Reading Ability*, New York: Oxford University Press. ♦
- Pennington, B. F., 1991, *Diagnosing Learning Disorders: A Neuropsychological Framework*, New York: Guilford Press. ♦
- Plaut, D. C., McClelland, J. L., Seidenberg, M. S., and Patterson, K., 1996, Understanding normal and impaired word reading: Computational principles in quasi-regular domains, *Psychol. Rev.*, 103:56–115.

Rayner, K., and Pollatsek, A., 1989, *The Psychology of Reading*, Englewood Cliffs, NJ: Prentice Hall. ♦
 Seidenberg, M. S., 1995, Visual word recognition: An overview, in *Speech, Language, and Communication* (J. L. Miller and P. D. Eimas, Eds.), New York: Academic Press, pp. 137–179.
 Simon, H. A., 1973, The organization of complex systems, in *The Chal-*

lenge of Complex Systems (H. H. Pattee, Ed.), New York: George Braziller, pp. 1–27.
 Sternberg, S., 1969, The discovery of processing stages: Extensions of Donders' method, *Acta Psychol.*, 30:276–315.
 Van Orden, G. C., Pennington, B. F., and Stone, G. O., 2001. What do double dissociations prove? *Cogn. Sci.*, 25:111–172.

Recurrent Networks: Learning Algorithms

Kenji Doya

Introduction

The backpropagation algorithm for feedforward networks (Figure 1A) has been successfully applied to a wide range of problems, from neuroscience to consumer electronics (see BACKPROPAGATION: GENERAL PRINCIPLES). However, what can be implemented by a feedforward network is just a static mapping of the input vectors. The human brain, however, is not a stateless input-output system but a high-dimensional nonlinear dynamical system. In order to model dynamical functions of the brain, or to design a machine that performs as well as a brain does, it is essential to utilize a system that is capable of storing internal states and implementing complex dynamics.

This is why learning algorithms for *recurrent neural networks* (Figure 1B), which have feedback connections and time delays, have been studied with enthusiasm. In a recurrent network, the state of the system can be encoded in the activity pattern of the units

and a wide variety of dynamical behaviors can be programmed by the connection weights.

A popular subclass of recurrent networks is those with symmetric connection weights. In this case, the network dynamics is guaranteed to converge to a minimum of “energy” function (see ENERGY FUNCTIONALS FOR NEURAL NETWORKS and COMPUTING WITH ATTRACTORS). Typical examples are associative memory networks (see ASSOCIATIVE NETWORKS), optimization networks (see OPTIMIZATION, NEURAL), and WINNER-TAKE-ALL NETWORKS (q.v.).

However, steady-state solutions are only a limited portion of the capabilities of recurrent networks. A recurrent network can serve as a sequence recognition system (see LANGUAGE PROCESSING) or as a sequential pattern generator (see MOTOR PATTERN GENERATION and SEQUENCE LEARNING). More generally, it is capable of transforming an input sequence into some other output sequence (see TEMPORAL PATTERN PROCESSING). It can be used as a nonlinear filter (see KALMAN FILTERING: NEURAL IMPLICATIONS), a nonlinear controller (see IDENTIFICATION AND CONTROL), or a finite-state machine (see LANGUAGE PROCESSING).

This article reviews the learning algorithms for training recurrent networks. There are three major frameworks of learning: *supervised learning*, based on the output error signal, *reinforcement learning*, based on the scalar reward signal, and *unsupervised learning*, based on the statistical feature of the input signal. Our main focus will be on supervised learning algorithms for recurrent networks. We also provide a brief overview of reinforcement and unsupervised learning algorithms.

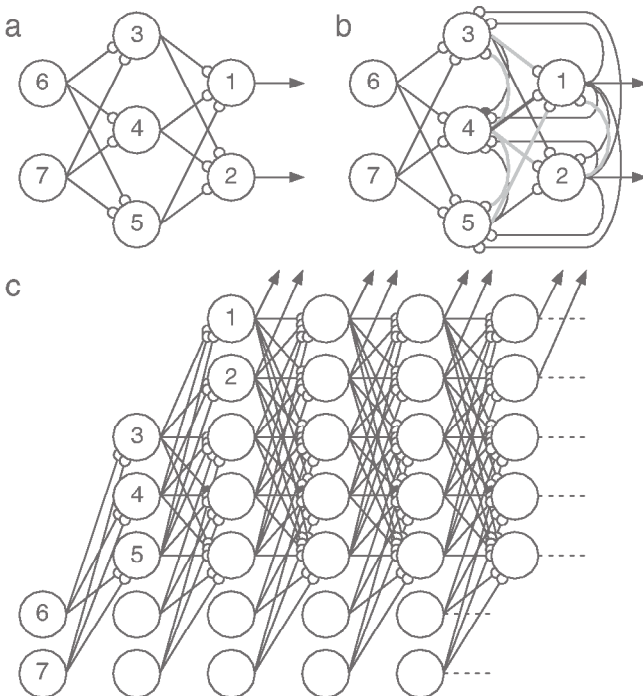


Figure 1. Examples of a feedforward network (A) and a recurrent network (B), where units 1 and 2 are output units; units 3, 4, and 5 are hidden units; and units 6 and 7 are input units. The multistep operation of a recurrent network (B) can be unrolled as a multilayer feedforward network (C).

Supervised Learning Algorithms

The problem setup of supervised learning in recurrent networks is similar to the case of feedforward networks: a network is given a desired output for an input. An error function is defined and its gradient with respect to the weights is derived. However, the major difference is that the input and output are not static vectors but *time sequences*.

For example, in the recurrent network shown in Figure 1B, units 1 and 2 are output units; units 3, 4, and 5 are hidden units; and units 6 and 7 are input units. A small change of a connection weight, say w_{43} (shown as black dot), affects the output units, say unit 1, not only through the direct connection from unit 4 to 1 (thick black line), but also through indirect connections, through units 2, 3, and 5 (thick gray lines), and through infinitely many multistep paths with multiple delays. It makes exact calculation of output error gradient rather complex.

One simple strategy is to neglect all the indirect paths. In this case, although the network state evolves according to the recurrent network, as in Figure 1B, a simple backpropagation algorithm is

applied by regarding it as a feedforward network, as in Figure 1A. Such coarse approximation methods turned out to be effective in work on language acquisition using *simple recurrent networks* that have recurrent connections between hidden units (see LANGUAGE PROCESSING).

There are two basic ways to calculate the exact gradient of the output with respect to the weights: forward methods and backward methods. Forward methods estimate the effects of a small change in a weight on the network state trajectory in the form of a linear dynamical equation system. This can be calculated concurrently with the network dynamic, and thus is useful for on-line learning. A drawback is the amount of computation needed to update a set of dynamical equations for each weight.

Backward methods estimate the causes of the output error backward in time. In discrete-time case, this method is realized by “unrolling” the multistep evolution of the network state as a multilayer feedforward network, as in Figure 1C, and applying the standard backpropagation algorithm (Rumelhart, Hinton, and Williams, 1986). In continuous-time cases it is done by running a set of “adjoint systems” backward in time. Although the method requires asynchronous operation, with the evolution and storage of the state trajectory done first and the error gradient calculation done afterwards, the amount of computation is much less than in the forward methods (see Pearlmutter, 1995, for a comprehensive review). In the following sections, we formulate these algorithms for both discrete-time and continuous-time models, and then discuss technical problems in using them.

Discrete-Time Model

We will start with a discrete-time recurrent network with n units and m inputs. We denote the state of the i th unit by y_i and the connection weights from the j th to the i th unit by w_{ij} . Both external inputs u_j and recurrent inputs y_j are represented as z_j for convenience:

$$y_i(t+1) = f\left(\sum_{j=1}^{n+m} w_{ij} z_j(t)\right) \quad (i = 1, \dots, n)$$

$$z_j(t) = \begin{cases} y_j(t), & j \leq n \\ u_{j-n}, & j > n \end{cases} \quad (1)$$

The output nonlinearity $f(\cdot)$ is usually a squashing function, such as $f(x) = 1/(1 + e^{-x})$ and $f(x) = \tanh x$, whose derivatives are conveniently given by $f'(x) = f(x)(1 - f(x))$ and $f'(x) = 1 - f(x)^2$, respectively. We can introduce a bias parameter by assuming that one of the inputs u_j is constant.

The goal of learning is to set the parameters w_{ij} so that the output trajectory $(y_1(t), \dots, y_n(t))$ follows a desired trajectory $(d_1(t), \dots, d_n(t))$ ($t = 1, \dots, T$) with a given initial state $(y_1(0), \dots, y_n(0))$ and an input sequence $(u_1(t), \dots, u_m(t))$ ($t = 0, \dots, T-1$). We define the error function

$$E = \sum_{t=1}^T \sum_{i=1}^n \mu_i(t) \frac{1}{2} (y_i(t) - d_i(t))^2 \quad (2)$$

and perform gradient descent on E with respect to the weights w_{ij} . The masking function $\mu_i(t)$ specifies which components of the trajectory are to be supervised at what time. In a typical case, $\mu_i(t) \equiv 1$ for output units and $\mu_i(t) \equiv 0$ for hidden units. When only the end point of the trajectory is specified, $\mu_i(T) = 1$ and $\mu_i(t) = 0$ for $t < T$.

Real-Time Recurrent Learning

The effect of weight change on the network dynamics can be seen by simply differentiating the network dynamics equation (Equation 1) by a weight w_{kl} (Williams and Zipser, 1989).

$$\frac{\partial y_i(t+1)}{\partial w_{kl}} = f'(x_i(t)) \left[\sum_{j=1}^{n+m} w_{ij} \frac{\partial y_j(t)}{\partial w_{kl}} + \delta_{ik} z_k(t) \right] \quad (i = 1, \dots, n) \quad (3)$$

where $x_i(t) = \sum_{j=1}^{n+m} w_{ij} z_j(t)$ is the net input to the unit and δ_{ik} is Kronecker's delta ($\delta_{ik} = 1$ if $i = k$ and otherwise 0). The term $\delta_{ik} z_k(t)$ represents an *explicit* effect of the weight w_{kl} on the unit k , and the term $\sum_{j=1}^{n+m} w_{ij} (\partial y_j(t) / \partial w_{kl})$ represents an *implicit* effect on all the units because of network dynamics.

Equation 3 for each unit $i = 1, \dots, n$ constitutes an n -dimensional linear dynamical system (with time-varying coefficients), where $((\partial y_1 / \partial w_{kl}), \dots, (\partial y_n / \partial w_{kl}))$ is taken as a dynamical variable. Since the initial state $y_i(0)$ of the network is independent of the connection weights, the appropriate initial condition for Equation 3 is

$$\frac{\partial y_i(0)}{\partial w_{kl}} = 0 \quad (i = 1, \dots, n)$$

Thus, we can compute $\partial y_i(t) / \partial w_{kl}$ *forward in time* by iterating Equation 3 simultaneously with the network dynamics (Equation 1). From this solution, we can calculate the error gradient as follows:

$$\frac{\partial E}{\partial w_{kl}} = \sum_{t=1}^T \sum_{i=1}^n \mu_i(t) (y_i(t) - d_i(t)) \frac{\partial y_i(t)}{\partial w_{kl}} \quad (4)$$

A standard *batch* gradient descent algorithm is to accumulate the error gradient by Equation 4 and update each weight w_{kl} by

$$w_{kl} := w_{kl} - \varepsilon \frac{\partial E}{\partial w_{kl}} \quad (5)$$

where $\varepsilon > 0$ is a learning rate parameter.

An alternative update scheme is the gradient descent of *current* output error $\sum_{i=1}^n (\frac{1}{2}) \mu_i(t) (y_i(t) - d_i(t))^2$ at each time step, namely,

$$w_{kl}(t+1) = w_{kl}(t) - \varepsilon \sum_{i=1}^n \mu_i(t) (y_i(t) - d_i(t)) \frac{\partial y_i(t)}{\partial w_{kl}} \quad (6)$$

Note that we assumed that w_{kl} is a constant, not a dynamical variable, in deriving Equation 3, so we have to keep the learning rate ε small enough. However, this on-line update scheme was shown to be effective on a number of temporal learning tasks (Williams and Zipser, 1989), and it is often called *real-time recurrent learning*.

A drawback to this error gradient calculation forward in time is that we have to solve an n -dimensional system (Equation 3) for each of the weights w_{kl} ($k = 1, \dots, n; l = 1, \dots, n+m$). It requires $O(n^3)$ memories and $O(n^4)$ computations.

Backpropagation Through Time

Another learning algorithm for a discrete-time model can be derived by “unfolding” a recurrent network into a multilayer network (Figure 1C) (Rumelhart et al., 1986). In this scheme, T -step iteration of a recurrent network is regarded as one sweep of operation in a T -layered feedforward network with identical connection weights w_{ij} between successive layers. The error gradient can be derived in the same way as in the standard backpropagation, except that the output errors are not only given in the last layer but added in each layer:

$$\frac{\partial E}{\partial y_i(t)} = \sum_{j=1}^n \frac{\partial E}{\partial y_j(t+1)} f'(x_j(t)) w_{ji} + \mu_i(t) (y_i(t) - d_i(t)) \quad (i = 1, \dots, n) \quad (7)$$

Since the error E is independent of the state at $t > T$, the boundary condition for Equation 7 is given at the final time step as

$$\frac{\partial E}{\partial y_i(T+1)} = 0 \quad (i = 1, \dots, n)$$

Thus, the learning equation (Equation 7) can be iterated *backward in time* from $t = T$ to 1.

From the solution $\partial E / \partial y_i$, the error gradients are given by

$$\frac{\partial E}{\partial w_{ij}} = \sum_{t=1}^T \frac{\partial E}{\partial y_i(t)} f'(x_i(t-1)) z_j(t-1) \quad (8)$$

and the weights are updated in a batch using Equation 5.

The advantage of this algorithm is that we have to solve only one n -dimensional system, Equation 7, for adjusting all the weights. Therefore, only $O(n^2)$ computations are required. However, since the learning equation, Equation 7, has to be solved *backward* in time, we cannot update the weights on-line, and we have to store the history of the network state $y_i(t)$ ($i = 1, \dots, n$; $t = 1, \dots, T$), which requires $O(nT)$ memories.

Continuous-Time Model

A continuous-time model is a natural choice for modeling systems that are governed by differential equations. Time constants of continuous-time models are convenient parameters for setting local memory spans for individual units. They can also be adjusted by learning, as discussed later.

Slightly different versions of continuous-time models have been studied. Here, we focus on the following model (Pearlmutter, 1989),

$$\begin{aligned} \tau_i \dot{y}_i(t) &= -y_i(t) + f\left(\sum_{j=1}^{n+m} w_{ij} z_j(t)\right) \quad (i = 1, \dots, n) \\ z_j(t) &= \begin{cases} y_j(t), & j \leq n \\ u_{j-n}, & j > n \end{cases} \end{aligned} \quad (9)$$

However, similar derivations apply to other models as well (Doya and Yoshizawa, 1989).

We define an error integral

$$E = \int_0^T \sum_{i=1}^n \mu_i(t) \frac{1}{2} (y_i(t) - d_i(t))^2 dt \quad (10)$$

and derive a gradient descent algorithm for minimizing E for a desired trajectory $(d_1(t), \dots, d_n(t))$ ($0 \leq t \leq T$) with a given initial state $(y_1(0), \dots, y_n(0))$ and an input sequence $(u_1(t), \dots, u_m(t))$.

Variation Method

The effect of a change in a weight w_{kl} on the state $y_i(t)$ can be estimated by differentiating Equation 9, the network dynamics equation, as follows:

$$\tau_i \frac{d}{dt} \left(\frac{\partial y_i}{\partial w_{kl}} \right) = -\frac{\partial y_i}{\partial w_{kl}} + f'(x_i(t)) \left[\sum_{j=1}^n w_{ij} \frac{\partial y_j}{\partial w_{kl}} + \delta_{ik} z_l(t) \right] \quad (i = 1, \dots, n) \quad (11)$$

This forms an n -dimensional linear differential equation system with the state variable $((\partial y_1 / \partial w_{kl}), \dots, (\partial y_n / \partial w_{kl}))$ and is called a *variation system* of the network dynamics equation, Equation 9. The initial condition for this system is given by

$$\frac{\partial y_i(0)}{\partial w_{kl}} = 0 \quad (i = 1, \dots, n)$$

because the initial state of the network is independent of the weights. We can numerically integrate Equation 11 forward in time concurrently with the network dynamics in Equation 9.

From the solution $(\partial y_i(t) / \partial w_{kl})$ ($0 \leq t \leq T$), the error gradient is given by

$$\frac{\partial E}{\partial w_{ij}} = \int_0^T \sum_{i=1}^n \mu_i(t) (y_i(t) - d_i(t)) \frac{\partial y_i(t)}{\partial w_{kl}} dt \quad (12)$$

We can use either the batch update scheme (Equation 5) at the end of a sequence, or the on-line update scheme

$$\dot{w}_{kl} = -\varepsilon \sum_{i=1}^n \mu_i(t) (y_i(t) - d_i(t)) \frac{\partial y_i(t)}{\partial w_{kl}} \quad (13)$$

with sufficiently small learning rate $\varepsilon > 0$.

The error gradient for a time constant τ_k is given by the following variation equation:

$$\tau_k \frac{d}{dt} \left(\frac{\partial y_i}{\partial \tau_k} \right) = -\frac{\partial y_i}{\partial \tau_k} + f'(x_i(t)) \left[\sum_{j=1}^n w_{ij} \frac{\partial y_j}{\partial \tau_k} - \delta_{ik} \dot{y}_k(t) \right] \quad (i = 1, \dots, n) \quad (14)$$

Adjoint Method

The backward algorithm for a continuous-time model can be derived in several ways, for example, by finite difference approximation (Pearlmutter, 1989). Here we derive the algorithm as an "adjoint" system of the forward learning equation, Equation 11.

A pair of n -dimensional linear systems

$$\dot{\mathbf{p}} = \mathbf{A}(t)\mathbf{p} + \mathbf{b}(t) \quad \text{and} \quad \dot{\mathbf{q}} = -\mathbf{A}^*(t)\mathbf{q} - \mathbf{c}(t)$$

are called *adjoint* to each other, where \mathbf{A}^* denotes the transpose of matrix \mathbf{A} . A useful property of adjoint systems is that their solutions satisfy the following *Green's equality*:

$$\int_0^T \mathbf{q}(t) \cdot \mathbf{b}(t) dt - \int_0^T \mathbf{c}(t) \cdot \mathbf{p}(t) dt = \mathbf{q}(T) \cdot \mathbf{p}(T) - \mathbf{q}(0) \cdot \mathbf{p}(0)$$

We can actually compose an adjoint system of the variation equation, Equation 11,

$$\dot{q}_i = \frac{q_i(t)}{\tau_i} - \sum_{j=1}^n \frac{f'(x_j(t))}{\tau_j} w_{ji} q_j(t) - \mu_i(t) (y_i(t) - d_i(t)) \quad (15)$$

where we put $p_i = (\partial y_i / \partial w_{kl})$, $A_{ij}(t) = (f'(x_i(t)) / \tau_i) w_{ij} - (\delta_{ij} / \tau_i)$, $b_i(t) = (f'(x_i(t)) / \tau_i) \delta_{ik} y_k(t)$, and $c_i(t) = \mu_i(t) (y_i(t) - d_i(t))$. With the boundary conditions $p_i(0) = (\partial y_i(0) / \partial w_{kl}) = 0$ and $q_i(T) = 0$, Green's equality becomes

$$\int_0^T \sum_{i=1}^n q_i(t) \frac{f'(x_i(t))}{\tau_i} \delta_{ik} y_k(t) dt = \int_0^T \sum_{i=1}^n \mu_i(t) (y_i(t) - d_i(t)) \frac{\partial y_i}{\partial w_{kl}} dt \quad (16)$$

Note that the right-hand side is identical to the error gradient (Equation 12). Thus, we have an alternative form of the error gradient

$$\frac{\partial E}{\partial w_{kl}} = \int_0^T q_k(t) \frac{f'(x_k(t))}{\tau_k} z_l(t) dt \quad (17)$$

Similarly, the error gradient for a time constant is given by

$$\frac{\partial E}{\partial \tau_k} = \int_0^T q_k(t) \frac{f'(x_k(t))}{\tau_k} (-\dot{y}_k(t)) dt \quad (18)$$

As in the discrete-time case, we first run the network dynamics (Equation 9) forward in time and then run the adjoint system (Equation 15) backward in time with the terminal condition $q_i(T) = 0$. The weights are updated in batch by Equation 5.

Technical Remarks

Forward or Backward?

The forward algorithms require $O(n^4)$ computations. Therefore, it is not suitable for a fully connected network with tens or hundreds

of units. However, for a small-sized network or a network with only local connections, on-line weight update can be an advantage.

In order to allow on-line weight update with the efficiency of the backward algorithm, a truncated version of the backpropagation-through-time algorithm has been proposed (Schmidhuber, 1992)

Teacher Forcing

The so-called teacher forcing technique has been shown to be helpful, especially in training a network into an autonomous dynamical system (Doya and Yoshizawa, 1989; Williams and Zipser, 1989). In this scheme, the desired output $d_i(t)$ is used to drive the network dynamics in place of the feedback of its actual output $y_i(t)$.

The reasons for needing teacher forcing are:

- The state of the network is assigned to the desired one of the many attractor domains.
- In learning oscillatory patterns, unless the phase of the network output is synchronized to the teacher signal, there will be an apparently large error (Doya and Yoshizawa, 1989).
- It will avoid a local minimum solution of static output at the mean value of the dynamic teacher signal (Williams and Zipser, 1989).
- The linearized equation for a limit cycle trajectory is not asymptotically stable if the system is running autonomously (Doya, 1992).

One problem with this technique is that the trajectory learned with teacher forcing may not be stable when the network is run autonomously after learning. Several heuristics have been proposed for enhancing the stability of the nonforced trajectory:

Noisy forcing: Add some noise to the forcing input.

Partial forcing: Use a mixed input $z_i(t) = y_i(t) + \alpha(d_i(t) - y_i(t))$ with $0 < \alpha < 1$ and decrease the forcing rate α with the progress of learning.

Part-time forcing: Turn on forcing to synchronize the network to the teacher, and then turn off forcing to train the autonomous trajectory.

Bifurcation Boundaries

In many learning tasks, the goal is not only to replicate particular sample trajectories but to reconstruct some *attractors* in the state space, such as fixed points, limit cycles, and chaotic attractors.

For example, when a network is trained as a finite-state machine, it must have distinct attractors in order to represent discrete states. As another example, when a network is trained as a periodic oscillator, it must have a limit cycle attractor. When we gradually change network parameters, we expect that the shape and location of the attractors will change continuously. However, that is not always true. At some points in the parameter space, attractors can emerge, disappear, or change their stability. Such a phenomenon is known as *bifurcation* in nonlinear systems theory (see DYNAMICS AND BIFURCATION IN NEURAL NETS and CANONICAL NEURAL MODELS).

With some kinds of bifurcation, such as saddle-node bifurcation, the state of the network changes drastically. Even if the equilibrium or the trajectory persists, the linearized equations that are used for gradient computation can lose asymptotic stability. Accordingly, when the network goes through a bifurcation point, the solution of the learning equation can grow rapidly, and the gradient descent algorithm can be unstable (Doya, 1992).

Although this might sound like a rare, pathetic situation, bifurcation is actually an inevitable step in many learning tasks (Doya,

1992). If the connection weights w_{ij} are initialized with small random values, the network dynamics has a single global attractor point. In order to have multiple attractor domains or a limit cycle, the network must go through some bifurcation boundary. Conversely, until the network goes through an appropriate bifurcation, even a simple memory task can be very difficult, owing to exponential decay of the error gradient.

Incremental Training

It has been reported that gradual increase in the complexity of training examples is critical for successfully training a network as a finite-state machine (see LANGUAGE ACQUISITION). A possible reason for this is that a network can acquire memory mechanisms only gradually, by going through bifurcation boundaries. If we impose examples that require many internal states with a long time delay from the beginning, we might simply mess up the network. This problem of the developmental capability of recurrent networks needs further examination.

Discussion

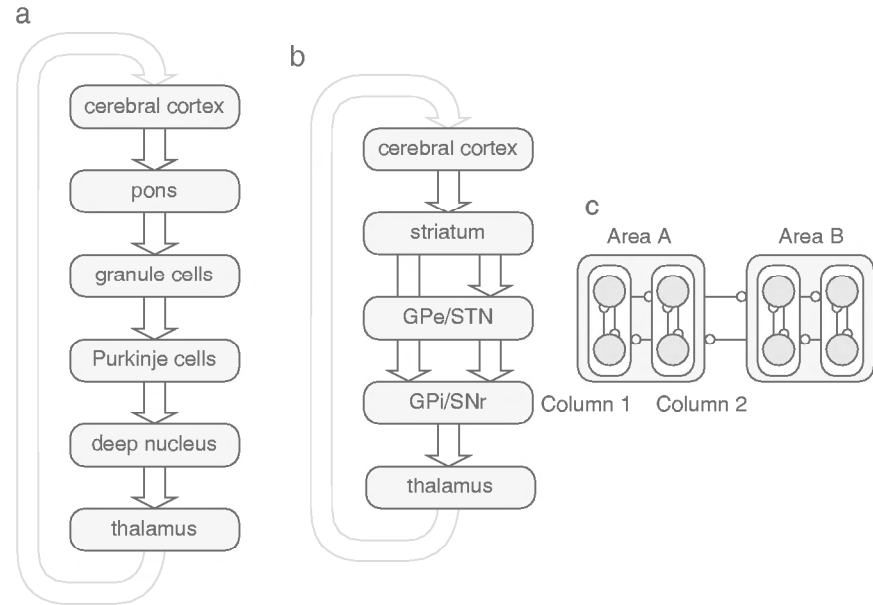
A fully connected recurrent neural network is potentially a very powerful system for temporal information processing. Based on the universal approximation theorem for three-layered networks (see UNIVERSAL APPROXIMATORS), it has been shown that a recurrent network can, with enough units, approximate any dynamical system (Funahashi and Nakamura, 1993). It has also been shown that a recurrent neural network, with its analog-valued computation, can have super-Turing computational power (see NEURAL AUTOMATA AND ANALOG COMPUTATIONAL COMPLEXITY). However, these theories do not guarantee that such a network can be readily realized by error gradient descent learning.

As already mentioned, the error gradient can decay or expand exponentially in time, which makes gradient descent more difficult than in the case of feedforward networks. Convergence of learning depends critically on the choice of network topology, initial weights, and the choice of training samples. These are some of the reasons why networks with specialized architectures have been crafted for specific problems, for example, networks with tapped delay-lines or local recurrent loops.

In a recent study of grammar learning (see LANGUAGE PROCESSING), recurrent neural networks were successfully trained to predict strings from context-free and context-sensitive languages (Rodriguez, 2001). In these examples, fractal structures in the network state space were utilized to approximate multiple "counters," which are necessary for processing complex grammatical structures such as palindromes. Interesting findings in such studies were that recurrent networks can generalize in terms of the depth of embedding.

Bayesian approaches have recently been applied to the learning of dynamics in recurrent networks (see BAYESIAN METHODS AND NEURAL NETWORKS; BAYESIAN NETWORKS; GRAPHICAL MODELS). A recurrent network can be trained by the method of extended Kalman filtering, which has properties similar to the RTRL algorithm with teacher forcing (Williams, 1992). EM methods for estimating the states of the hidden units and the weight parameters have been formulated (Ghahramani and Hinton, 2000). This seems to be a theoretically more sound way of nonlinear dynamical system estimation. However, since EM is essentially a local optimization process, whether this new wave of modeling methods can escape from the issue of bifurcation remains to be seen. Many recent approaches to temporal sequence processing are reviewed in Sun (2001) and other articles in the same book.

Figure 2. Network architectures of the cerebellum (A), the basal ganglia (B), and the cerebral cortex (C).



Biologically Inspired Learning Methods

It has been suggested that the network architectures of the cerebellum, the basal ganglia, and the cerebral cortex are specialized for different frameworks of learning, namely, the cerebellum for supervised learning, the basal ganglia for reinforcement learning, and cerebral cortex for unsupervised learning (Doya, 1999). The circuits of the cerebellum (Figure 2A) and the basal ganglia (Figure 2B) have roughly feedforward structures. While learning in the cerebellum is characterized by specific error signals carried by the climbing fibers to the Purkinje cells, learning in the basal ganglia is characterized by the reward signal broadcasted by the dopaminergic input to the striatum (see CEREbellum AND MOTOR CONTROL AND BASAL GANGLIA). They both form long recurrent loops starting from and ending in the cerebral cortex. The circuit of the cerebral cortex is characterized by massive recurrent connections, within and between functional columns, and between cortical areas (Figure 2C). Learning in the cerebral cortex is characterized by Hebbian learning (see CORTICAL HEBBIAN MODULES). Since the cerebral cortex embodies the most successful application of recurrent networks, both within the cortex and in the corticocerebellar and cortico-basal ganglia recurrent loops, it is natural to try to draw insights from the cortical network architecture.

The combination of recurrent excitation and lateral inhibition can implement a winner-take-all mechanism (see WINNER-TAKE-ALL NETWORKS). In combination with Hebbian plasticity and certain regulatory mechanisms, self-organization of feature detectors can be achieved (see SELF-ORGANIZATION AND THE BRAIN; COMPETITIVE LEARNING; HEBBIAN LEARNING AND NEURONAL REGULATION). This basic framework is shared by recent studies of receptive field formation and INDEPENDENT COMPONENT ANALYSIS (q.v.), which combine bottom-up Hebbian plasticity with lateral or top-down anti-Hebbian plasticity (see PATTERN FORMATION, NEURAL; INDEPENDENT COMPONENT ANALYSIS; UNSUPERVISED LEARNING WITH GLOBAL OBJECTIVE FUNCTIONS).

The Boltzmann Machine (see SIMULATED ANNEALING AND BOLTZMANN MACHINES) with its wake and sleep modes, is another basic model of cortical processing. Its extension to layered recurrent networks, the Helmholtz machine, is capable of extracting the hidden structure of sensory data and reproducing the data by top-down processing (see HELMHOLTZ MACHINES AND SLEEP-WAKE LEARNING).

One of the main open issues in REINFORCEMENT LEARNING (q.v.) is how to learn a good behavior when the environmental states are not perfectly observable (see IDENTIFICATION AND CONTROL). In such a case, the agent should store a certain “belief state” and update it according to the model of the environment. Actions are chosen according to the predicted future reward based on the belief state. Such complex operations could be implemented in the corticocerebellar and corticobasal ganglia loops, with the cerebral cortex representing the belief state, the cerebellum implementing the internal model of the environment, and the basal ganglia predicting the future reward. Better understanding of the corticocerebellar-basal ganglia system may give some clue for designing an adaptive agent under uncertainty.

Road Map: Learning in Artificial Networks

Related Reading: Computing with Attractors; Dynamics and Bifurcation in Neural Nets; Dynamics of Association and Recall; Helmholtz Machines and Sleep-Wake Learning; Recurrent Networks: Neurophysiological Modeling; Simulated Annealing and Boltzmann Machines

References

- Doya, K., 1992, Bifurcations in the learning of recurrent neural networks, in *Proceedings of the 1992 IEEE International Symposium on Circuits and Systems*, vol. 6, New York: IEEE, pp. 2777–2780.
- Doya, K., 1999, What are the computations in the cerebellum, the basal ganglia, and the cerebral cortex? *Neural Netw.*, 12:961–974.
- Doya, K., and Yoshizawa, S., 1989, Adaptive neural oscillator using continuous-time backpropagation learning, *Neural Netw.*, 2:375–386.
- Elman, J. L., 1990, Finding structure in time, *Cognit. Sci.*, 14:179–211.
- Funahashi, K., and Nakamura, Y., 1993, Approximation of dynamical systems by continuous time recurrent neural networks, *Neural Netw.*, 6:801–806.
- Ghahramani, Z., and Hinton, G. E., 2000, Variational learning for switching state-space models, *Neural Computat.*, 12:831–864.
- Pearlmutter, B. A., 1989, Learning state space trajectories in recurrent neural networks, *Neural Computat.*, 1:263–269.
- Pearlmutter, B. A., 1995, Gradient calculations for dynamic recurrent neural networks: A survey, *IEEE Trans. Neural Netw.*, 6:1212–1228. ♦
- Rodriguez, P., 2001, Simple recurrent networks learn context-free and context-sensitive languages by counting, *Neural Computat.*, 13:2093–2118.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, Learning representations by back-propagating errors, *Nature*, 323:533–536.

- Schmidhuber, J., 1992, A fixed size storage $O(n^3)$ time complexity learning algorithm for fully recurrent continually running networks, *Neural Comput.*, 4:243–248.
- Sun, R., 2001, Introduction to sequence learning, in *Sequence Learning: Paradigms, Algorithms, and Applications* (R. Sun and C. L. Giles, Eds.), New York: Springer-Verlag, pp. 1–10. ♦

- Williams, R. J., 1992, Training recurrent networks using the extended Kalman filter, in *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, Piscataway, NJ: IEEE, pp. 241–250.
- Williams, R. J., and Zipser, D., 1989, A learning algorithm for continually running fully recurrent neural networks, *Neural Comput.*, 1:270–280.

Recurrent Networks: Neurophysiological Modeling

Eberhard E. Fetz and Larry E. Shupe

Introduction

Dynamic recurrent network models can provide invaluable tools to help systems neurophysiologists understand the neural mechanisms mediating behavior. They can help overcome the limitations of biological experiments, which typically provide limited samples of the system, such as anatomical structures and their connections, the effects of lesions on behavior, or the activity of single neurons in behaving animals. The missing element required to synthesize these pieces can be provided by neural network models of the complete system. New algorithms make it possible to derive networks that simulate dynamic sensorimotor behavior and incorporate anatomically appropriate recurrent connectivity. The resulting networks determine the remaining free parameters based on examples of the behavior itself.

Training procedures initially developed for feedforward networks have been extended to dynamic recurrent networks, which differ from other modeling approaches in three key properties. First, the units are *dynamic*, meaning they can exhibit time-varying activity that can represent the mean firing rates of single or multiple neurons, membrane potentials, or some relevant time-varying stimulus or motor parameter. Second, the networks can have *recurrent* connectivity, including feedback and cross-connections. Third, the network connections required to simulate a particular dynamic behavior can be derived from examples of the behavior by *gradient descent* methods, such as backpropagated error correction. The resulting models provide complete neural network solutions of the behavior, insofar as they determine all the connections and activations of the units that simulate the behavior.

Neural networks that emulate particular dynamic behaviors basically transform spatiotemporal inputs into appropriate spatiotemporal outputs. These networks are usually comprised of interconnected “sigmoidal” units (units whose outputs are sigmoidal functions of their inputs); this mimics a biological neuron’s property of saturating at maximal rates for large inputs and decreasing to zero for low inputs.

To illustrate the training procedure, Figure 1 shows a representative network of such units, with input and output patterns that simulate a target-tracking task. Four input units carry signals representing the step changes in target locations; eight output patterns represent the firing rates of motor units in monkeys tracking such targets. To train the network, the synaptic weights between units are initially assigned randomly and the output response of the network is determined. The difference between network output patterns $N(t)$ and the desired target output activations $T(t)$ is the error $E(t)$. The backpropagation algorithm calculates the weight changes that would reduce this error, and therefore implements a “gradient descent” of the error as a function of the weights (Figure 1, inset). The process of presenting input patterns and changing the weights to reduce the remaining error is iterated until the network converges on a solution with minimal error. Various training methods for recurrent networks are presented in Williams and Zipser (1989) (see also RECURRENT NETWORKS: LEARNING ALGORITHMS). It should be recognized that backpropagation is not a model for biological learning, simply an effective method of obtaining a solution. Biologically plausible learning algorithms will also find the same solutions, but usually take longer (Mazzoni, Andersen, and Jordan, 1991; see also REINFORCEMENT LEARNING IN MOTOR CONTROL).

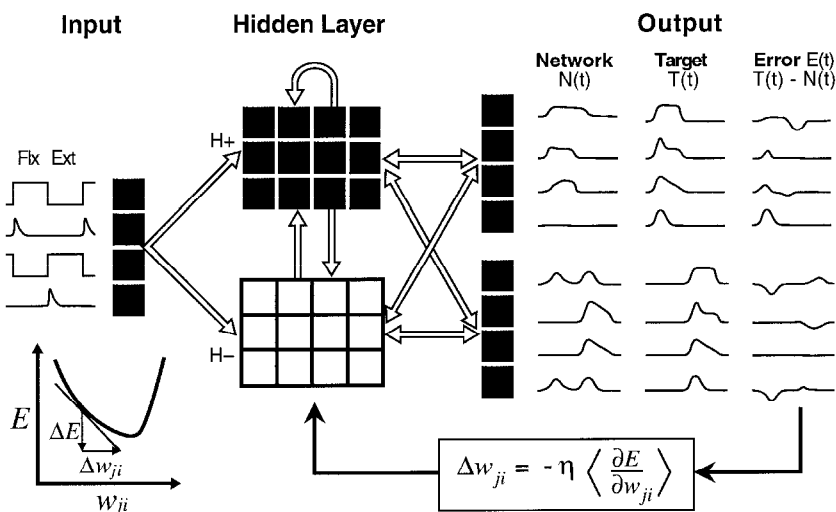


Figure 1. Typical network architecture and training procedure used with dynamic recurrent networks. This network simulates the step-tracking task. The network input consists of four representations of the step target position and target change; the output represents the firing patterns of eight representative motor units in flexor and extensor muscles. The intervening hidden units consist of excitatory and inhibitory groups, with distributed connections indicated by the open arrows. Network training proceeds by calculating the difference between the network output $[N(t)]$ and the desired target activations $[T(t)]$, and changing the connection weights in such a way as to reduce the error $[E(t)]$. Inset at lower left illustrates the error as a function of one weight, and how the gradient of this function is used to determine the appropriate weight change.

Other algorithms, such as genetic algorithms (see LOCOMOTION, VERTEBRATE) or random weight perturbations (Arnold and Robinson, 1991), can also be applied when the unit input-output functions are not differentiable.

Applications

The applications of these dynamic recurrent networks fall into three general categories:

1. *Pattern recognition* applications involve sorting of spatiotemporal input patterns into discrete categories. A set of input units receiving time-varying signals can represent a spatiotemporal pattern, and the output codes the appropriate categories.

2. *Pattern generation* networks produce temporal patterns in one or more output units, either autonomously or under the control of a gating input. These include oscillating networks (Williams and Zipser, 1989) and simulations of central pattern generators (Tsong, Cottrell, and Selverston, 1990; Rowat and Selverston, 1993; Lansner, Kotaleski, and Grillner, 1998; see also ACTIVITY-DEPENDENT REGULATION OF NEURONAL CONDUCTANCES).

3. *Pattern transformation* networks convert spatiotemporal input patterns into spatiotemporal outputs. Examples include simulations of the leech withdrawal reflex (Lockery and Sejnowski, 1992), step target tracking in the primate (Fetz, 1993), the vestibulo-ocular reflex (Arnold and Robinson, 1991; Lisberger and Sejnowski, 1992) and short-term memory tasks (Zipser, 1991; Moody et al., 1998). Recurrent networks can also simulate analytical transforms such as integration and differentiation of input signals (Munro, Shupe, and Fetz, 1994).

Oscillating Networks

Among the many examples of autonomously generated periodic motor activity to be found in biological systems are locomotion, mastication, and respiration. The neural circuitry underlying cyclic periodic movements has been called a *central pattern generator* (CPG). Williams and Zipser (1989) first trained dynamic recurrent networks to generate oscillatory activity with various frequencies. The smallest circuit that sustained quasi-sinusoidal oscillations consisted of two interconnected sigmoidal units.

Tsong et al. (1990) trained a network with the connectivity and sign constraints of neurons in the lobster gastric mill circuit to simulate their oscillatory activity. This network replicated the correct phase relations of the biological interneurons. If its activity was perturbed, the network reverted to the original pattern, indicating that the weights found by the learning algorithm represented a strong limit cycle attractor. Dynamic recurrent networks simulating the oscillatory activity of the gastric mill circuit have shown remarkably robust abilities to mimic the observed patterns (Rowat and Selverston, 1993; see also ACTIVITY-DEPENDENT REGULATION OF NEURONAL CONDUCTANCES).

Primate Target Tracking

We used dynamic networks to simulate the neural circuitry controlling forelimb muscles of the primate. In monkeys performing a step-tracking task, physiological experiments documented the discharge patterns and output connections of task-related neurons. Premotoneural (PreM) cells were identified by postspike facilitation of target muscle activity in spike-triggered averages of EMG. During alternating wrist movements, the response patterns of different PreM cells—corticomotoneuronal (CM), rubromotoneuronal (RM), dorsal root afferents, and PreM interneurons—as well as of single motor units (MU) of agonist muscles fall into specific classes (Fetz et al., 1989). All groups include cells that exhibit phasic-tonic, tonic, or phasic discharge, as well as cells with

unique firing properties. Many MUs show decremending discharge through the static hold period. Some RM cells fire during both flexion and extension, and some are unmodulated with the task.

To investigate the function of these diverse cells and to determine what other types of discharge patterns might be required to transform a step signal to the observed output of motor neurons, we derived dynamic networks that generated as outputs the average firing rates of motor units recorded in monkeys performing a step-tracking task (Figure 1). Changes in target position are represented by step inputs to the network and/or by brief transient bursts at the onset of target changes. The input signals are transformed to eight output patterns by intervening hidden units consisting of interconnected excitatory and inhibitory units.

The activation patterns and connection matrix of units in such networks are illustrated elsewhere (Fetz, 1993). In these simulations the network solutions have features that resemble biological situations but that were not explicitly incorporated: (1) Divergent connections of hidden units to different co-activated motor units are representative of divergent outputs of physiological PreM neurons (Fetz et al., 1989). (2) Some hidden units have counterintuitive discharge patterns also seen in biological neurons, e.g., bidirectional and sustained activity. (3) Different network simulations with the same architecture but initialized with different weights often converged on different solutions, comparable to the diversity of neural relations seen in biological networks.

A useful heuristic feature of these networks is the ability to quickly probe their operation with manipulations (Fetz, 1993). The contributions of hidden units can be tested by making selective *lesions* and analyzing the behavior of the remaining network. The output effects of a given unit can also be tested by delivering a simulated *stimulus* and analyzing the propagated network response. Because of changing activation levels, the effect of a stimulus depends on the time it is delivered, as is also observed in physiological experiments. These networks can also be trained to scale their responses, that is, to generate output activation patterns proportional to the size of the input. Their ability to generalize can be quickly tested by presenting different inputs.

To generate more realistic models of the primate motor system, the same approach has been used with networks incorporating additional biological features (Maier, Shupe, and Fetz, 1993): (1) the connectivity of central and segmental neurons was included with appropriate conduction delays; (2) the known activity of some central units was required to be part of the solution; and (3) in addition to the active target-tracking task, the network was required to simulate reflex responses to peripheral perturbations of the limb. The resulting networks can generate both types of behaviors and have more realistic properties. Some complex activity patterns seen in PreM neurons of monkeys, such as bidirectional responses of RM cells, also appear in the networks. Even some apparently paradoxical relations seen in monkeys, such as PreM units that covary with muscles that they inhibit, appear in networks and make contributions that are understandable in terms of other units: their activity subtracts out inappropriate components of bidirectional activity patterns. Thus, network simulations have proved useful in elucidating the function of many puzzling features of biological networks.

In contrast to such simulations of a specific neuronal system, others have modeled the representation of reaching movements, as described in REACHING MOVEMENTS: IMPLICATIONS FOR CONNECTIONIST MODELS.

Short-Term Memory Tasks

Neural mechanisms of short-term memory have been investigated in many experiments by recording cortical cell activity in animals performing instructed delay tasks. A common type of instructed

delay task involves the requirement to remember the value of a particular stimulus. Zipser (1991) trained recurrent networks to simulate short-term memory of an analog value during the delay; the resulting network implements a sample-and-hold function. The network has two inputs: an analog signal representing the stimulus value to be remembered and a gate signal specifying the times to take samples. The network output is the value of the analog input at the time of the previous gate. During the delay between gate signals, the activity of many hidden units resembles the response patterns of cortical neurons recorded in monkeys performing comparable instructed delay tasks. The activity patterns of hidden units, like those of cortical neurons, fall into three main classes: sustained activation proportional to the remembered analog value, often with a decay or rise; transient modulation during the gate signal; and combinations of the two. The network simulations allow the function of the patterns observed in the animal to be interpreted in terms of their possible role in the memory task.

We investigated such short-term memory networks to further analyze their operation. To elucidate the underlying computational algorithm, we constrained units to have either excitatory or inhibitory output weights, and reduced the network to the minimal essential network. A larger network was initially trained, then reduced by (1) combining units with similar responses and connections into one equivalent unit and (2) eliminating units with negligible activation or weak connections, then (3) retraining the smaller networks to perform the same operation. A reduced network performing the sample-and-hold function (Figure 2) consists of three excitatory and one inhibitory unit. The two inputs are the sample gate signal (S) and the analog variable (A); the output (O) is the value of A at the last sample gate. This reduced version reveals a computational algorithm that exploits the nonlinear sigmoidal input-output function of the units. The first excitatory unit (SA) carries a transient signal proportional to the value of A at the time of the gate. This signal is derived by clipping the sum of the analog and gating inputs with a negative bias, as shown by the input weights to SA in the first column. This input sample is then fed to two excitatory units (M1 and M2) that maintain their activity by reciprocal connections and also feed their summed activity to the output (M1 and M2 could also be replaced by a single self-connected M unit). The inhibitory unit (SM) carries a transient signal proportional to the previous value of A. Its value is derived from a clipped sum of the gate S and the previous values held in M1 and M2. The function of SM is to subtract the previously held value from the integrating hidden units and from the output. Thus, the network uses nonlinearity and integration to yield the appropriate remembered value.

More sophisticated recurrent networks have been derived that perform delayed matching-to-sample tasks (Moody et al., 1998). These networks identified test stimuli presented at the location of a previous sample and ignored intervening distractor stimuli. In reduced networks, the hidden units performed either storage or comparator functions. Another form of spatial memory is involved in making delayed saccades to remembered targets. This function can be simulated in networks whose inputs represent visual targets in space and eye position, and whose hidden units have recurrent connections. The outputs can represent either motor error (Xing and Andersen, 2000) or stored locations in retinal and head-centered coordinates that remain stable in the face of intervening saccades (Mitchell and Zipser, 2001).

Neural Integration

In biological motor systems, neural integrators have been postulated to transform transient commands into sustained activity and to mediate the vestibulo-ocular reflex (VOR) (see VESTIBULO-OCULAR REFLEX). Arnold and Robinson (1991) modeled the VOR integrator with a recurrent network whose connections resembled those of the vestibulo-ocular system. Two input signals represented the reciprocal responses of opposed vestibular afferents to head movement; these connected to four interneurons that were interconnected to each other and to motor neurons. Since vestibular afferents carry tonic activity in the absence of head movement, the integrator had to be configured so as to integrate only deviations from baseline, but not the baseline activity itself. The authors used units with intrinsically sustained activity with decay and a nondifferentiable rectifying input-output characteristic. To train the networks, they tweaked individual weights, and used the effect on the error to update the weights. Integration was performed through positive recurrent connections between the interneurons. The networks could mimic physiological responses to lesions and postsaccadic drift.

Lisberger and Sejnowski (1992) used dynamic networks to investigate mechanisms of learning in the vestibulo-ocular system. The network was constructed to include many anatomical and physiological constraints, including pathways through the cerebellar flocculus, with appropriate delays. The two inputs to the network, head velocity and target velocity, were converted to a single output: eye velocity. The network was initially trained to simulate three behaviors: smooth pursuit of a moving visual target, the VOR to head movement, and suppression of the VOR (when head and target move together). Then the network was required to change the gain of the VOR (as occurs after wearing magnifying or minifying goggles) and also to maintain accurate smooth pursuit visual

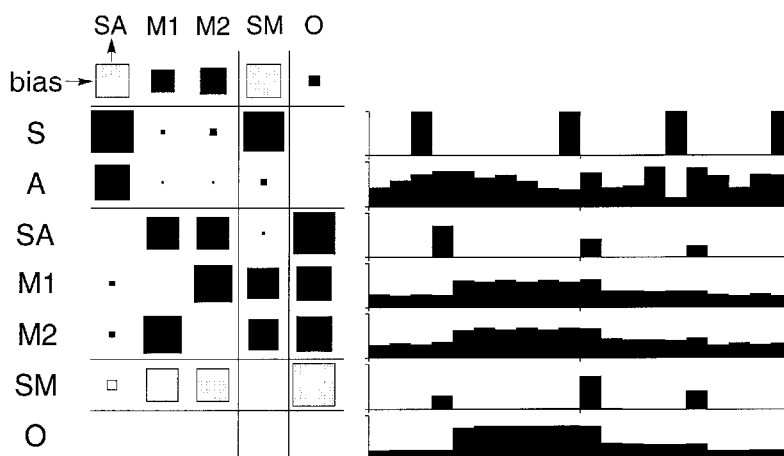


Figure 2. Reduced network performing a sample-and-hold function, simulating short-term memory. The units are indicated by abbreviations and representative activation patterns at right. The weights are indicated by squares (black = excitatory; gray = inhibitory) proportional to the connection from row unit to column unit (e.g., arrows). The two inputs are the sample signal (S) and a random analog value (A); the output (O) is the sustained value of the last sampled analog value.

tracking. Performing these functions required changes in the connection weights at both of two specific sites: the vestibular input to the flocculus and to the brainstem neurons controlling oculomotor neurons. This study exemplifies the insights gained from a biologically constrained dynamic model that can incorporate the time course of neural activity observed under different behavioral conditions, and shows the power of such simulations to reveal novel network mechanisms.

Discussion

The unique insights provided by neural network simulations assures their continued use in elucidating the operations of neural systems. The basic limitation of conventional physiological and anatomical data is that they provide a selective sample of a complex system, leaving a gap between particular glimpses of neural activity or anatomical structure and the behavior of the overall system. This gap is usually bridged by intuitive inferences, often based on selective interpretations of the data (Fetz, 1992). A more objective approach would be to derive neural network models that simulate the behavior. These models can incorporate the observed responses of units and can help explain the functional meaning of neural patterns. Thus, integrative neurophysiologists can profitably use a combination of unit recording and neural modeling to elucidate network mechanisms. To the extent that models can incorporate anatomical and physiological constraints, they can provide plausible explanations of the biological neural mechanisms mediating behavior.

Acknowledgments. Work was supported by ONR (grant No. N18-89-J-1240) and NIH grants NS12542 and RR00166.

Road Map: Biological Networks

Related Reading: Layered Computation in Neural Networks; Recurrent Networks: Learning Algorithms; Short-Term Memory

References

- Arnold, D. B., and Robinson, D. A., 1991, A learning network model of the neural integrator of the oculomotor system, *Biol. Cybern.*, 64:447–454.
- Fetz, E. E., 1992, Are movement parameters recognizably coded in the activity of single neurons? *Behav. Brain Sci.*, 15:679–690. ♦
- Fetz, E. E., 1993, Dynamic neural network models of sensorimotor behavior, in *The Neurobiology of Neural Networks* (D. Gardner, Ed.), Cambridge, MA: MIT Press, pp. 165–190. ♦
- Fetz, E. E., Cheney, P. D., Mewes, K., and Palmer, S., 1989, Control of forelimb muscle activity by populations of corticomotoneuronal and rubromotoneuronal cells, *Prog. Brain Res.*, 80:437–449. ♦
- Lansner, A., Kotaleski, J. H., and Grillner, S., 1998, Modeling of the spinal neuronal circuitry underlying locomotion in a lower vertebrate, *Ann. N. Y. Acad. Sci.*, 860:239–249.
- Lisberger, S. G., and Sejnowski, T. J., 1992, Computational Analysis Suggests a New Hypothesis for Motor Learning in the Vestibulo-ocular Reflex, Technical Report INC-9201, Institute for Neural Computation, University of California at San Diego.
- Lockery, S. R., and Sejnowski, T. J., 1992, Distributed processing of sensory information in the leech: A dynamical neural network model of the local bending reflex, *J. Neurosci.*, 12:3877–3895.
- Maier, M., Shupe, L. E., and Fetz, E. E., 1993, A spiking neural network model for neurons controlling wrist movement, *Soc. Neurosci. Abstr.*, 19:993.
- Mazzoni, P., Andersen, R. A., and Jordan, M. I., 1991, A more biologically plausible learning rule than backpropagation applied to a network model of cortical area 7a, *Cereb. Cortex*, 1:293–307.
- Mitchell, J., and Zipser, D., 2001, A model of visual-spatial memory across saccades, *Vision Res.*, 41:1575–1592.
- Moody, S. L., Wise, S. P., di Pellegrino, G., and Zipser, D., 1998, A model that accounts for activity in primate frontal cortex during a delayed matching-to-sample task, *J. Neurosci.*, 18:399–410.
- Munro, E., Shupe, L., and Fetz, E., 1994, Integration and differentiation in dynamic recurrent neural networks, *Neural Computat.*, 6:405–419.
- Rowat, P. F., and Selverston, A. I., 1993, Modeling the gastric mill central pattern generator of the lobster with a relaxation-oscillator network, *J. Neurophysiol.*, 70:1030–1053.
- Tsung, F.-S., Cottrell, G. W., and Selverston, A. I., 1990, Experiments on learning stable network oscillations, *Proceedings IJCNN-90*, vol. 1, pp. 169–174.
- Williams, R. J., and Zipser, D., 1989, A learning algorithm for continually running fully recurrent neural networks, *Neural Computat.*, 1:270–280.
- Xing, J., and Andersen, R. A., 2000, Memory activity of LIP neurons for sequential eye movements simulated with neural networks, *J. Neurophysiol.*, 84:651–665.
- Zipser, D., 1991, Recurrent network model of the neural mechanism of short-term active memory, *Neural Computat.*, 3:179–193.

Reinforcement Learning

Andrew G. Barto

Introduction

The term *reinforcement* comes from studies of animal learning in experimental psychology, where it refers to the occurrence of an event, in the proper relation to a response, that tends to increase the probability that the response will occur again in the same situation. Although not used by psychologists, the expression *reinforcement learning* has been widely adopted by theorists in artificial intelligence and engineering to refer to learning tasks and algorithms based on this principle of reinforcement. The simplest reinforcement learning methods use the commonsense idea that if an action is followed by a satisfactory state of affairs, or an improvement in the state of affairs, then the tendency to produce that action is strengthened, i.e., reinforced. The ideas of reinforcement learning have been present in engineering for many decades (e.g., Mendel and McClaren, 1970) and in artificial intelligence since its earliest days (Turing, 1950; Minsky, 1954, 1961; Samuel, 1959). It is only relatively recently, however, that the development and

application of reinforcement learning methods have occupied a significant number of researchers in these fields. Fueling this interest are two basic challenges: (1) designing autonomous robotic agents that can operate under uncertainty in complex dynamic environments, and (2) finding useful approximate solutions to very-large-scale dynamic decision-making problems.

Reinforcement learning is usually formulated as an *optimization problem* with the objective of finding a strategy for producing actions that is optimal, or best, in some well-defined way. In practice, however, it is usually more important for a reinforcement learning system to continue to improve than it is for it to actually achieve optimal behavior. Reinforcement learning differs from the more commonly studied paradigm of supervised learning, or “learning with a teacher,” in significant ways that we discuss in the course of this article. It also differs significantly from various forms of unsupervised learning. The article REINFORCEMENT LEARNING IN MOTOR CONTROL (q.v.) contains additional information. For a

more detailed introductory treatment, the reader should consult Sutton and Barto (1998); for a more in-depth mathematical treatment, the reader should consult Bertsekas and Tsitsiklis (1996).

The Reinforcement Learning Problem

Think of an agent interacting with its environment over a potentially infinite sequence of discrete-time steps $t = 1, 2, 3, \dots$. At each time step t , the reinforcement learning agent receives some representation of the environment's current *state*, $s_t \in S$, where S is the set of possible states, and on that basis executes an *action*, $a_t \in A(s_t)$, where $A(s_t)$ is the set of actions that can be executed in state s_t . One time step later, the agent receives a *reward*, r_{t+1} , a real number, and finds itself facing a new state, $s_{t+1} \in S$ (Figure 1). The reward and new state are influenced not only by the agent's action, they are also influenced by the state, s_t , in which the action was taken, and they can depend on random factors as well. Throughout this article we assume that S and $A(s)$, $s \in S$, are finite sets, but extension to infinite sets is possible, as is extension to continuous-time formulations.

The rule the agent uses to select actions is called its *policy*. It is a function, often denoted π , that for each state assigns a probability to each possible action: for all $s \in S$ and all $a \in A(s)$, $\pi(s, a)$ is the probability that the agent executes a when in state s . While interacting with its environment, a reinforcement learning agent adjusts its policy based on its accumulating experience to try to improve the the amount of reward it receives over time. More specifically, it tries to maximize the *return* it receives after each time step. The most commonly studied type of return is the *discounted return*. If $r_{t+1}, r_{t+2}, r_{t+3}, \dots$, denotes the sequence of rewards received after time step t , then the discounted return for step t is

$$\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (1)$$

where $\gamma \in [0, 1)$ is the *discount factor*. A reinforcement learning agent adjusts its policy to try to maximize the expected value of this quantity for all $t \geq 0$.

The discount factor determines the present value of future rewards. If $\gamma = 0$, the agent is only concerned with maximizing immediate rewards: its objective would be to learn how to act at each time step t so as to maximize only r_{t+1} . But in general, acting to maximize immediate reward can reduce access to future rewards, so that a longer-term return may actually be reduced. As γ approaches one, the objective takes future rewards into account more strongly: the agent becomes more far-sighted. Discounting is used because it is mathematically the simplest way to deal with cases in which the agent and environment can interact for an unbounded

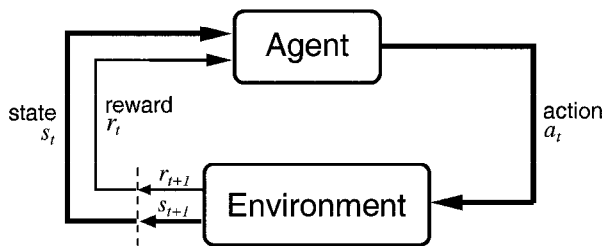


Figure 1. A reinforcement learning model. A reinforcement learning agent and its environment interact over a sequence of discrete-time steps. The *actions* are the choices made by the agent, the *states* provide the agent's basis for making the choices, and the *rewards* are the basis for evaluating these choices. (From Sutton, R. S., and Barto, A. G., 1998, *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press.)

number of time steps. In many problems only finite numbers of steps can ever happen in each learning trial, so that γ can be set to one. These are called *episodic* problems. Other definitions of return have been extensively studied as well.

This model of the reinforcement learning problem is based on the theory of *Markov decision processes* (MDPs), which has been extensively developed in decision theory and stochastic control (see, e.g., Bertsekas, 1987). An MDP has the property that the environment satisfies the Markov property, which means that environment state at any time step $t > 0$ provides the same information about what will happen next as would the entire history of the process up to step t . A full specification of an MDP includes the probabilistic details of how state transitions and rewards are influenced by states and actions, i.e., a full probabilistic model of the environment and how it is influenced by the agent's actions. The objective is to compute an *optimal policy*, i.e., a policy that maximizes the expected return from each state. In theory, this can be done using any of several stochastic dynamic programming algorithms, although their computational complexity makes them impractical for large-scale problems.

Reinforcement learning has much in common with this traditional study of MDPs, but it emphasizes approximating optimal behavior during on-line behavior instead of computing optimal policies off-line on the basis of known probabilistic models. In particular, the objective in reinforcement learning is actually not to compute an optimal policy; it is instead to allow the agent to receive as much reward as possible during its behavior. This does not always require a policy that is optimal for all possible states, since the agent may not visit all of these states while it is behaving.

Following are some key observations about the reinforcement learning problem:

1. *Uncertainty* plays a central role in reinforcement learning. The agent's environment and its own behavior can be subject to random fluctuations, so that the outcomes of decisions cannot be known beforehand with complete certainty. An accurate probabilistic model of the these uncertainties may or may not be available to the agent.
2. The reward input to the agent can be any scalar signal evaluating the agent's behavior. It might indicate just success when a goal state is reached or just failure while not reaching a goal state; or it might provide moment-by-moment evaluations of ongoing behavior (as, for example, in giving the amount of energy currently being consumed while a task is being accomplished). Moreover, multiple evaluation criteria can be combined in various ways to form the scalar reward signal (for example, via a weighted sum).
3. An important difficulty faced by a reinforcement learning system is the *credit-assignment problem* (Minsky, 1961): How do you distribute credit for success among the many decisions that may have been involved in producing it? (see also REINFORCEMENT LEARNING IN MOTOR CONTROL).
4. A reinforcement learning system often has to forgo immediate reward in order to obtain more reward later or over the long run. This kind of "sacrificing" behavior arises because the agent's actions influence not only each reward input but also the environment's state transitions. An action may be preferred because it sets the stage for a large reward later rather than for its immediate reward.
5. The reward signal does not directly tell the agent what action is best; it only evaluates the action taken. A reward input also does not directly tell the agent how to change its actions. These are key features distinguishing reinforcement learning from supervised learning, and we discuss them further below.
6. Reinforcement learning algorithms are *selectional* processes. There must be *variety* in the action-generation process so that

the consequences of alternative actions can be compared to select the best. Behavioral variety is called *exploration*; it is often generated through randomness, but it need not be.

7. Reinforcement learning involves a conflict between *exploitation* and *exploration*. In deciding which action to take, the agent has to balance two conflicting objectives: it has to exploit what it has already learned to obtain high rewards, and it has to behave in new ways—explore—to learn more. Because these needs ordinarily conflict, reinforcement learning systems have to somehow balance them. In control engineering, this is known as the conflict between control and identification.
8. Some researchers think of reinforcement learning as a form of supervised learning (because the reward input is a kind of supervision), and others think of it as a form of unsupervised learning (because the reward input is not like the label of an example). There is some truth to each of these views, but reinforcement learning is really different from both. A key distinguishing feature is the presence in reinforcement learning of the conflict between exploitation and exploration. This is absent from supervised and unsupervised learning unless the learning system is also engaged in influencing which training examples it sees.

Value Functions

The most commonly studied reinforcement learning algorithms are based on estimating *value functions*, which are scalar functions of states, or of state-action pairs, that tell how good it is for the agent to be in a state, or to take an action in a state. The notion of “how good” is the return expected to accumulate over the future, which is well-defined if the Markov property holds and the agent’s policy is specified.

If the agent uses policy π , then the state value function V^π gives the *value*, $V^\pi(s)$, of each $s \in S$, which is the return expected to accumulate over the time period after visiting s , assuming that actions are chosen according to π . For the discounted return defined by Equation 1, the value of state s is

$$V^\pi(s) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s \right]$$

where E_π is the expected value given that policy π is followed. A state’s *optimal value*, $V^*(s)$, is the return expected after visiting s , assuming that actions are chosen optimally; i.e., it is the largest expected return possible after visiting s .

Similarly, the *action value* of taking action a in state s under a policy π , denoted $Q^\pi(s, a)$, is the expected return starting from s , taking the action a , and thereafter following policy π :

$$Q^\pi(s, a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s, a_t = a \right]$$

The *optimal action value* of taking action a in state s , denoted $Q^*(s, a)$, is the expected return starting from s , taking the action a , and thereafter following an optimal policy.

Value functions are useful because of several properties of MDPs. If V^* is known, optimal policies can be found by looking ahead only one time step. That is, if s_t is the state at step t , then an optimal action is any $a \in A(s_t)$ that maximizes the expected value of $r_{t+1} + \gamma V^*(s_{t+1})$. Thus, given V^* and an accurate model of the immediate effects on the environment of all of the actions, acting optimally does not require deep look-ahead because V^* summarizes the effects of future behavior. If Q^* is known, then finding optimal actions is even easier. An optimal action at step t is any action that maximizes $Q^*(s_t, a)$. In this case, it is not necessary to look ahead one step, so that no model is needed of the effect of actions on the environment. This is what makes reinforcement

learning algorithms that use action-value functions a popular choice in many applications. Any such one-step-ahead maximizing action for a state value function, or a maximizing action for an action value function, is called a *greedy* action with respect to that function.

Value functions that depend on a policy, that is, V^π and Q^π , are useful for improving behavior because of the *policy improvement property*. Suppose the agent is deciding which action to execute in a state. It could pick an action using its current policy, π , or it could select some other action. If it picks an action that is greedy with respect to V^π , and otherwise follows π , then its performance is guaranteed to be at least as good as it would have been under π , and possibly better. This fact is the basis of the policy improvement, or policy iteration, dynamic programming algorithm, and it motivates many reinforcement learning algorithms, as we explain below.

A fundamental property of value functions is that they satisfy particular consistency conditions if the Markov property holds. For any policy π and any state s , the following is true (for the discounted return case):

$$\begin{aligned} V^\pi(s) &= E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s \right] \\ &= E_\pi \left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \middle| s_t = s \right] \\ &= E_\pi [r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s] \end{aligned} \quad (2)$$

An analogous consistency condition holds for values of Q^π . Similarly, V^* satisfies the following equation for all $s \in S$:

$$V^*(s) = \max_a E[r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a]$$

and Q^* satisfies

$$Q^*(s, a) = E[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a]$$

for all pairs (s, a) , $s \in S$, $a \in A(s)$.

If a model is available giving the probabilistic details of how the environment responds to actions, then these equations (or, more precisely, these *sets* of equations) are completely specified and can in principle be solved using one of a variety of methods for solving systems of linear equations (to obtain V^π or Q^π) or nonlinear equations (to obtain V^* or Q^*). These are often called *Bellman equations*, after Richard Bellman, who introduced the term *dynamic programming* to refer to a collection of solution methods (Bellman, 1957). There are many books that explain dynamic programming (e.g., Bertsekas, 1987).

Solving Bellman equations is therefore one route to finding optimal policies. Unfortunately, in many problems of interest one does not have the complete Markov model of the environment needed to fully define the Bellman equations, or the state set may be so large that it is not computationally feasible to exactly solve the Bellman equations. Unless some special additional structure can be exploited, one has to settle for approximate solutions.

Reinforcement Learning Based on Value Functions

Value functions are used in several different ways in reinforcement learning. One approach uses the *actor-critic architecture*, which maintains a representation of both a value function and a policy (Figure 2). To select actions, an agent using this architecture consults its current policy, represented by the *actor* component. The policy might be represented by a lookup table, by an artificial neural network, with its input coding the current state and its output coding the action to be taken, or by some other means. To evaluate the action just taken, the *critic* component is consulted, which

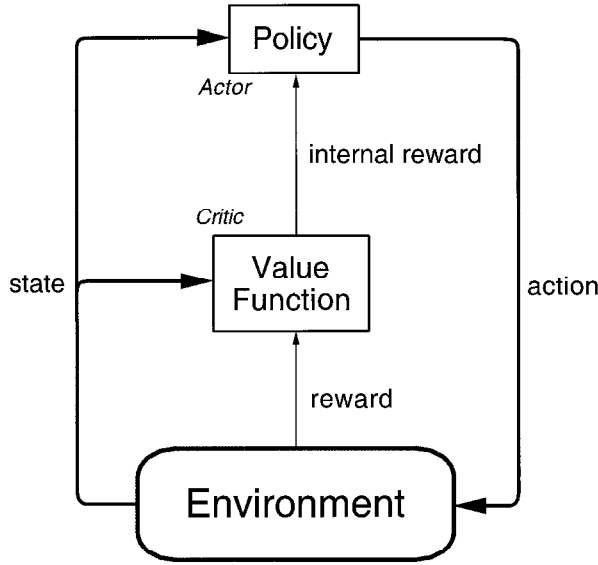


Figure 2. Actor-critic architecture. The *critic* provides an internal reward signal to an *actor*, which learns a policy for interacting with the environment.

maintains an estimate of the value function of the current policy. The action is considered to be “good” (“bad”) to the extent that it leads to a next state with a value higher (lower) than that of s , both state values being estimated by the critic. Upon receiving this evaluation, the actor updates the policy by making a good action more likely to be selected on revisiting s , or a bad action less likely, thus implementing a version of Edward Thorndike’s famous Law of Effect (Thorndike, 1911). The critic component then updates its value function estimate using a temporal difference learning algorithm of the kind described below.

Barto, Sutton, and Anderson (1983) used this architecture for learning to balance a simulated pole mounted on a cart. Their perspective was that the critic provides an internal reinforcement signal—changes in estimated values—that provides *immediate* action evaluations, even though the goal is to maximize reward over the long term. To the extent that the critic’s value estimates are correct given the actor’s current policy, the actor actually learns to increase the total amount of future reinforcement. This method thus relies on the policy improvement property. Although not a fail-safe approach from a theoretical perspective, it is often successful in improving the agent’s behavior.

Another type of reinforcement learning algorithm that uses value functions maintains an estimate of the current policy’s value function but does not keep an explicit representation of the current policy. Instead, it selects actions solely by consulting its current value function estimate. At each time step, the agent selects an action that is either greedy with respect to its current estimate of the value function or is an exploratory action chosen on some other basis (see below). If state values are being estimated, finding a greedy action requires projecting ahead one step using an environment model; if action values are being estimated, no look-ahead is required, as explained above. Like actor-critic methods, this approach also relies on the policy improvement property, but since there is no separate policy representation and no separate policy update rule, it is more closely related to various dynamic programming algorithms and is therefore somewhat easier to understand.

Estimating Value Functions

The simplest method for estimating the value function of the current policy while the agent is behaving is to average an ensemble

of returns actually observed. For example, if an agent follows policy π and maintains, for each state s encountered, an average of the actual returns that have followed that state, then the averages will converge to $V^\pi(s)$ as the number of times that state is encountered approaches infinity. If separate averages are kept for each action, a , taken in a state, then these averages will similarly converge to the action values, $Q^\pi(s, a)$. This is easiest to do in episodic problems, where return is accumulated over finite numbers of time steps. Methods like this are sometimes called *simple Monte Carlo* value estimation methods.

Another class of value estimation methods are called *temporal difference* (TD) algorithms (Sutton, 1988). The most basic TD algorithm, called *tabular TD(0)*, estimates V^π while the agent is behaving according to π and is applicable when the state set is small enough to store the state values in a lookup table with a separate entry for the value of each state. Suppose the agent is in state s , executes action a , and then observes the resulting reward r and the next state s' . TD(0) updates the current estimate of the value of state s , $V(s)$, using the following update:

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)] \quad (3)$$

where α is a positive step-size parameter. TD algorithms are based on the consistency condition expressed by the Bellman equations. This TD algorithm is designed to move the term $r + \gamma V(s') - V(s)$, called the *TD error*, toward zero for every state. If the expected TD error could be made to equal zero for every state, then the corresponding Bellman equation (Equation 2) would be satisfied. An update of this general form is often called a *backup* because the value of a state is moved toward the current value of a successor state, plus any reward that is received on the transition.

This algorithm converges to the correct state values under certain conditions (Sutton, 1988). This and other TD algorithms have been extended to include *eligibility traces*, which allow values to be backed up over more than one time step. When so extended, these are called TD(λ) algorithms, where λ is a parameter determining the temporal characteristics of the backups: λ ranges from zero (no eligibility traces as above) to one (resulting in a simple Monte Carlo method). Forms of this TD algorithm are also known as *adaptive critic algorithms*.

Another TD algorithm, known as *Q-learning*, was proposed by Watkins in 1989 (see Sutton and Barto, 1998). This algorithm directly estimates Q^* without relying on the policy improvement property. Its tabular form works as follows. Suppose the agent is in state s , executes action a , and then observes the resulting reward r and the next state s' . The Q-learning algorithm updates the action value estimate, $Q(s, a)$, of the pair (s, a) using the following backup:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (4)$$

where α is a positive step-size parameter. If α decreases appropriately with time and each state-action pair is visited infinitely often in the limit, then this algorithm converges to $Q^*(s, a)$ for all $s \in S$ and $a \in A(s)$ with probability one. Unless it is known that the environment is deterministic, the “infinitely often” requirement is necessary for this kind of strong convergence of any method that is based, as this one is, on sampling environment state transitions and rewards. Letting the agent sometimes select actions randomly from a uniform distribution is one simple way to help the agent maintain enough variety in its behavior to try to satisfy this condition. Otherwise, the agent executes actions that are greedy with respect to its current estimate of Q^* .

Closely related to Q-learning is the *Sarsa* algorithm. Suppose the agent is in state s , executes action a , observes reward r and the next state s' , and then executes action a' . Then the Sarsa update is

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$$

which is the same as the Q-learning update (Equation 4) except that instead of taking the maximum over the actions available in s' , it uses the action, a' , which was actually executed. (This requirement of s, a, r, s', a' is what accounts for the algorithm's name.) Notice that if actions are always greedy with respect to the current estimate of Q^* , then Sarsa is the same as Q-learning. Despite this similarity, Sarsa and Q-learning have somewhat different properties (see Sutton and Barto, 1998). Whereas Q-learning converges to Q^* independently of the agent's behavior (as long as the conditions for convergence are satisfied), Sarsa converges to an action value function that is optimal given the agent's mode of exploration. Like the TD algorithm for state values described above, both Q-learning and Sarsa can be extended to include eligibility traces.

TD algorithms are closely related to dynamic programming algorithms, which also use backup operations derived from Bellman equations. There are two main differences. First, a dynamic programming backup computes the expected value of successor states using the state-transition distribution of the MDP, whereas a TD backup uses a sample from this distribution. (TD backups are sometimes called *sample backups*, in contrast to the *full backups* of dynamic programming.) A second difference is that dynamic programming uses multiple exhaustive "sweeps" of the MDP's state set, whereas TD algorithms operate on states as they occur in actual or simulated experiences. These differences make it possible to use TD algorithms on problems for which it is not feasible to use dynamic programming.

Function Approximation

Instead of storing the estimated values of states or state-action pairs in lookup tables, it is possible to represent them more compactly. This is an important feature of reinforcement learning because it enables its use for problems whose state sets are too large to allow explicit representation of each value estimate, and hence too large for textbook dynamic programming algorithms to be feasible. Very large state sets often arise due to combinatorial explosions in representing states that are configurations of discrete objects. They also arise when multidimensional continuous spaces are discretized (prompting Bellman to coin the familiar phrase, "the curse of dimensionality"). For example, the game of backgammon, to which reinforcement learning has been applied with striking success (Tesauro, 1992), has more than 10^{20} states.

Any of the TD backup rules described above can be used to derive an update rule for a parameterized function approximation method of the type developed for supervised learning. Many reinforcement learning applications have used multilayer artificial neural networks and error backpropagation (see BACKPROPAGATION: GENERAL PRINCIPLES). To do this requires representing states or state-action pairs as feature vectors. Training examples are extracted from the agent's behavioral trajectory. For example, suppose one approximated the value of any state s by a function of a feature vector $\vec{\phi}(s)$ and parameter vector $\vec{\theta}$: $V(s) = f(\vec{\phi}(s), \vec{\theta})$. Then the agent's experience of observing state s , followed by reward r and successor state s' , would yield the training example consisting of input vector $\vec{\phi}(s)$ and the target output $r + \gamma f(\vec{\phi}(s'), \vec{\theta}_t)$. A gradient-descent update of $\vec{\theta}$ derived from Equation 7 is

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \alpha [r + \gamma f(\vec{\phi}(s'), \vec{\theta}_t) - f(\vec{\phi}(s), \vec{\theta}_t)] \nabla_{\vec{\theta}} f(\vec{\phi}(s), \vec{\theta}_t)$$

Notice that unlike the case in supervised learning, the target output also depends on the parameter vector. This complicates the behavior and analysis of this type of learning rule. Convergence results have been derived for TD(λ) algorithms in the case of function approximators that are linear in $\vec{\theta}$, and counterexamples to convergence have been presented for Q-learning (see Sutton and Barto, 1998, and Bertsekas and Tsitsiklis, 1996, for discussions of these

results). Despite a shortage of theoretical guarantees, many reinforcement learning systems using nonlinear function approximators have demonstrated good performance, and much current research is examining these issues.

Exploration

Reinforcement learning agents have to explore: they have to sometimes select actions that appear to be suboptimal according to their current state of knowledge (e.g., current value and/or policy estimates). Otherwise, behavior can become irretrievably suboptimal as the knowledge base comes to reflect only limited experiences. Balancing exploration with the exploitation of current knowledge is a subtle problem that has been extensively studied. In principle, it is possible to optimally balance exploration and exploitation by solving an MDP whose states are *belief states* that summarize the agent's entire history of observations and actions. But this approach is not feasible for most tasks of interest.

Several simple heuristic exploration methods are usually used in applications of reinforcement learning. In the simplest, the agent selects ϵ -greedy actions. This means that with probability $1 - \epsilon$, the agent exploits its current knowledge by selecting a greedy action, that is, an action that is optimal given its current value estimates, and with probability ϵ , it selects an action at random, uniformly, independently of its of its current value estimates. Somewhat more complicated is the *softmax* method, which selects actions according to a Boltzmann distribution based on the current action values. This gives actions with higher estimated values higher probabilities of being selected, with a "temperature" parameter determining how much an action's estimated value influences its selection probability. More sophisticated methods monitor the degree of certainty involved in action choices and direct exploration accordingly. How to design methods for balancing exploration and exploitation that are practical, effective, and amenable to theoretical treatment is an important research area.

Direct Policy Search

Not all reinforcement learning methods use value functions. It is possible to search directly in the space of policies. For example, the amount of reward that a policy yields can be estimated by running the policy for some number of time steps, possibly repeating many times from different initial states. This provides an evaluation of the entire policy that can be used to direct the search in policy space. The success of the approach usually depends on suitably parameterizing policies by vectors of real numbers so that the search can be conducted in parameter space using any of a large number of optimization algorithms. Some of these algorithms require estimates of the gradient of the policy evaluation with respect to the parameters, which can also be extracted from sample policy executions.

If the agent-environment interaction is approximately Markov, TD methods can take advantage of local consistency conditions to obtain state-localized information about how to improve a policy. On the other hand, direct policy search does not depend on the Markov property and so can be used when state information is not close to being available. Direct policy search methods also do not require the use of function approximation methods to represent value functions. Offsetting these advantages of direct policy search methods, however, is the more coarse form of credit assignment that is possible and the difficulty of efficiently evaluating entire policies. Which type of method is to be recommended is highly problem dependent. The actor-critic architecture can be considered to combine aspects of value function and direct policy search algorithms, and there is considerable interest in this hybrid approach.

Using Environment Models

Algorithms like Q-learning and Sarsa do not need a model of the agent's environment. They can learn from the agent's actual experience as the agent behaves in the real world. However, many reinforcement learning systems do take advantage of environment models. For example, algorithms like Q-learning and Sarsa are often applied to experience generated as the agent interacts with a simulation of its environment. This not only allows much faster learning (since simulations can run much faster than real time), it eliminates the potential of catastrophic consequences that can occur in some domains when a learning system is given control over a real system.

Sutton and Barto (1998) called models that can support learning from simulations *sample models*. In contrast, stochastic dynamic programming algorithms need *distribution models*, which explicitly represent the environment's state-transition and reward probabilities. Since sample models can sometimes be much easier to construct than distribution models, their ability to form policies through simulation is an important advantage of reinforcement learning methods for some applications. It is also easy to devise algorithms that learn from both real and simulated experience. Other reinforcement learning algorithms take advantage of distribution models by using full, instead of sample, backups, while still applying backups to states encountered along simulated or actual behavioral trajectories. This approach makes each backup more informative than a sample backup but avoids the exhaustive sweeping of dynamic programming.

Determining a policy from an environment model, either a distribution or a sample model, is a form of *planning*. Reinforcement learning algorithms that use models are not clearly distinct from some types of planning algorithms. Their main distinguishing characteristic is probably that they often do not fully complete a planning process before committing to actions. The planning process is extended over time, with knowledge in the form of a value function and/or a policy accumulating as behavior continues. Model-based reinforcement learning is closely related to *decision-theoretic planning* in artificial intelligence, which also makes use of the MDP formalism.

Elaborations and Extensions

Among the many topics being addressed by current reinforcement learning research are (1) extending theoretical results to include parameterized function approximation methods; (2) understanding how exploratory behavior is best introduced and controlled; (3) learning under conditions in which the environment state cannot be fully observed (related to the theory of partially observable MDPs, or POMDPs); (4) exploiting the structure present when states and/or actions are represented as vectors giving the values

of descriptive variables (formalized in terms of *factored* or *structured* MDPs); and (5) introducing various forms of abstraction such as temporally extended actions and hierarchy (which rely strongly on the theory of semi-Markov decision processes, or SMDPs). Finally, researchers are studying the relationship of computational reinforcement learning theories to brain reward mechanisms. Strong parallels exist between TD learning and the activity of dopamine neurons (Schultz, 1998; see also DOPAMINE, ROLES OF).

Road Maps: Grounding Models of Neurons; Learning in Artificial Networks

Related Reading: Dopamine, Roles of; Q-Learning for Robots; Reinforcement Learning in Motor Control

References

- Barto, A. G., Sutton, R. S., and Anderson, C. W., 1983, Neuronlike elements that can solve difficult learning control problems, *IEEE Trans. Syst. Man Cybern.*, 13:835–846. Reprinted in *Neurocomputing: Foundations of Research* (J. A. Anderson and E. Rosenfeld, Eds.), Cambridge, MA: MIT Press, 1988, pp. 535–549.
- Bellman, R. E., 1957, *Dynamic Programming*, Princeton, NJ: Princeton University Press.
- Bertsekas, D. P., 1987, *Dynamic Programming: Deterministic and Stochastic Models*, Englewood Cliffs, NJ: Prentice-Hall. ♦
- Bertsekas, D. P., and Tsitsiklis, J. N., 1996, *Neuro-Dynamic Programming*, Belmont, MA: Athena Scientific. ♦
- Mendel, J. M., and McLaren, R. W., 1970, Reinforcement learning control and pattern recognition systems, in *Adaptive Learning and Pattern Recognition Systems: Theory and Applications* (J. M. Mendel and K. S. Fu, Eds.), New York: Academic Press, pp. 287–318.
- Minsky, M. L., 1954, Theory of neural-analog reinforcement systems and its application to the brain-model problem, Ph.D. diss., Princeton University.
- Minsky, M. L., 1961, Steps toward artificial intelligence, *Proc. Inst. Radio Eng.*, 49:8–30. Reprinted in *Computers and Thought* (E. A. Feigenbaum and J. Feldman, Eds.), New York: McGraw-Hill, 1963, pp. 406–450.
- Samuel, A. L., 1959, Some studies in machine learning using the game of checkers, *IBM J. Res. Dev.*, 3:210–229. Reprinted in *Computers and Thought* (E. A. Feigenbaum and J. Feldman, Eds.), New York: McGraw-Hill, 1963, pp. 71–105.
- Schultz, W., 1998, Predictive reward signal of dopamine neurons, *J. Neurophysiol.*, 80:1–27.
- Sutton, R. S., 1988, Learning to predict by the method of temporal differences, *Machine Learn.*, 3:9–44.
- Sutton, R. S., and Barto, A. G., 1998, *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press. ♦
- Tesauro, G. J., 1992, Practical issues in temporal difference learning, *Machine Learn.*, 8:257–277.
- Thorndike, E. L., 1911, *Animal Intelligence*, Darien, CT: Hafner.
- Turing, A. M., 1950, Computing machinery and intelligence, *Mind*, 59:433–460. Reprinted in *Computers and Thought* (E. A. Feigenbaum and J. Feldman, Eds.), New York: McGraw-Hill, 1963, pp. 11–35.

Reinforcement Learning in Motor Control

Andrew G. Barto

Introduction

How do we learn motor skills such as reaching, walking, swimming, or riding a bicycle? There is a large literature on motor skill acquisition that is full of controversies (for an introduction to human motor control, see Schmidt and Lee, 1999), but there is general agreement that motor learning requires the learner, human or not,

to receive response-produced feedback through various senses providing information about performance. Careful consideration of the nature of the feedback used in learning is important for understanding the role of reinforcement learning in motor control (see REINFORCEMENT LEARNING). One function of feedback is to guide the performance of movements. This is the kind of feedback with which we are familiar from control theory, where it is the basis of

servocontrol, although its role in guiding animal movement is more complex. Another function of feedback is to provide information useful for improving *subsequent* movement. Feedback having this function has been called *learning feedback*. Note that this functional distinction between feedback for control and feedback for learning does not mean that the signals or channels serving these functions need to be different.

Learning Feedback

When motor skills are acquired without the help of an explicit teacher or trainer, learning feedback must consist of information automatically generated by the movement and its consequences on the environment. This has been called *intrinsic feedback* (Schmidt and Lee, 1999). The “feel” of a successfully completed movement and the sight of a basketball going through the hoop are examples of intrinsic learning feedback. A teacher or trainer can augment intrinsic feedback by providing *extrinsic feedback* (Schmidt and Lee, 1999) consisting of extra information added for training purposes, such as a buzzer indicating that a movement was on target, a word of praise or encouragement, or an indication that a certain kind of error was made.

Most research in the fields of machine learning and artificial neural networks has focused on the learning paradigm called *supervised learning*, which emphasizes the role of training information in the form of desired, or “target,” network responses for a set of training inputs (see PERCEPTRONS, ADALINES, AND BACKPROPAGATION). However, motor learning is more complex than supervised learning, even when it involves extrinsic feedback provided by a trainer. For example, a trainer can tell or show us what to do, explicitly guide our movements, give us hints on how to deal with difficult parts of a skill, tell us when we have improved or done badly, etc. The aspect of real training that corresponds most closely to the supervised learning paradigm is the trainer’s role in telling or showing the learner what to do, or explicitly guiding his or her movements. These activities provide standards of correctness that the learner can try to match as closely as possible by reducing the error between his or her behavior and the standard. Supervised learning can also be relevant to motor learning when there is no trainer, because intrinsic feedback can be used to learn various kinds of *models* that are useful for motor control. Kawato (1999) and Desmurget and Grafton (2000) discuss some of the uses of models in motor control.

In contrast to supervised learning, *reinforcement learning* emphasizes learning feedback that *evaluates* the learner’s performance without providing standards of correctness in the form of behavioral targets (see REINFORCEMENT LEARNING). Although the most obvious evaluative feedback is extrinsic feedback provided by a trainer, most evaluative feedback is probably intrinsic, being derived by the learner from sensations generated by a movement and its consequences on the environment: the kinesthetic and tactile feel of a successful grasp or the swish of a basketball through the hoop. Evaluative feedback is often called *reinforcement* feedback (and it need not involve pleasure or pain). A reinforcement learning system has to actively try alternatives, compare the resulting evaluations, and use some kind of selection mechanism to guide behavior toward the better alternatives. This basic idea follows Thorndike’s classical law of effect (Thorndike, 1911) and is commonly called learning by trial and error (not to be confused with error-correction, or supervised, learning).

The great Russian physiologist Nikolai Bernstein discussed the role of trial-and-error learning in motor control in his classic 1967 book (Bernstein, 1967). He distinguished his view from the concept of random undirected search, which he attributed to the behaviorists. According to Bernstein, the process must be an active search involving “gradient extrapolation” by probabilistic sampling so that each attempt is informed by previously acquired information about

“how and where the next step must be taken.” This is very much in accord with modern concepts of reinforcement learning, where randomness is often used to generate behavioral variety, but action selections are strongly constrained by evaluations of earlier experience (see REINFORCEMENT LEARNING). To Bernstein, this kind of search was important for motor behavior, especially for movements requiring high levels of coordination. Another motor control theorist, Jack Adams, provided an interesting discussion of the role of the law of effect in motor control in a 1978 article (Adams, 1978). Although he called into question some of the details of Thorndike’s theories, he affirmed the importance of reinforcement learning in motor control.

Motor learning involves feedback carrying many different kinds of information. Consequently, it is incorrect to view motor learning strictly in terms of supervised, reinforcement, or any other learning paradigms that have been formulated for theoretical study. Aspects of all of these paradigms play interlocking roles, with their relative importance undoubtedly varying with the type of task as well as the developmental stage. However, reinforcement learning may be an essential component of motor learning simply because evaluative feedback is more easily obtained than many other kinds of learning feedback.

Learning from Consequences

To illustrate how reinforcement learning applies to motor learning, we first discuss it within the general context of control. Then we describe several special cases related to motor control. Figure 1, panel A, is a variation of the classical control system diagram. A controller provides control signals to a controlled system. The behavior of the controlled system is influenced by disturbances, and feedback from the controlled system to the controller provides information on which the control signals can depend. Commands to the controller specify aspects of the control task’s objective.

In Figure 1, panel B, the control loop is augmented with another feedback loop that provides learning feedback to the controller. In accordance with common practice in reinforcement learning, a

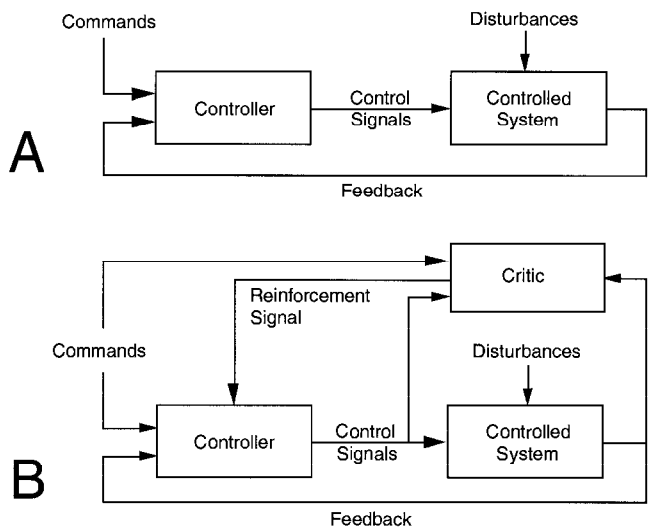


Figure 1. A, A basic control loop. A controller provides control signals to a controlled system, whose behavior is influenced by disturbances. Feedback from the controlled system to the controller provides information on which the control signals can depend. Commands to the controller specify aspects of the control task’s objective. B, A control system with learning feedback. A critic provides the controller with a reinforcement signal evaluating its success in achieving the control objectives.

critic is included that generates evaluative learning feedback on the basis of observing the control signals and their consequences on the behavior of the controlled system. The critic also needs to know the command to the controller because its evaluations must be different depending on what the controller should be trying to do. The critic is an abstraction of whatever process supplies evaluative learning feedback, both intrinsic and extrinsic, to the learning system. It is often said that the critic provides a *reinforcement signal* to the learning system. In most artificial reinforcement learning systems, the critic's output at any time is a number that scores the controller's behavior: the higher the number, the better the behavior. Assume for a moment that the behavior being scored is the immediately preceding unit of behavior. (We discuss what a unit of behavior might be, as well as more complex temporal relationships, in later sections.) For this process to work, there must be some *variability* in the controller's behavior so that the critic can evaluate many alternatives. A learning mechanism can then adjust the controller's behavior so that it tends toward behavior that is favored by the critic.

A learning rule particularly suited to reinforcement learning control systems implemented as artificial neural networks was developed by Gullapalli (1990) in the form of what he called a Stochastic Real-Valued (SRV) unit. An SRV unit's output is produced by adding a random number to the weighted sum of the components of its input pattern. The random number is drawn from a zero-mean Gaussian distribution. This random component provides the unit with the variability necessary for it to "explore" its activity space. When the reinforcement signal indicates that something good happened just after the unit emitted a particular output value in the presence of some input pattern, the unit's weights are adjusted to move the activation in the direction in which it was perturbed by the random number. This has the effect of increasing the probability that future outputs generated for that input pattern (and similar input patterns) will be closer to the output value just emitted. If the reinforcement signal indicates that something bad happened, the weights are adjusted to move future output values away from the value just emitted. Another part of the SRV learning rule decreases the variance of the Gaussian distribution as learning proceeds. This decreases the variability of the unit's behavior, with the goal of making it eventually stick (i.e., become deterministic) at the best output value for each input pattern. Using this learning rule, an SRV unit learns to produce the best output in response to each input pattern (given appropriate assumptions). Unlike more familiar supervised learning units, it is never given target outputs; it has to discover what outputs are best through an active exploration process.

Overcoming the Distal Error Problem

As a simple illustration of how reinforcement learning can be useful in motor learning, consider the problem of learning to reach to

specific points in space starting from a variety of initial hand positions. Lipitkas et al. (1993) proposed a particularly straightforward method (although not as a model of the human learning process, which is much more complex). Their controller is an artificial neural network receiving inputs coding the initial spatial location and the desired, or target, spatial location of the hand (ignoring hand orientation). The six outputs of the network provide parameters to a torque generator that generates time-varying signals for driving the joint actuators of a dynamic arm model (Figure 2). The time-varying signals are parameterized by six numbers determining characteristics of their wave-like shapes (e.g., giving the magnitudes and relative timing of the half-waves). During each movement, the controller operates in open-loop mode, generating the torque time functions without the aid of sensory feedback. The problem for the network is to learn a function associating each pair of hand starting and target positions with the values of the six torque-generator parameters that will accomplish the movement.

A straightforward application of supervised learning is not possible here because the required training examples are not available: it is not known what parameters will work for any pair of starting and target positions (except possibly the trivial cases in which the starting position is already the target position, but these are not useful as training examples). This is an instance of what has been called the *distal error problem* (Jordan and Rumelhart, 1992) for supervised learning. This problem is present whenever the standard of correctness required for supervised learning is available in a coordinate system that is different from the one in which the learning system's activity must be specified for learning. In the case of learning how to move the hand from a starting position to a target position, the standard of correctness is the target position, but what must be learned is the control signals to the joint actuators, that is, to the muscles. The hand position error is distal to the output of the controller that has to be learned. Although a non-zero distal error vector indicates that the controller made an error, it does not tell the controller how it should change its output in order to reduce the error.

The distal error problem can be solved by using a model of the controller's influence on the arm's movement (possibly learned via supervised learning) to translate distal error vectors into error vectors required for supervised learning (Jordan and Rumelhart, 1992). Another approach is to learn an inverse model of the controller's influence on the arm's movement (Jordan and Rumelhart, 1992; Kawato, 1999). Reinforcement learning offers another way to overcome the distal error problem because it does not need learning feedback in the form of error vectors. Continuing with the reaching example, Lipitkas et al. (1993) defined a reinforcement signal that attains a maximum value of 1 if the hand reaches the desired position and stops there. The signal decreases, depending on the distance between the hand's final position and the target position and on its tangential velocity as it passes the target position. The reinforcement signal could include other criteria of successful move-

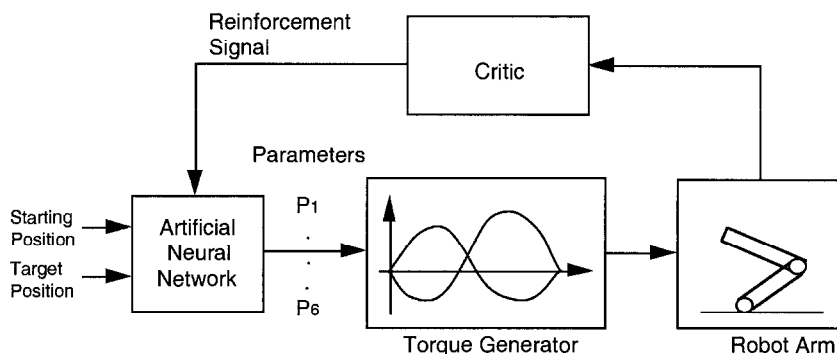


Figure 2. Block diagram of a reinforcement learning controller of an arm. Given inputs coding the starting and target positions of the hand, the network controller learns to provide correct parameters to a torque generator that generates, in open-loop mode, time-varying torque signals to the arm. The reinforcement signal evaluates the success of each movement after its completion. (Modified from Lipitkas et al., 1993, Figure 1.)

ments as well. With inputs coding starting and target hand positions, the network employs SRV units to generate six parameter values using its current weights. The torque generator generates a movement using these parameter values. When the movement is completed, it is scored by the reinforcement signal, and the network's weights are changed according to Gullapalli's SRV learning rule. After a few thousand movements with different starting and target hand positions, the system could move with reasonable accuracy for new pairs of starting and target positions as well as for the pairs on which it was trained. This amount of practice is required because the system effectively has to search the six-dimensional parameter space for each starting and target position. A more complicated example of reinforcement learning using SRV units is the work on biped walking by Benbrahim and Franklin (1997).

The relative advantages and disadvantages of supervised and reinforcement learning approaches to the distal error problem have been discussed by many researchers. It is clear that reinforcement learning approaches are simpler, but reinforcement learning is usually slower in terms of the amount of experience required for learning. This is true because reinforcement learning methods extract less information from each experience than do the model-based supervised approaches. However, in some problems it is easier to learn the right actions than it is to model their effects on a complicated process. Reinforcement learning methods are also more plausible from the perspective of neuroscience (see below), while the backpropagation process often used by supervised approaches is more difficult to reconcile with what we know about neural mechanisms. In practical terms, which approach is more advantageous will depend on aspects of the specific problem being considered.

Collective Behavior

Another property of reinforcement learning that might be relevant to motor control is the ability of a "team" of reinforcement learning systems to learn to cooperate so that the team as a whole improves performance. Here is an example presented in a 1965 lecture by the cybernetician Mikhail Tsetlin (Tsetlin, 1973), a pioneer in the study of simple reinforcement learning systems called learning automata. He presented the basic idea as follows in terms of human players (the so-called Goore game). Suppose there is a referee and some number of players. The referee can see the players, but the players cannot see one another. At the sound of a buzzer, each player is to raise one or two fingers. The referee determines what percentage of players raised one finger, then pays each player a fixed amount with a probability that depends only on this percentage (and is the same for each player). The process repeats each time the buzzer sounds. It turns out that for any number of players each implementing a sufficiently competent reinforcement learning rule, eventually each player will settle on raising either one or two fingers, so that the percentage of those raising one finger is (with probability close to 1) a local maximum of whatever payoff function the referee uses. This occurs with no direct communication among the players and no agreements of any kind among them.

It is possible to extend this result to one in which the referee provides payments based not just on the percentage of players raising one finger, but on *any function whatsoever* of the pattern of players' fingers. One can see how this is an instance of the problem of learning with a distal teacher, with the added complication that the payoff, or reinforcement signal, to each player is extremely noisy due to the noise introduced by the actions of the other players (in addition to the referee's probabilistic payoff method).

Tsetlin speculated that the recruitment of motor units can be reduced to this type of problem. Here, the problem would be to activate the right number of motor units to obtain a pull of a given

force. The referee corresponds to a process that evaluates the results of the collective behavior of the entire pool of motor units on the resulting force. The collective behavior of reinforcement learning systems has been studied by many researchers (e.g., Narendra and Thathachar, 1989; Barto, 1985), although no modern work following up Tsetlin's suggestion about motor unit recruitment appears to exist.

Credit Assignment Problems

The challenge of reinforcement learning is often summed up as various kinds of *credit assignment* problems. A scalar evaluation of a complex mechanism's behavior does not indicate which of its many action components, both internal and external, were responsible for the evaluation. This makes it difficult to determine which of these components deserve the credit (or the blame) for the evaluation. This problem is sometimes referred to as the *structural credit assignment* problem: How is credit assigned to the internal workings of a complex structure? One approach is to assign credit equally to *all* the components, so that, through a process of averaging over many variations of the behavior, the components that are key in producing laudable behavior end up gaining the most strength, while inappropriate components are weakened. This is the general approach illustrated above by the Goore game.

The fact that reinforcement learning can work under these circumstances makes neural implementation quite plausible. A single reinforcement signal uniformly *broadcasted* to all the sites of learning, either neurons or individual synapses, is consistent with anatomical and physiological evidence showing the existence of diffusely projecting neural pathways by which neuromodulatory chemicals can be widely and nonspecifically distributed. It has been suggested that some of these pathways may play a role in reward-mediated learning. A specific hypothesis is that dopamine mediates synaptic enhancement in the corticostriatal pathway in the manner of a broadcasted reinforcement signal (see DOPAMINE, ROLES OF). This may be one of the ways in which reinforcement learning is implemented for motor control.

Another aspect of the credit assignment problem occurs when the temporal relationship between a system's behavior and evaluations of that behavior is not as simple as assumed above. How can reinforcement learning work when the learner's behavior is temporally extended and evaluations occur at varying and unpredictable times? Under these more realistic conditions, it is not always clear what elements of behavior are being evaluated. This has been called the *temporal credit assignment* problem. It is especially relevant in motor control because movements extend over time and evaluative feedback may become available only after the end of a movement. An approach to this problem that is receiving considerable attention is the use of methods by which the critic itself can learn to provide useful evaluative feedback immediately after the evaluated event. According to this approach, reinforcement learning is not only the process of improving behavior according to given evaluative feedback; it also includes learning how to improve the evaluative feedback itself. The strong parallels between algorithms for adapting evaluative feedback (temporal difference methods; see REINFORCEMENT LEARNING) and the properties of dopamine-producing neurons in the brain (see DOPAMINE, ROLES OF) make it plausible that the brain uses similar methods for dealing with the temporal credit assignment problem.

The modern view of reinforcement learning developed by machine learning researchers uses the framework of stochastic optimal control to study the temporal credit assignment problem (see REINFORCEMENT LEARNING). From this perspective, reinforcement learning algorithms are methods for approximating solutions to complex stochastic optimal control problems via relatively simple mechanistic learning rules. Because optimality principles have

played significant roles in theories of motor control (Engelbrecht, 2001), and because stochasticity may be an important element of motor control (Harris, 1998), the modern theory of reinforcement may prove to be of great utility in extending our understanding of motor learning.

Discussion

As this article has emphasized, motor learning is too complex to be viewed strictly in terms of either supervised learning or reinforcement learning. Feedback used in motor learning ranges from specific standards of correctness to nonspecific evaluative information, and many learning mechanisms with differing characteristics probably interact to produce the motor learning capabilities of animals. However, reinforcement learning principles may be indispensable for motor learning because they seem necessary for improving motor performance when the standards of correctness required by supervised learning are not available.

Road Maps: Mammalian Motor Control; Robotics and Control Theory

Background: Reinforcement Learning

Related Reading: Basal Ganglia; Dopamine, Roles of; Q-Learning for Robots

References

- Adams, J. A., 1978, Theoretical issues for knowledge of results, in *Information Processing in Motor Control and Learning* (G. E. Stelmach, Ed.), New York: Academic Press, pp. 229–240.
- Barto, A. G., 1985, Learning by statistical cooperation of self-interested neuron-like adaptive elements, *Hum. Neurobiol.*, 4:229–256.
- Benbrahim, H., and Franklin, J. A., 1997, Biped dynamic walking using reinforcement learning, *Robot. Auton. Systems*, 22:283–302.
- Bernstein, N., 1967, *The Co-ordination and Regulation of Movements*, Oxford, Engl.: Pergamon Press.
- Desmurget, M., and Grafton, S., 2000, Forward modeling allows feedback control for fast reaching movements, *Trends Cognit. Sci.*, 4:423–431. ♦
- Engelbrecht, S. E., 2001, Minimum principles in motor control, *J. Math. Psychol.*, 45:497–542. ♦
- Gullapalli, V., 1990, A stochastic reinforcement algorithm for learning real-valued functions, *Neural Netw.*, 3:671–692.
- Harris, C. M., 1998, On the optimal control of behavior: A stochastic perspective, *J. Neurosci. Methods*, 83:73–88.
- Jordan, M. I., and Rumelhart, D. E., 1992, Supervised learning with a distal teacher, *Cognit. Sci.*, 16:307–354.
- Kawato, M., 1999, Internal models for motor control and trajectory planning, *Curr. Opin. Neurobiol.*, 9:718–727.
- Lipitkas, J., D'Eleuterio, G. M. T., Bock, O., and Grodski, J. J., 1993, Reinforcement learning and the parametric motor control hypothesis applied to robotic arm movements, in *Proceedings of the DND Workshop on Advanced Technologies*, Ottawa, CDN, 1993.
- Narendra, K., and Thathachar, M. A. L., 1989, *Learning Automata: An Introduction*, Englewood Cliffs, NJ: Prentice Hall.
- Schmidt, R. A., and Lee, T. D., 1999, *Motor Control and Learning: A Behavioral Emphasis*, 3rd ed., Champaign, IL: Human Kinetics Publishers. ♦
- Thorndike, E. L., 1911, *Animal Intelligence*, Darien, CT: Hafner.
- Tsetlin, M. L., 1973, *Automata Theory and Modeling of Biological Systems*, New York: Academic Press.

Respiratory Rhythm Generation

Richard J. A. Wilson, John E. Lewis, and John E. Remmers

Introduction

After several decades of intense debate, fueled by exciting experimental advance, a number of fundamental issues regarding the neuronal mechanisms that synthesize normal breathing (eupnea) remain unresolved. This article examines recent data and evaluates insights from modeling studies. We begin by discussing potential neuronal components of the respiratory rhythmogenic network and then assess current models of the respiratory oscillator in the context of the mechanisms of rhythmogenesis.

The apparent simplicity and reliability of breathing are enticing to the experimenter and modeler alike. During inspiration, activity in the phrenic nerve contracts the diaphragm, sucking air into the lungs. During expiration, passive recoil of the lungs, diaphragm, and ribs almost suffices for stale air expulsion. However, a closer look reveals a fascinating and complex behavior involving recruitment of many muscles (facial, upper airway, thoracic, abdominal, postural), modulation from many different sources (mechanosensory, chemosensory, descending), and coordination with many other behaviors (e.g., swallowing, locomotion).

In light of this complexity, perhaps one of the most remarkable ideas to arise in the field over the last decade is that only a small kernel of brainstem neurons (perhaps as few as 1,200) are responsible for the rudimentary respiratory rhythm (Gray et al., 2001). This idea originated from studies in vitro. If this idea is correct, then in vivo these neurons should meet *all* the following criteria: (1) have activity correlated with breathing, (2) be necessary for breathing (i.e., breathing is abolished if they are functionally

ablated), and (3) be sufficient for breathing (functionally defined as capable of accelerating breathing when stimulated). Unfortunately, demonstrating all but the last criterion experimentally has proved to be very difficult, as reviewed below.

Possible Components of Rhythm Generator

Correlation

Respiratory neurons are generally defined by and categorized in relation to inspiratory phrenic discharge. At least six categories have been identified (e.g., Richter, Ballanyi, and Schwarzscher, 1992): pre-inspiratory (PreI), early inspiratory (EI), throughout inspiratory (I), late inspiratory (LateI), post-inspiratory (PostI), and expiratory (E2). These categories are found across different species and, some claim, experimental preparations. Note, however, that while the phrenic discharge during eupnea consists of augmenting bursts, phrenic activity changes dramatically during other behaviors such as hiccupping, coughing, and gasping. For unambiguous identification of eupneic respiratory neurons therefore, the *pattern* of the phrenic discharge must be considered.

Correlation is but one of the three criteria that should be applied before a neuron can be given a functional definition. A neuron that fires in relation to the eupneic rhythm could conceivably be more important (i.e., necessary) for some other behavior. Furthermore, despite the presence of several categories of respiratory neurons, the generation of the eupneic rhythm may require but a subset of the types found.

Further details of the data discussed below may be found in the papers by Ballanyi, Onimaru, and Homma (1999); St. Jacques and St. John (1999); and St. John, (1996). Further information on models is given by Smith et al. (2000).

Sufficiency and Necessity

The brainstem alone can produce the respiratory rhythm. Rhythmic augmenting bursts in cranial nerves, a characteristic of eupnea, continues in vagotomized animals after transections at the midcollicular level and at the obex. Several regions in the brainstem cause acceleration of respiratory rhythm when stimulated (i.e., are sufficient). Two of these areas in the medulla, the pre-Bötzinger complex (preBötC) and the pre-inspiratory region have received particularly intense study.

The preBötC is situated rostral to the ventral respiratory group and ventral to the nucleus ambiguus (Rekling and Feldman, 1998). Gray et al. (2001) determined that the preBötC can be defined anatomically according to the distribution of Substance P receptors. The pre-inspiratory region lies just rostral and ventral to the preBötC (see Ballanyi, Onimaru, and Homma, 1999).

The preBötC contains the complete suite of respiratory neurons, and several groups have shown it to be necessary for eupnea. Ramirez et al. showed that in anesthetized, artificially ventilated, vagotomized cats, irreversible inactivation of this region with tetrodotoxin caused the cessation of the eupneic motor pattern but not anoxia-induced gasping. Gray et al. (2001) demonstrated that in awake rats, four to five days after injection of poison-conjugated Substance P into the preBötC, blood gases and breathing were abnormal, with ablated cells concentrated in, though not limited to, the preBötC.

Other *in vivo* experiments have generated contrasting results. Huang et al. have reported that lesioning both preBötC and pre-inspiratory regions of anesthetized, artificially ventilated, vagotomized neonatal rats with kainic acid had no effect on eupnea but eliminated anoxia-induced gasping. Lucid histological boundaries of the preBötC have been identified only recently. Therefore, discrepant results between lesion studies may reflect ablations in different regions. While this explanation may explain why some studies failed to eliminate eupnea following ablation of sites in the ventral medulla, in other studies, the lesions seem too extensive to have spared the preBötC. St. Jacques and St. John (1999) found that massive kainic acid injections (whether ipsilateral or bilateral) in the vicinity of the preBötC eliminate eupnea but only transiently. Gasping, however, was permanently abolished. While these results support the importance of the preBötC for eupnea, they also suggest that it may not be necessary, at least in the strictest sense, and that sites other than the preBötC can produce the eupneic rhythm. Where might these sites be?

One candidate is the pre-inspiratory area. This area contains large numbers of neurons that begin firing before inspiration (e.g., PreI and PostI) and, when stimulated with single shocks, can reset the respiratory rhythm. In addition, brainstem areas rostral to the medulla (i.e., in the pons) may also be necessary for eupnea. The pons also contains large numbers of respiratory neurons and may have an inherent rhythm-generating capability (St. John, 1996). It is possible that this pontine rhythm generator may interact with that in the medulla to generate breathing. Indeed, transecting the brainstem at the pontine-medulla level causes profound changes in phrenic nerve activity, to a pattern that is no longer eupneic and therefore could be produced by a different brainstem circuit (see St. John, 1996).

In summary, there is mounting evidence that the preBötC plays an important role in respiratory rhythmogenesis (Rekling and Feldman, 1998). However, we are of the opinion that the preBötC might not act alone to produce the respiratory rhythm: other areas, most

notably the pons, may also be necessary for eupnea. Slices containing the preBötC complex are capable of producing rhythmic activity. However, elevated extracellular $[K^+]$ is usually required, and some argue that the rhythm produced is more gasp-like than eupneic (St. John, 1996). The ability to generate a rhythm when isolated from other neuronal structures does not dictate that the preBötC is solely responsible for the eupneic rhythm in the intact animal. In our thinking, we allow for the possibility that sites other than the preBötC may be necessary and that rhythmogenesis may be distributed.

Mechanism of Rhythm Generation

Despite the difficulty of localizing categorically the necessary and sufficient components responsible for the eupneic rhythm, models have been proposed for the mechanism by which the rhythm is generated. These models fall into two distinct groups: (a) *hybrid-pacemaker models*, in which the basic rhythm is generated by the endogenous pacemaker-like membrane properties of a small subset of respiratory neurons (Figure 1A), and (b) *network models*, in which the rhythm is generated by synaptic interactions within a large network (Figure 1B). These models are based largely on different data, obtained from *in vitro* and *in vivo* experiments, respectively.

Hybrid-Pacemaker Models

These models predict that the basic rhythm stems from a small population of respiratory neurons. These neurons, when synaptically isolated, burst as a result of inherent membrane properties (e.g., Ramirez and Richter, 1996; Rekling and Feldman, 1998; Smith et al., 2000). In these models, synaptic interactions between respiratory neurons are not necessary for rhythmogenesis *per se* but function primarily to elaborate, shape, and transmit the basic rhythm to generate the eupneic motor pattern.

Evidence that some respiratory neurons have pacemaker-like membrane properties arises mainly from *in vitro* neonatal preparations. For example, I and PostI neurons in the pre-inspiratory area of the *in vitro* neonatal rat brainstem-spinal cord preparation retained rhythmic bursts of action potentials after blocking inhibitory synaptic transmission either with GABA and glycine antagonists

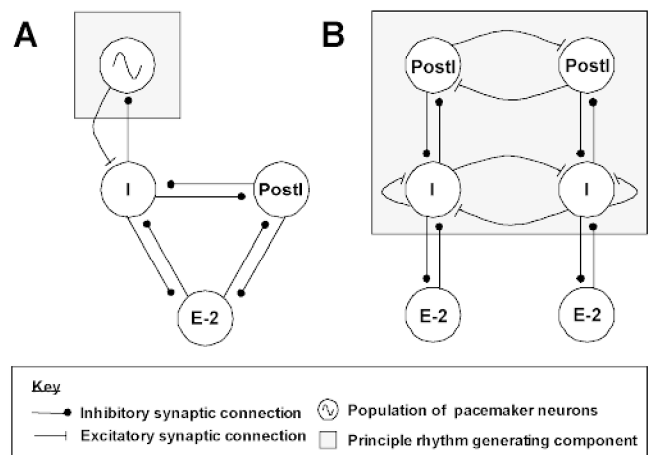


Figure 1. Two different models of the eupneic respiratory rhythm generator. *A*, Hybrid pacemaker model. *B*, Network model. Shaded boxes encapsulate key rhythm-generating components. Each circle represents a population of neurons.

or by bathing the preparation in a low Cl^- solution (Ballanyi et al., 1999). Similarly, in the transverse slice preparation from the neonatal rat, rhythmic bursts persisted in I and E2 preBötC neurons despite blocking of chemical synaptic connections with low Ca^{2+} (Johnson et al., 1994).

These important experiments suggest that some respiratory neurons have endogenous bursting properties, at least under certain conditions. However, Johnson et al. (1994) identified a third class of cell whose activity was transformed from tonic to bursts when exposed to low Ca^{2+} . This illustrates the difficulty of ascertaining the normal properties of neurons when they are examined outside their usual ionic and synaptic realms. In this light, one should note that the in vitro neonatal rat brainstem-spinal cord preparation has an anoxic, acidic (pH ~ 6.5) core that compromises synaptic inhibition and produces a gasp-like motor pattern (see St. John, 1996). One should also note that the preBötC in the in vitro slice preparation has lost many synaptic inputs and usually requires elevated $[\text{K}^+]$ for rhythmogenesis. Therefore, bursting cells in vitro may operate at membrane potentials that are not normally encountered. Whether endogenous bursting drives eupnea in vivo will be an important question for future research.

The hybrid-pacemaker model was first implemented by using conductance-based Hodgkin-Huxley-type model neurons, each with up to nine ionic conductances (see Smith et al., 2000). The early simulations consisted of a kernel of endogenous bursting neurons, which fired during the inspiratory phase. Burst initiation was determined largely by a subthreshold activating persistent Na^+ conductance ($I_{\text{Na(p)}}$), whereas a persistent K^+ conductance was critical for burst termination. These neurons provided drive to both inhibitory interneurons and premotor neurons. Interactions between interneurons and premotor neurons form the basis of the pattern-forming network, appropriately phasing the motor output. It should be emphasized that feedback from the inhibitory interneurons onto the endogenous bursters, while not necessary for eupneic rhythmogenesis per se, nonetheless makes an important contribution to the eupneic rhythmogenesis by both controlling baseline membrane potential (which influences burst frequency) and facilitating burst termination. Increasingly elaborate hybrid pacemaker-type simulations are now being used. Although the voltage clamp data to support the use and magnitude of specific conductances are extremely sparse, these second-generation simulations benefit from recent work to characterize the voltage-dependent behavior of preBötC neurons. They also involve much larger numbers of neurons to account for heterogeneous cellular properties within given neuronal populations. As such, the output of these simulations is becoming increasingly realistic, but more important, they are now being used to formulate experimentally testable hypotheses about how rhythmicity in vitro is generated (Del Negro et al., 2001). With the third generation of models, we can look forward to simulation of (1) conductances based on voltage clamp data from preBötC neurons, (2) electrical synaptic connections, and (3) effects of neuronal modulation.

A different approach to using conductance-based modeling is to use more abstract components, which make the analysis of dynamic behavior more tractable. For example, Matsugu, Duffin, and Poon (1998) considering a minimal respiratory network consisting of two mutually inhibitory elements (representing I and E neurons) that were driven by different periodic inputs (representing “pacemaker” neurons). By varying the nature of the coupling between elements, the authors found that 1:1 entrainment to the “pacemaker” required a number of specific conditions. The simplified nature of this model, while allowing a rigorous analysis, does not allow an easy transfer of these predictions to practical experimental tests. However, an important aspect of this study is that it aimed specifically at evaluating the class of hybrid-pacemaker model in the context of respiratory rhythmogenesis.

Network Models

By far the most common feature of neuronal networks that generate rhythmic output is the presence of cells with mutually inhibitory synaptic connections. Such networks have an inherent tendency to generate rhythmic outputs without the need for cells with endogenous bursting properties. The network models for respiratory rhythm generation can be thought of as an elaboration on this basic theme (Figure 1B). These models have been constructed by analyzing the activity and phase of synaptic inputs of various types of respiratory neurons (see Richter et al., 1992). The data necessary to construct these models come from cross-correlation and spike-triggered averaging from single-unit recordings largely from in vivo preparations.

Data supporting inhibitory synaptic interactions in rhythmogenesis include the following:

1. Reduced Cl^- solutions caused cessation of eupneic bursts in an adult arterially perfused rodent preparation (Hayashi and Lipski, 1992).
2. GABA and glycine antagonists injected bilaterally into the preBötC of the in vivo cat abolished eupnea (Pierrefiche et al., 1998).

These data are consistent with a role for phasic inhibition in rhythmogenesis. However, these manipulations may have had direct effects on membrane potential and/or removed tonic inhibition, either of which could silence pacemaker neurons.

Paton and Richter (1995), using both in vivo and in vitro preparations, showed comparable results blocking inhibitory synaptic transmission in postnatal animals. However, in neonates, they found that blocking inhibitory synaptic transmission did not eliminate rhythmic motor output, in line with previous results from experiments on neonates (e.g., Ballanyi et al., 1999; Johnson et al., 1994). The authors conclude that inhibitory synaptic connections are necessary for respiratory rhythmogenesis, but only in postnatal (>15 days) animals. Others have found no dependence of rhythmicity on postnatal age (Ramirez in Ramirez and Richter, 1996). These dichotomous results might be explained by the fact that the preparations used by Paton and Richter included the pons. Some have argued that a late-developing pontine input negates the bursting properties of pacemaker neurons, making inhibitory connections in the ventral medulla necessary for eupneic rhythmogenesis (Pierrefiche et al., 1998).

These data have motivated a number of modeling efforts. Early network models took the view that each population of respiratory neurons could be considered as a homogenous group, often described with a few simple equations, with connections between populations as the underlying mechanism of rhythmogenesis (Lewis, 1995). More recent simulations have used individual conductance-based model neurons. In one study, by balancing the relative contributions of six different channels, two different types of neurons were modeled. The first type shows an “adapting” response to synaptic excitation (resembling Early I, PostI), while the second type exhibits a “ramping” response to release from inhibition (resembling I, Late I, E2). Modeling demonstrated that the ratio of high- and low-threshold Ca^{2+} channels was the principle factor in determining type. The greater the high-threshold Ca^{2+} conductance, the more likely the neuron would have an adaptive firing pattern. This prediction remains to be tested experimentally. Using these two types of “canonical” neurons, the authors then constructed a network consisting of seven cells representing seven classes of respiratory neurons—those described in the classification of Richter et al. (1992) plus an additional class of expiratory neuron. The model also includes inputs from pulmonary stretch receptors. The authors used various network configurations (based on

experimental data) to show how intrinsic membrane properties interact with synaptic (network) properties to control the firing patterns of individual neurons as well as the dynamics involved in changing between inspiratory and expiratory phases.

Discussion: A Hybrid-Pacemaker or a Network-Based Rhythm Generator?

We have outlined experimental data from two types of preparations that lead to very different predictions about the mechanisms of respiratory rhythmogenesis. In vivo data point to the importance of network interactions in generating the basic rhythm, whereas results from in vitro experiments indicate the importance of endogenous bursting neurons. Most recent modeling studies retain this polarity, concentrating on one or the other data set, resulting in models in which rhythmogenesis is either pacemaker driven or network driven. By providing a quantitative framework, these modeling studies have been invaluable in helping to understand the cellular basis of each of these mechanisms. In listing the assumptions they require to make the models, they also illustrate the need for more voltage clamp data from respiratory neurons.

An important use of modeling in the future will likely be to determine how to distinguish experimentally between pacemaker-driven and network-driven rhythmogenesis. Rybak, St. John, and Paton (2001) recently published a study demonstrating the possibility of using models in this way. Using a conductance-based pacemaker model as a starting point, they shifted the voltage dependence of $I_{Na(p)}$ by -9 mV (which they considered more physiological) and added a transient K^+ current (I_{KA}). These modifications did not affect the model neuron's ability to burst under the simulated conditions of elevated extracellular K^+ (9 mM) used in vitro. However, when $[K^+]$ was reduced toward levels closer to those in vivo (6 mM), the model neuron lost its ability to generate spontaneous bursts, owing to suppression of $I_{Na(p)}$ by I_{KA} . Furthermore, simulations suggested that bursting was disrupted by phasic synaptic inhibition. The authors suggest that rhythmogenesis in vitro represents a "switching" from a network-driven mechanism. They speculate that in vivo, the $I_{Na(p)}$ in conditional pacemakers is suppressed, but in vitro, the $I_{Na(p)}$ is released from suppression, owing to higher extracellular $[K^+]$ and the loss of synaptic inhibition. The authors tested this hypothesis in an artificially perfused preparation by blocking I_{KA} (using 4-AP), reducing inhibitory synaptic transmission (with a low dose of strychnine), and elevating extracellular K^+ (10 mM). The eupneic bursts were transformed to gasp-like events similar to those produced by in vitro preparations. While the details of Rybak et al.'s model are likely to be debated and the effects of their experimental manipulations should be scrutinized, the elegance of this study demonstrates the true potential of modeling and its growing importance in the study of respiratory rhythm generation.

Road Map: Motor Pattern Generators

Related Reading: Chains of Oscillators in Motor and Sensory Systems; Half-Center Oscillators Underlying Rhythmic Movements; Spinal Cord of Lamprey: Generation of Locomotor Patterns

References

- Ballanyi, K., Onimaru, H., and Homma, I., 1999, Respiratory network function in the isolated brainstem-spinal cord of newborn rats, *Prog. Neurobiol.*, 59:583–634. ♦
- Del Negro, C. A., Johnson, S. M., Butera, R. J., and Smith, J. C., 2001, Models of respiratory rhythm generation in the pre-Bötzinger complex: III. Experimental tests of model predictions, *J. Neurophysiol.*, 86:59–74.
- Gray, P. A., Janczewski, W. A., Mellen, N., McCrimmon, D. R., and Feldman, J. L., 2001, Normal breathing requires preBotzinger complex neurokinin-1 receptor-expressing neurons, *Nat. Neurosci.*, 4:927–930.
- Hayashi, F., and Lipski, J., 1992, The role of inhibitory amino acids in control of respiratory motor output in an arterially perfused rat, *Resp. Physiol.*, 89:47–63.
- Johnson, S. M., Smith, J. C., Funk, G. D., and Feldman, J. L., 1994, Pacemaker behavior of respiratory neurons in medullary slices from neonatal rat, *J. Neurophysiol.*, 72:2598–2608.
- Lewis, J. E., 1995, Respiratory rhythm generation, in *Handbook of Brain Theory and Neural Networks*, 1st ed. (M. A. Arbib, Ed.), Cambridge, MA: MIT Press, pp. 813–816.
- Matsugu, M., Duffin, J., and Poon, C.-S., 1998, Entrainment, instability, quasi-periodicity, and chaos in a compound neural oscillator, *J. Comput. Neurosci.*, 5:35–51.
- Paton, J. F., and Richter, D. W., 1995, Role of fast inhibitory synaptic mechanisms in respiratory rhythm generation in the maturing mouse, *J. Physiol.*, 484:505–521.
- Pierrefiche, O., Schwarzscher, S. W., Bischoff, A. M., and Richter, D. W., 1998, Blockade of synaptic inhibition within the pre-Bötzinger complex in the cat suppresses respiratory rhythm generation *in vivo*, *J. Physiol. (Lond.)*, 509:245–254.
- Ramirez, J. M., and Richter, D. W., 1996, The neuronal mechanisms of respiratory rhythm generation, *Curr. Opin. Neurobiol.*, 6:817–825. ♦
- Rekling, J. C., and Feldman, J. L., 1998, PreBotzinger complex and pacemaker neurons: Hypothesized site and kernel for respiratory rhythm generation, *Annu. Rev. Physiol.*, 60:385–405. ♦
- Richter, D. W., Ballanyi, K., and Schwarzscher, S., 1992, Mechanisms of respiratory rhythm generation, *Curr. Opin. Neurobiol.*, 2:788–793.
- Rybak, I. A., St. John, W. M., and Paton, J. F. R., 2001, Models of neuronal bursting behavior: Implications for *in vivo* versus *in vitro* respiratory rhythmogenesis, *Adv. Exp. Med. Biol.*, 499:159–164.
- Smith, J. C., Butera, R. J., Koshiya, N., Del Negro, C., Wilson, C. G., and Johnson, S. M., 2000, Respiratory rhythm generation in neonatal and adult mammals: The hybrid pacemaker-network model, *Respir. Physiol.*, 122:131–147. ♦
- St. John, W. M., 1996, Medullary regions for neurogenesis of gasping: Noeud vital or noeuds vitaux, *J. Appl. Physiol.*, 81:1865–1877. ♦
- St. Jacques, R., and St. John, W. M., 1999, Transient, reversible apnoea following ablation of the pre-Bötzinger complex in rats, *J. Physiol. (Lond.)*, 520:303–314.

Retina

Robert G. Smith

Introduction

At the most basic level, the retina transduces spatial and temporal variations in light intensity and transmits them to the brain. However, instead of directly coding intensity, the retina transforms visual signals in a multitude of ways to code properties of the visual world such as contrast, color, and motion. This article develops a

conceptual theory to explain why the retina codes visual signals and how the structure of the retina is related to its coding function.

The vertebrate retina reliably responds to light contrast as low as 1% (Shapley and Enroth-Cugell, 1984). Yet as the delicate visual signal is amplified in its passage through the retina, the biological limitations of neural processing add distortion and noise. The ease with which we see fine details in the presence of such biological

limitations suggests that one function of retinal circuitry is to maintain the signal's quality by removing redundant signals (Laughlin, 1994). This hypothesis predicts that much of the retina's signal coding and structural detail is derived from the need to optimally amplify the signal and eliminate noise.

Structure

Layers and Cell Classes

The retina is a thin (100–200 μm) tissue at the rear surface of the eye consisting of three layers of neurons and glial cells (Figure 1) (see Dowling, 1987; Sterling, 1997; Rodieck, 1998). Neurons in the *outer nuclear layer* (ONL) are exclusively photoreceptors. The *inner nuclear layer* (INL) (i.e., the middle layer) contains the cell bodies of horizontal cells (H), bipolar cells (B), and amacrine cells (A). Between these two layers lies the *outer plexiform layer* (OPL), in which bipolar and horizontal cells extend dendritic processes laterally to receive synaptic contacts from photoreceptors. The innermost cell layer, called the *ganglion cell layer* (GCL), contains cell bodies of ganglion cells and amacrine cells. Between the INL and GCL lies the *inner plexiform layer* (IPL), where bipolar, amacrine, and ganglion cells are synaptically connected. Ganglion cells send their output to the brain through axons that lie on the inner surface of the retina.

Cell Types: Specificity in Form and Function

Each class of neuron described above comprises several cell types, and overall the retina comprises several dozen (Sterling, 1997; Rodieck, 1998). A cell type is defined by a distinctive morphology, distribution, synaptic connection pattern, physiology, and/or immunocytochemical staining pattern (Rodieck, 1998). That distinct cell types exist suggests that each has a specific function. Although

the retina of one species may contain cell types not present in another, the same five retinal cell classes exist in all vertebrate species (Dowling, 1987; Masland, 1988; Sterling, 1997; Rodieck, 1998; Kandel, Schwartz, and Jessel, 2000). Therefore all vertebrates likely share similar neural circuit organization.

Receptive Fields and Connectivity

Neurons of each type are spaced in a regular array across the retina (see Figure 1), so the key to understanding retinal function is to identify the processing strategies of repeating functional circuits or modules (Sterling, 1997). To understand a retinal neuron's physiological function, investigators measure its *receptive field*, the region in space and time over which it responds to light. Receptive fields of retinal neurons consist of a sensitive circular region in visual space, called the *center*, and a larger but weaker antagonistic region concentric with the center, called the *surround* (Rodieck, 1998). These receptive fields are determined by intrinsic and pre-synaptic mechanisms. For example, a ganglion cell's receptive field shape and properties reflect its morphology and biophysical properties (Kandel et al., 2000), and also the receptive field properties of its presynaptic bipolar and amacrine cells, which in turn originate to some extent in the receptive field properties of photoreceptors and horizontal cells (Dowling, 1987; Sterling, 1997; Rodieck, 1998).

Although receptive field analysis is a powerful method for studying the function of a neural circuit (see Rodieck, 1998; Shapley and Enroth-Cugell, 1984), the origin of a receptive field in a circuit that includes several layers of neurons is difficult to grasp. The difficulty is to separate the effects of cell morphology, synaptic connectivity, and membrane channels on the receptive field (see, e.g., DIRECTIONAL SELECTIVITY). However, by computationally simulating these biophysical details based even on partial knowledge, it is possible to test specific hypotheses about neural circuit connectivity (Teeters and Arbib, 1991; Smith, 1995).

Functional Circuits

Photoreceptors and Adaptation

The outer segment (OS) of a vertebrate photoreceptor transduces light via a multistep biochemical cascade (Rodieck, 1998; Kandel et al., 2000) into an electrical signal that is conducted through the photoreceptor's axon to its terminal in the OPL. In response to a flash of light, ion channels close, hyperpolarizing the photoreceptor proportionately over a limited range of stimulus intensity. The advantage of this coding function is that a photoreceptor responds well to low-contrast signals common in the visual world. The disadvantage is that outside this limited range the photoreceptor responds poorly. At lower intensities, the photoreceptor's transduction gain (i.e., proportion of change in its output signal to a change in light) tends to be insufficient, and at higher intensities the photoreceptor's response tends to saturate. To solve such saturation problems, the photoreceptor adjusts its gain in a process called *adaptation*, which in some species can modulate transduction gain by up to 4 log units.

The two classes of photoreceptors, rods and cones (Rodieck, 1998), differ in that rods are sensitive to single photons and are bleached by daylight, but cones are less sensitive and can regenerate their pigment in daylight (photopic intensity range). At twilight (the mesopic intensity range), rods are coupled via gap junctions to neighboring cones, causing the rod signal to pass directly into cones, where it is carried by the lower-gain cone pathway (Rodieck, 1998). At night (scotopic intensity range), a special *rod bipolar* pathway (RB in Figure 1) carries quantal *single-photon* signals, removes dark noise, and adapts over an extra 3 log units

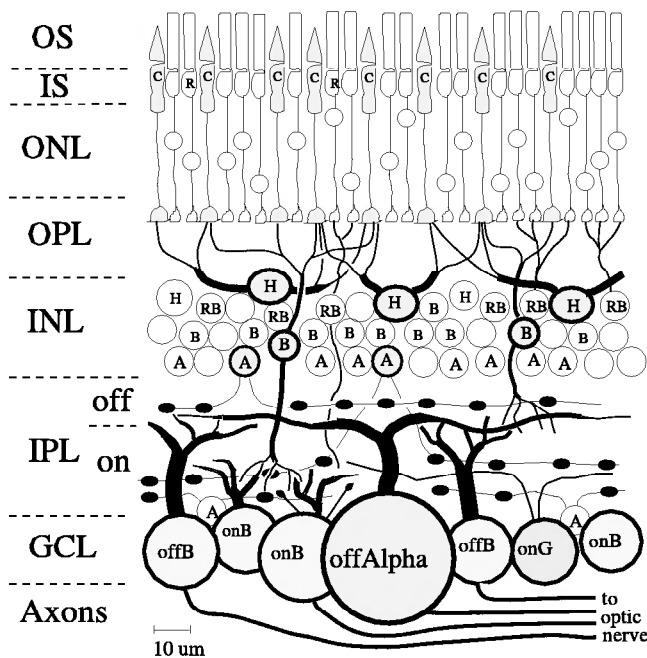


Figure 1. Structure of the retina, showing the outer segments (OS), inner segments (IS), outer nuclear layer (ONL), outer plexiform layer (OPL), inner nuclear layer (INL), inner plexiform layer (IPL), ganglion cell layer (GCL), horizontal cells (H), bipolar cells (B), amacrine (A), and rod bipolar (RB) cells.

of intensity (Sterling, 1997; Rodieck, 1998; Smith and van Rossum, 1998).

Outer Plexiform Layer

The axon terminal of a cone transmits its signal to bipolar cells with a chemical synapse, increasing signal gain at the cost of extra noise and a reduction in the intensity range over which the bipolar cell can respond (Laughlin, 1994). To reduce the tendency of the cone signal to saturate its synapse, the OPL filters the signal (Laughlin, 1994). The filter consists of two components, a spatial low-pass filter constructed from lateral electrical connections (gap junctions) between cones, and a spatiotemporal high-pass filter constructed by horizontal cells. Cone coupling removes uncorrelated noise from the cone's response, and consequently causes cone synaptic release to be more correlated. Although the coupling also causes "neural blur," it is useful to provide an anti-aliasing filter for the next stage of processing in the IPL.

The OPL's high-pass filter is constructed by subtracting a local average from the cone. Horizontal cells, also coupled laterally by gap junctions, sum inputs from many cone terminals and provide negative feedback to each via a feedback synapse. The negative feedback mechanism in some cases is a GABA-ergic synapse (Dowling, 1987; Sterling, 1997), but has also recently been postulated to be a form of electrical feedback. The synaptic structure that performs this function, called a *triad*, has both feedforward and feedback contacts, so it is termed *reciprocal* (Dowling, 1987; Rodieck, 1998). This type of coding has been termed *predictive* (Laughlin, 1994) because the ideal signal to subtract would be a *local average* of signals from neighboring cones (Smith, 1995).

Synaptic Function and Noise: Signal-Processing Mechanisms

The glutamatergic synapse that transmits a cone's signal to bipolar and horizontal cells adds noise originating in the random fluctuation of synaptic vesicle release rate (Sterling, 1997; Kandel et al., 2000). To reduce the amount of noise relative to the signal, vesicles are released at a high rate by a ribbon that functions as a docking site and reservoir for vesicles (Sterling, 1997; Rodieck, 1998; Kandel et al., 2000). The synapse that relays rod signals to rod-bipolar cells in starlight has a special challenge because noise generated by the rod's transduction cascade and synapse would swamp the tiny single-photon signal. A computer simulation suggested the solution (recently verified by *in vitro* recordings): a nonlinear threshold in the postsynaptic second-messenger system removes the noise (Smith and van Rossum, 1998).

IPL: Bipolar and Amacrine Circuits

The retina contains about 10 types of bipolar cell and more than 20 types of amacrine cell (Kolb, Nelson, and Mariana, 1981; Masland, 1988; Sterling, 1997; Rodieck, 1998). Bipolar cell dendrites arborize in the OPL to receive multiple synaptic contacts from photoreceptors, and their axons terminate in the IPL. Amacrine cells extend their dendrites laterally in the IPL to contact bipolar, amacrine, and ganglion cells.

Bipolar cells respond as photoreceptors do with a voltage proportional to light intensity, but their response range is narrower and they adapt over a wider range of stimuli. Adaptation occurs at the dendritic tip from changes in gain at a second-messenger biochemical cascade, at the membrane by voltage-gated ion channels, or at the axon, where gain of output synapses is regulated in several ways by feedback. Bipolar cells contact ganglion cells with glutamatergic ribbon synapses to allow high release rates and reduce noise. A bipolar cell may contact several ganglion cell types, each

with a different characteristic number of synapses, which implies a specific coding of the bipolar signal (Teeters and Arbib, 1991; Sterling, 1997).

Function of Amacrine Cells

Amacrine cells are a diverse group in both morphology (Kolb et al., 1981; Rodieck, 1998) and neurochemistry (Masland, 1988). Many have a large (0.5–2 mm) but sparse dendritic field with very fine dendritic processes (0.2 μ m diameter) that stretch between small swellings, called *varicosities*, where synaptic connections are made (Kolb et al., 1981; Dowling, 1987). Most amacrine cells contain voltage-gated Na⁺ channels and fire action potentials, which allows them to transmit signals laterally over the extent of their dendritic field (Masland, 1988). Amacrine cells are generally either GABAergic (Rodieck, 1998; Kandel et al., 2000) or glycinergic, which implies that they perform subtractive or shunting control functions. Some, such as the cholinergic "starburst" amacrine, are involved in temporal processing and respond transiently to light (Masland, 1988). Amacrine circuitry is thought to be responsible for accentuating the surround, directional selectivity in ganglion cells (see DIRECTIONAL SELECTIVITY), excitatory transient and peripheral effects, and several types of gain control (Shapley and Enroth-Cugell, 1984; Dowling, 1987; Rodieck, 1998).

Amacrine cells receive synaptic contacts from bipolar cells at a *dyad*, where a bipolar ribbon synapse contacts two postsynaptic neurons, usually ganglion and amacrine cells (Rodieck, 1998). The similarity between the synaptic dyad in the IPL and the triad in the OPL is striking (Rodieck, 1998). Both contain synaptic ribbons, and both include reciprocal feedback from a lateral neuron. The reason may be the identical problem of noise. The reciprocal feedback from an amacrine varicosity to its presynaptic bipolar cell can process the signal, reducing its dynamic range before transmission to ganglion cells (Masland, 1988; Dowling, 1987; Rodieck, 1998).

Gap Junction Coupling in Amacrine and Bipolar Cells

Amacrine and bipolar cells, like many types of neuron in the brain, are widely coupled by gap junctions to their neighbors. Bipolar cell coupling, like cone coupling, correlates neighboring cells' signals to enhance synchronous vesicle release. Since many amacrine cells fire action potentials, one possibility is that gap junctions allow them to synchronize their firing. But their diversity emphasizes the complexity of retinal circuitry (Masland, 1988; Rodieck, 1998). The AII amacrine cell is small-field and carries rod signals from the rod bipolar to cone bipolars at night (Kolb et al., 1981; Rodieck, 1998). To grasp the function of the AII has been a special challenge because it is coupled by gap junctions to its AII and bipolar cell neighbors, and these two types of electrical coupling are controlled by independently modulated second-messenger systems. The AII also contains voltage-gated Na⁺ channels and generates action-potential-like transients. These specialized biophysical properties elegantly solve a signal-processing challenge: in starlight, the AII collects single-photon signals from an array of presynaptic rod bipolars, but synaptic noise is collected even when photons are rare. The AII's strategy, therefore, is to reduce noise by electrical coupling with neighbors, and to nonlinearly amplify the single-photon signal with voltage-gated channels (Smith and Vardi, 1995; Sterling, 1997), removing noise and reshaping the signal before passing it on.

Diversity of Coding

Ganglion Cells

Ganglion cells have exquisite sensitivity to low contrast stimuli over a wide range of light intensity (Kandel et al., 2000). They are

specialized into diverse types that code different properties of the visual world (see **DIRECTIONAL SELECTIVITY**; see also Maturana, Lettvin, and McCulloch, 1960; Kolb et al., 1981; Rodieck, 1998). Some give a tonic response to stationary stimuli (e.g., the X or W cells of cat retina), and others give a more phasic response to signal the presence of flashing or moving stimuli (e.g., the Y cell of cat retina). Many species (lower vertebrates but also mammals) possess ganglion cells with more complex receptive fields; for example, they respond only to small or large moving objects (Maturana et al., 1960; Teeters and Arbib, 1991). In many species color-opponent ganglion cells provide excellent color vision (Dowling, 1987; Rodieck, 1998).

Coding by the Spike Generator

To transmit a signal to the brain, the ganglion cell codes its intracellular voltage (the generator potential) as the firing rate of action potentials along its axon (Kandel et al., 2000). Like synaptic coding, this process is limited by noise and dynamic range, so optimal coding of information is at a premium. The ganglion cell's spike generator, consisting of voltage- and ion-gated channels, traditionally thought to be located in the axon hillock and soma, is responsible for the coding properties. However, ganglion cells have voltage-gated channels in their dendritic membrane, and recently it was shown by simulation that the dendritic tree must contain these channels at sufficient densities to conduct action potentials, for without them the spike rate becomes too high (Fohlmeister and Miller, 1997). Dendritic morphology and slowly activated K^+ channels (Kandel et al., 2000) are also involved in shaping the ganglion cell's response. Simulations have also shown that noise in the spike generator causes variability in the spike rate, and that a significant portion of the information available in the ganglion cells' generator potential is lost in the process.

IPL: Specific Circuits in Sublayers

One problem faced by the spike generator is inherent: it cannot respond well to hyperpolarizations below a certain threshold, and just above threshold, spiking is noisy. To cope with this problem, the retina contains two subclasses of ganglion cell, called *on* and *off*, that respond with opposite polarity to a light stimulus. The *on* cell increases its firing rate to a flash of light and the *off* cell reduces its firing rate. Responses of *on*- and *off*-ganglion cells in many species are symmetric, which allows the retina to code bright and dark objects without much distortion or noise.

To supply *on*- and *off*-ganglion cells with appropriate signals, the IPL is organized into *on*- and *off*-layers (sublaminae). Two bipolar cell subclasses, *on*, and *off*, respond oppositely to glutamate released by cones. Bipolar and amacrine cell types are divided roughly equally between the two layers, although some arborize in both. The *on*- and *off*-layers are in turn organized into specific sublayers defined by microcircuits comprising bipolar, amacrine, and ganglion cells, each generating a specific spatiotemporal code (Sterling, 1997).

On-bipolar dendrites contain *metabotropic* receptors that, when bound by glutamate released by a photoreceptor, signal a cytoplasmic second messenger to turn off the synapse's ionic channels (Dowling, 1987; Sterling, 1997; Rodieck, 1998). Thus, an *on*-bipolar depolarizes when the photoreceptor decreases its glutamate release (i.e., in response to light). An *off*-bipolar contains ionotropic glutamate receptors that directly open an ion channel and hyperpolarize to light. Each *off*-bipolar type contains glutamate receptors with different kinetic parameters, which is the first step in generating a specific temporal code. Some bipolar cells code stimulus velocity, direction, or color (Rodieck, 1998; Haverkamp, Moeckel, and Ammermuller, 1999). These specializations increase

signal fidelity, which is an advantage for a visual signal that is destined to pass through a noisy channel to the brain.

Discussion

There are several explanations for the diversity of retinal circuitry. By discarding part of the information it receives, a neuron specializes in coding specific properties of the signal, e.g., contrast, motion, bright, dark, colored light flashes, and so forth. The exact details of the coding scheme are probably related to the ecological niche occupied by the organism. Rod signals, because of their quantal nature, are qualitatively different from cone signals, so there is an advantage to having a separate rod pathway. Such specialization in coding increases the signal/noise ratio and makes better use of the limited dynamic range of neurons, synapses, and the spike train in the ganglion cell axon (Laughlin, 1994). Specialization in coding also simplifies the task of brain circuitry in visual segmentation, which may imply a function for retinal circuit structure but involves later visual processes as well (see **VISUAL SCENE SEGMENTATION**).

Local Processing in Retinal Circuits

The receptive fields of many retinal neurons, and particularly of ganglion cells, share important properties, among them a center-surround organization, high sensitivity to contrast, and wide-ranging adaptation. To the extent that each retinal circuit amplifies the signal, it adapts to reduce the signal's dynamic range, which implies that the retina's high sensitivity is achieved at the cost of complexity. For example, the net effect of the OPL circuit is to create for the photoreceptor a receptive field with a broad center region and a wide antagonistic surround (Sterling, 1997; Rodieck 1998; Kandel et al., 2000) that adapts temporally and spatially. By removing information about absolute light intensity, the OPL circuit transmits what is left, i.e., information about contrast. In turn, the IPL circuit removes more information about light intensity and contrast, shaping the signal in time to code transients, and accentuating the spatial center-surround receptive field in bipolar and amacrine cells (Dowling, 1987; Rodieck, 1998). This process further regulates the visual signal's gain to improve discrimination of low-contrast objects from noise and to prevent saturation at high contrast (Shapley and Enroth-Cugell, 1984). The result of these operations is that retinal receptive fields change with background intensity to maximize information transfer (Laughlin, 1994), and the consequence of this processing is the familiar center and surround of the ganglion cell. Thus, it appears that circuits along the retinal pathway all contribute to the ganglion cell's receptive field properties for a similar reason: to prevent noise or saturation from degrading the signal (Laughlin, 1994).

The well-known antagonistic center-surround and adaptation properties of the ganglion cell receptive field, therefore, seem driven by the goal of preserving signal quality. To accomplish this, the circuitry of both OPL and IPL increase the receptive field's lateral extent. But the need for high visual acuity mandates that OPL and IPL circuits not extend laterally too far. Thus, the retina is shaped to compensate for biological limitations by a compromise between spatial acuity and accuracy of coding.

Testing the Theory

Although knowledge of the biophysical components of retinal circuitry and its receptive fields is progressing rapidly, such knowledge does not guarantee a useful theory. For example, whole-cell patch recordings allow the biophysical properties and visual responses of bipolar and amacrine cells presynaptic to a ganglion cell to be measured, and these presynaptic responses contribute to the

ganglion cell's receptive field. Yet such knowledge alone cannot answer the question of function in design: what function the individual components add to the circuit, and therefore why they exist. The answer can only be derived from synthetic models that integrate details of the retina's neural circuitry with the noise and dynamic range limitations inherent to neurobiology.

Computational modeling promises to help find the answers (Teeters and Arbib, 1991; Smith, 1995; Smith and Vardi, 1995; Fohlmeister and Miller, 1997; Haverkamp et al., 1999). Once the basic signal flow and function in a retinal circuit have been established, simulations can help determine overall strategies, and with information theory can find what biological limitations are most serious to the circuit (Laughlin, 1994). The effect of noise on the retina's performance can be tested by simulating noise from all the sources in the signal pathway, and comparing the resulting signal/noise ratios as a measure of signal quality.

Road Map: Vision

Related Reading: Color Perception; Directional Selectivity; Motion Perception: Elementary Mechanisms; Visuomotor Coordination in Frog and Toad

References

- Dowling, J. E., 1987, *The Retina: An Approachable Part of the Brain*, Cambridge, MA: Harvard University Press. ♦
- Fohlmeister, J. F., and Miller, R. F., 1997, Mechanisms by which cell geometry controls repetitive impulse firing in ganglion cells, *J. Neurophysiol.*, 78:1948–1964.
- Kolb, H., Nelson, R., and Mariani, A., 1981, Amacrine cells, bipolar cells, and ganglion cells of the cat retina: A Golgi study, *Vision Res.*, 21:1081–1114.
- Haverkamp, S., Moeckel, W., and Ammermuller, J., 1999, Different types of synapses with different spectral types of cones underlie color opponency in a bipolar cell of the turtle retina, *Vis. Neurosci.*, 16:801–809, 1999.
- Kandel, E. R., Schwartz, J. H., and Jessel, T. M., 2000, *Principles of Neural Science*, 4th ed., New York: McGraw-Hill. ♦
- Laughlin, S. B., 1994, Matching coding, circuits, cells, and molecules to signals: General synaptic principles of retinal design in the fly's eye, *Prog. Retinal Eye Res.*, 13:165–196. ♦
- Masland, R. H., 1988, Amacrine cells, *Trends Neurosci.*, 11:405–410. ♦
- Maturana, H. R., Lettvin, J. Y., and McCulloch, W. S., 1960, Anatomy and physiology of vision in the frog (*Rana pipiens*), *J. Gen. Physiol.*, 43:129–175.
- Rodieck, R. W., 1998, *The First Steps in Seeing*, Sunderland, MA: Sinauer. ♦
- Shapley, R. M., and Enroth-Cugell, C., 1984, Visual adaptation and retinal gain controls, *Progr. Retinal Res.*, 3:263–346. ♦
- Smith, R. G., 1995, Simulation of an anatomically defined local circuit: The cone-horizontal cell network in cat retina, *Vis. Neurosci.*, 12:545–561.
- Smith, R. G., and van Rossum, M. C. W., 1998, Noise removal at the rod synapse of mammalian retina, *Vis. Neurosci.*, 15:809–821.
- Smith, R. G., and Vardi, N., 1995, Simulation of the AII amacrine cell of mammalian retina: Functional consequences of electrical coupling and regenerative membrane properties, *Vis. Neurosci.*, 12:851–860.
- Sterling, P., 1997, Retina, in *The Synaptic Organization of the Brain*, 4th ed. (G. M. Shepherd, Ed.), New York: Oxford University Press. ♦
- Teeters, J. L., and Arbib, M. A., 1991, A model of anuran retina relating interneurons to ganglion cell responses, *Biol. Cybern.*, 64:197–207.

Robot Arm Control

Carme Torras

Introduction

A robot is a multifunctional and reprogrammable mechanism able to move in a given environment. Three broad classes of robots can be distinguished on the basis of their mobility. *Robot arms* have a fixed base, and their mobility comes from their articulated structure; thus, they operate on a bounded three-dimensional (3D) subspace. *Robot vehicles* move on two-dimensional (2D) surfaces by using wheels or other similar continuous traction elements (see ROBOT NAVIGATION). *Walking robots* are designed to move through rough terrains by using articulated legs. Of course, mixed possibilities also exist, such as robot arms mounted on wheeled vehicles. This article is devoted to robot arms, or manipulators, which we will call simply robots.

Each robot is endowed with a controller that commands its mechanical structure to perform the desired tasks. Controllers are usually *hierarchically* structured from the lowest level of servomotors to the highest levels of trajectory generation and task supervision. The activity taking place at all these levels is conceptually the same: an actual motion (of a single joint, the end-effector, or the entire robot) is made to follow as closely as possible a commanded motion through the use of feedback. The difference lies in the coordinate systems used at each level.

At least four coordinate spaces can be distinguished: the task space (used to specify tasks, possibly in terms of sensor readings), the workspace (six-dimensional Cartesian coordinates defining a position and orientation of the end-effector), the joint space (intrinsic coordinates determining a robot configuration), and the actuator space (in which actual motions are commanded). Because

robot control entails transforming a specification in task space into actuator commands, it critically depends on accurate mappings between the various coordinate spaces.

Neural networks have been used to approximate these mappings when they are difficult or impossible to derive analytically (as in the case of flexible or redundant robots, or in tasks entailing sensorimotor coordination) and when, because of environmental changes or robot wear-and-tear, the mappings vary in time and the controller needs to adapt on-line to these variations. This article discusses four such mappings and the neural models used to implement them adaptively.

Neural Adaptivity in Robot Control

A robot, when moving, can be thought of as realizing a mapping from actuator space to joint space, and to the workspace and task space as well. These mappings are referred to as *forward mappings* and are parameterized by the current state of the robot. In the same general terms, a controller can be viewed as implementing *inverse mappings* in that, given the current state and a desired output (sensory pattern or robot pose), the controller has to generate the appropriate commands to attain that output.

Control strategies often rely on models of the various mappings we have delineated. Forward models are used to provide fast internal feedback in order to prevent instabilities in the control loop. Inverse models lie at the core of feedforward control. The learning of such models by means of neural networks is described in SENSORIMOTOR LEARNING (q.v.), and their use inside robot controllers is discussed in ROBOT LEARNING (q.v.).

For the purposes of the present discussion, let us mention that inverse models can be acquired under four schemes, namely *direct inverse modeling*, *feedback-error learning*, *distal supervised learning*, and *reinforcement learning* (see SENSORIMOTOR LEARNING for details). These schemes can be applied under both supervised and unsupervised (or self-supervised) training modes and through the use of correlational, reinforcement, or error-minimization adaptation procedures (Torras, 1995). This distinction between adaptation procedures is made on the basis of the type of problem information they use (Figure 1).

Correlational procedures use no problem information, and their goal is to carry out feature discovery or clustering. In a robot control setting, these procedures are often used to represent a given state space in a compact and topology-preserving manner. Two applications to be described later rely on representations of this type for the robot workspace (see discussion under “Inverse Kinematics”) and the space of joint positions, velocities, and accelerations (see “Inverse Dynamics”). The correlational procedures most widely used for robot control are SELF-ORGANIZING FEATURE MAPS (q.v.) and ADAPTIVE RESONANCE THEORY (ART) (q.v.).

Error-minimization procedures require complete target information in the form of input-output pairs. The goal of using such procedures is to build a mapping from inputs to outputs that generalizes properly. These procedures are the most widely used in applications. Among others, the LMS rule, backpropagation (see PERCEPTRONS, ADALINES, AND BACKPROPAGATION), locally weighted projection regression, and conjugate gradient optimization have been applied to robot control.

Reinforcement-based procedures lie between both extremes. They make use only of a reward/penalty signal to build a mapping that maximizes reward (see REINFORCEMENT LEARNING). These procedures have been applied for learning sensorimotor maps.

Inverse Kinematics

Inverse kinematics mapping provides joint coordinates as a function of the position and orientation of the robot end-effector, thus

relating the workspace to the joint space. The use of neural networks to learn this mapping is of particular interest when a precise model of some joints is lacking or when, because of the conditions of operation of the robot (in space, underwater, etc.), it is hardly possible to recalibrate it.

Feedforward networks using backpropagation have been extensively tested in this context, under both the *direct inverse modeling* and the *distal supervised learning* approaches (Jordan and Rumelhart, 1992), leading to the conclusion that a coarse mapping can be obtained quickly but an accurate representation of the true mapping often is not feasible or is extremely difficult. The reason for this seems to be the *global* effect that every connection weight has on the final approximation obtained (Kröse and van der Smagt, 1993).

A way to avoid this global effect is to use local representations, so that every part of the network is responsible for a small subspace of the total input space. One such representation is the 3D self-organizing feature map (SOM) used by Ritter, Martinetz, and Schulten (1992) to encode the robot workspace. This is combined with the LMS rule to learn the inverse kinematics of a robot arm with 3 degrees of freedom (dof) under a *direct inverse modeling* approach. The inputs to each neuron are the coordinates of the desired end-effector position, and the outputs (after correct learning) are the joint angles and the Jacobian corresponding to that position. Thus, this model provides a discrete encoding of the inverse kinematics mapping augmented by a linear approximation at each sample point that permits interpolating joint angles with higher precision. The network has been shown to self-organize into a reasonable representation of the workspace in about 30,000 learning cycles. This should be taken as an experimental demonstration of the powerful learning capabilities of this model, because the conditions in which it was made to operate are the worst possible ones: no a priori knowledge of the robot kinematics, random weight initialization, and random sampling of the workspace during training.

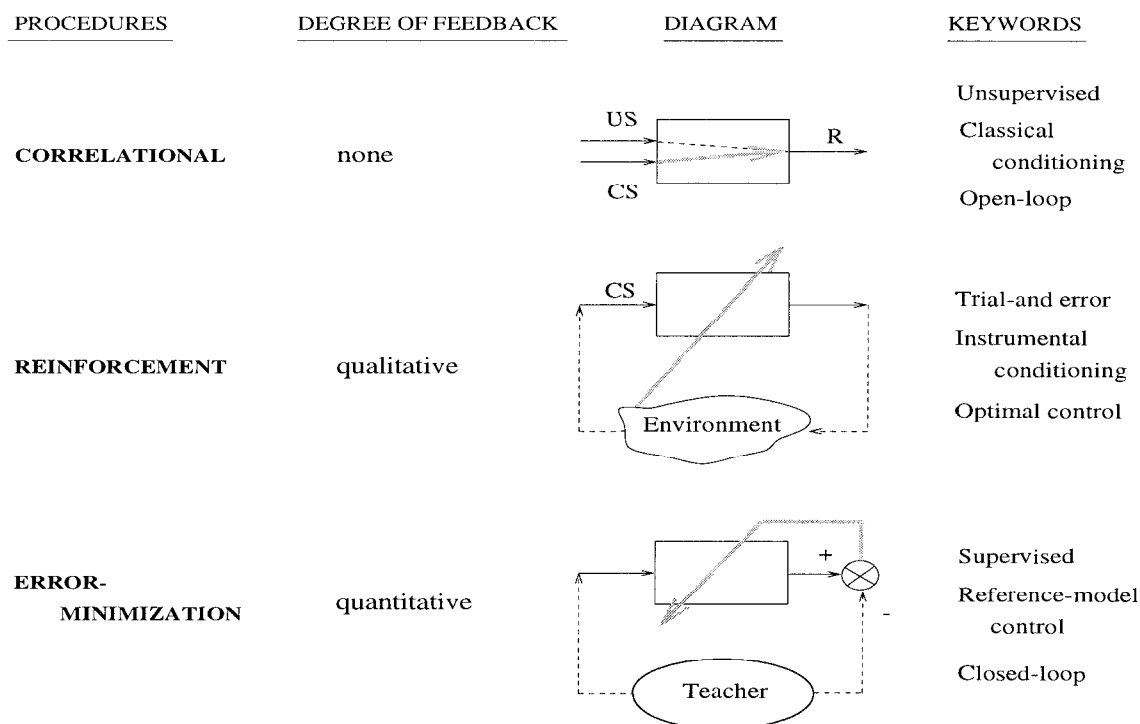


Figure 1. Procedures for neural adaptivity. See Torras (1995) for a detailed explanation.

This basic model has been extended in three directions to cope with higher-dof robots. First, a hierarchical version, consisting of a 3D SOM whose nodes each have associated a 2D SOM, was applied to a 5-dof robot. The 3D net encodes the workspace as before, while each 2D subnet approximates the end-effector orientation space at the corresponding position (Ritter et al., 1992).

Ruiz de Angulo and Torras (1997) have adapted this hierarchical model to suit a practical setting. Thus, instead of learning the kinematics from scratch, only the deviations from the nominal kinematics embedded in the original robot controller are learned. This, together with informed initialization and sampling, as well as several modifications in the learning algorithm aimed at improving the cooperation between neurons, led to a speed-up of two orders of magnitude with respect to the original model. Thus, when applied to the self-calibration of a 6-dof robot installed in a space station mockup, 95% of the decalibration was corrected with the first 25 movements, with the percentage rising to 98% after 100 movements. Moreover, other desirable features in stand-alone applications, such as parameter stability, are guaranteed.

The third extension relies on the generalization of SOMs to parameterized SOMs (PSOMs). The idea is to turn the discrete representation into a continuous one by associating a basis function with each neuron, so that a parameterized mapping manifold is obtained. Moreover, PSOMs make no distinction between inputs and outputs, thus encoding bidirectional mappings. Compared with the SOM, the PSOM considerably reduces the number of training samples required to attain a given precision (Walter and Ritter, 1996), allowing the learning of the full inverse kinematics of a 6-dof robot with less than 800 movements.

Recently, the development of humanoid robots has heightened interest in learning inverse kinematics. Because of the many dofs involved, the aim is no longer learning the mapping for the whole workspace but is focused on a specific trajectory. Following the trend of using localized representations, D'Souza, Vijayakumar, and Schaal (2001) have applied a supervised algorithm, locally weighted projection regression, in this context, with promising results (see ROBOT LEARNING).

Inverse Dynamics

When the robot dynamics needs to be taken into account, the control learning problem becomes more involved. An inverse dynamics mapping relating end-effector accelerations to the required joint forces and torques should be considered now.

Because the cerebellum is involved in the production and learning of smooth movements, several cerebellar models have been proposed and applied to control robot arms. The pioneer such model was the Cerebellar Model Articulation Controller (CMAC), developed by Albus in 1975 (see CEREbellum AND MOTOR CONTROL for related material), but today the debate is still open as to what model best captures the functionality of the cerebellum and whether any such model can constitute a practical option to control robots (van der Smagt and Bullock, 1997). A point of agreement is that the cerebellum constructs an inverse dynamics model as it learns. Thus, cerebellar models have been used for this purpose inside robot controllers.

Miller et al. (1990) combined the table look-up facilities provided by CMAC with an error-correction scheme similar to the LMS rule to accomplish the dynamic control of a 5-dof robot. The idea underlying this combination is similar to that of enlarging SOMs with the LMS rule, as described in the preceding section. Here, CMAC is used to represent the state space in a compact and localized manner, as SOMs were used to cover the robot workspace in the preceding section. To teach the robot to follow a given trajectory, successive points along it are supplied to both the neural network and a fixed-gain controller, and then their responses are

added up to command the robot. Therefore, the neural network acts as a feedforward component. After each cycle, the actual command given to the robot and its current state are used as an input-output pair to train the neural network, thus following a *direct inverse modeling* approach. As learning progresses, the CMAC network approximates the inverse dynamics mapping, and consequently, the effect of the fixed-gain controller tends to zero. The network converges to a low error (between one and two position encoder units) within ten trials, provided enough weight vectors are used.

The same trajectory learning task as described above was tackled by Miyamoto et al. (1988) using a *feedback-error learning* approach. They used directly as error signal the output of the feedback controller, which can be interpreted as a local linearization of the inverse dynamics mapping if the learning rate is sufficiently small. This error measure is less accurate than that used by Miller et al. (1990) but has the advantage of being directly available in the control loop, thus avoiding the computation of the current state of the robot required in the direct inverse modeling approach. The authors report that, after training the robot to follow a trajectory lasting 6 seconds for 300 trials, the average feedback torque decreased from a few hundred to just a few units, demonstrating that the neural network had taken over control from the fixed-gain controller. Moreover, the mean square error in the joint angles decreased steadily by 1.5 orders of magnitude.

Force-Motor Mapping

For tasks entailing the achievement of a goal using sensory feedback, even programming in task-space coordinates can be very complex. An example is the insertion of components with small clearance, since devising a detailed force-control strategy that performs correctly in all possible situations, and subject to real-world conditions of uncertainty and noise, is extremely difficult. What is needed to accomplish this type of task is an appropriate *sensory-motor mapping* that relates sensory patterns to actuator commands. A relay through the intermediate workspace and joint space representations may or may not be required (see LIMB GEOMETRY, NEURAL CONTROL).

Gullapalli, Barto, and Grupen (1994) have used an associative reinforcement learning system to learn active compliant control for peg-in-hole insertion using a 6-dof robot (see REINFORCEMENT LEARNING IN MOTOR CONTROL). The system takes the sensed peg positions and forces, as well as the previous position command, as inputs, and produces a new position command as output. Thus, 18 real values are entered into a network with two hidden layers of backpropagation units, and six real values are produced by its output layer of stochastic reinforcement-learning units. The reinforcement signal depends on the discrepancy between the sensed and the desired position of the peg, with a penalty term being activated whenever the sensed forces on the peg exceed a preset maximum. The training runs start with the peg at a random position and orientation with respect to the hole, and end when either the peg is successfully inserted or 100 time steps have elapsed. Experimental results show that after 150 trials, the robot is consistently able to complete the insertion. Moreover, the time to insertion decreases continuously from 100 to 45 time steps over the subsequent 500 training runs.

Visuomotor Mappings

Depending on the task to be performed and the camera-robot arrangement, visuomotor mappings take different forms. Thus, in eye-hand coordination, where cameras external to the robot are used to monitor the pose (position and orientation) of its end-effector, a mapping from the camera coordinates of a desired end-effector pose to the joint angles that permit attaining that pose is

Table 1. Neuroadaptive Procedures Used to Approximate Several Robot Mappings

Mapping	Correlational + Error Minimization	Error Minimization	Reinforcement Learning
Inverse kinematics	SOM + LMS (Ritter et al., 1992; Ruiz de Angulo and Torras, 1997) PSOM + LMS (Walter and Ritter, 1996)	BP (Jordan and Rumelhart, 1992; Kröse and van der Smagt, 1993) LWPR (D'Souza et al., 2001)	
Inverse dynamics	CMAC + LMS (Miller et al., 1990)	LMS (Miyamoto et al., 1988)	
Force-motor			RL (Gullapalli et al., 1992)
Visuomotor	SOM + LMS (Ritter et al., 1992)	BP (Wells et al., 1996) CG (Schram et al., 1996)	

Abbreviations: SOM, self-organizing feature map; PSOM, parameterized self-organizing feature map; LMS, least-mean-square algorithm; BP, backpropagation; LWPR, locally weighted projection regression; CMAC, Cerebellar Model Articulation Controller; RL, reinforcement learning; CG, conjugate gradient learning algorithm.

sought. This mapping is closely related to the inverse kinematics one, especially if the camera coordinates of selected points in the end-effector uniquely characterize its pose. Therefore, the same models used to learn inverse kinematics have been applied in this context (Ritter et al., 1992).

A camera mounted on a robot arm is used in tasks such as visual positioning and object tracking. The goal of these tasks is to move the camera so that the image captured matches a given reference pattern. The target is thus no longer a position in space but a desired image pattern, and the desired visuomotor mapping needs to relate offsets with respect to that pattern with appropriate movements to cancel them. In visual positioning, the scene is assumed to be static and the main issue is to attain high precision. Applications include inspection and grasping of parts that cannot be precisely placed (e.g., in underwater or space settings). The aim of object tracking is to maintain a moving object within the field of view, speed being the critical parameter here instead of precision.

The classical way of tackling these tasks consists in defining a set of image features and then deriving an interaction matrix relating 2D shifts of these features in the image to 3D movements of the camera (Samson, LeBorgne, and Espiau, 1990). Note that the visuomotor mapping can be implemented with or without a relay through the workspace, depending on how the movements of the camera are commanded.

In the case of visual positioning, Wells, Venaille, and Torras (1996) have used backpropagation to learn the interaction matrix. The training procedure consists in moving the camera from the reference position to random positions and then using the displacement in image features together with the motion performed as input-output pairs. The system thus follows a *direct inverse modeling* approach. In operation, the robot is commanded to execute the inverse of the motion that the network has associated with the given input. The key option in this work is the use of global image descriptors, which permits avoiding the costly matching of local geometric features in the current and reference images. By using a statistical measure of variable interdependence (the mutual information criterion), sets of global descriptors as variant as possible with each robot dof are selected from a battery of features, including geometric moments, eigenvectors, pose-image covariance vectors, and local feature analysis vectors (Wells and Torras, 1998). The results obtained with a 6-dof robot show that after 10,000 learning cycles, translation and rotation errors are less than 2 mm and 0.1 degrees, respectively.

Concerning object tracking, Schram et al. (1996) have used a feedforward network together with a conjugate gradient learning algorithm to make a camera track a cart moving arbitrarily on a table. A visuomotor mapping relating the current and past visual coordinates of the cart with joint displacements is built on-line as the robot moves. Only 2 robot dofs need to be controlled, and thus the network has two outputs, while several numbers of inputs have been tried. The tracking performance improves as more previous

positions of both the cart and the robot are used, attaining an average lag of only 8 mm in the case of seven inputs.

Discussion

After surveying several robot neurocontrol applications entailing the learning of various mappings (Table 1), we have extracted some guidelines.

In the case of *mappings that can be easily sampled*, it seems sufficient to apply a direct inverse modelling approach combined with an error-minimization learning procedure. Some simple inverse kinematics mappings and visuomotor mappings used for visual positioning have been learned in this way. If *the input space is complex*, then many researchers have resorted to a combination of correlational rules for the efficient coding of that space, with error-minimization procedures to build the appropriate association with the outputs. The use of self-organizing feature maps to encode the robot workspace or the sensor space, as well as the application of CMAC to the coding of the robot dynamics state space, fall into this category. When *the task is specified as a goal to be attained* using sensory feedback, without making explicit the movements necessary to attain it, the only possibility is to resort to reinforcement learning procedures, which depend just on the availability of a measure of success rather than on an error measure.

The number of learning cycles required ranges widely in the applications described, depending on the complexity of the mapping to be learned as well as on the accuracy required. Only ten trials are needed to get a useful mapping in the case of inverse dynamics using CMAC. The explanation is that only a very coarse mapping is needed, since the neural controller is used as a feed-forward component in combination with a fixed-gain feedback controller. The number of trials rises to a few hundred in the case of force-motor mappings for insertion of components. One hundred learning cycles suffice to correct the distortions in the inverse kinematics mapping resulting from robot wear-and-tear, while this number rises to nearly 1,000 when the full mapping has to be learned from scratch. And the progression continues, to up to 10,000 trials when the inputs are not spatial coordinates but global descriptors extracted from images. Of course, some of these figures might be considerably lower in the future, especially the last one, if more efficient codings of the input space are found.

Road Map: Robotics and Control Theory

Background: Cerebellum and Motor Control

Related Reading: Arm and Hand Movement Control; Identification and Control; Robot Learning; Sensorimotor Learning

References

D'Souza, A., Vijayakumar, S., and Schaal, S., 2001, Learning inverse kinematics, in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, Piscataway, NJ: IEEE Press, pp. 298–303.

- Gullapalli, V., Barto, A. G., and Gruben, R., 1994, Learning admittance mappings for force-guided assembly, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Los Alamitos, CA: IEEE Computer Society Press, pp. 2633–2638.
- Gullapalli, V., Gruben, R., and Barto, A., 1992, Learning reactive admittance control, in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, Los Alamitos, CA: IEEE Computer Society Press, pp. 1475–1480.
- Jordan, M. I., and Rumelhart, D. E., 1992, Forward models: Supervised learning with a distal teacher, *Cognit. Sci.*, 16:307–354.
- Kröse, B. J. A., and van der Smagt, P. P., 1993, Robot control, in *An Introduction to Neural Networks*, 5th ed., Amsterdam: University of Amsterdam, chap. 7. ♦
- Miller, W. T., Hewes, R. P., Glanz, F. H., and Kraft, L. G., 1990, Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller, *IEEE Trans. Robot. Automat.*, 6:1–9.
- Miyamoto, H., Kawato, M., Setoyama, T., and Suzuki, R., 1988, Feedback-error-learning neural network for trajectory control of a robotic manipulator, *Neural Netw.*, 1:251–265.
- Ritter, H., Martinetz, T., and Schulten, K., 1992, *Neural Computation and Self-Organizing Maps*, New York: Addison-Wesley. ♦
- Ruiz de Angulo, V., and Torras, C., 1997, Self-calibration of a space robot, *IEEE Trans. Neural Netw.*, 8:951–963.
- Samson, C., LeBorgne, M., and Espiau, B., 1990, *Robot Control: The Task Function Approach*, Oxford Engineering Science Series 22, Oxford, Engl.: Oxford Science Publications. ♦
- Schram, G., van der Linden, F. X., Kröse, B. J. A., and Groen, F. C. A., 1996, Visual tracking of moving objects using a neural network controller, *Robot. Auton. Syst.*, 18:293–299.
- Torras, C., 1995, Robot adaptivity, *Robot. Auton. Syst.*, 15:11–23. ♦
- van der Smagt, P., and Bullock, D., Eds., 1997, Can artificial cerebellar models compete to control robots? in *Extended abstracts of the NIPS*97 Workshop*, DLR Technical Report No. 515-97-28, German Aerospace Center (DLR Oberpfaffenhofen).
- Walter, J., and Ritter, H., 1996, Rapid learning with parameterized self-organizing maps, *Neurocomputing*, 12:131–153.
- Wells, G., Venaille, C., and Torras, C., 1996, Vision-based robot positioning using neural networks, *Image Vision Comput.*, 14:715–732.
- Wells, G., and Torras, C., 1998, Selection of image features for robot positioning using mutual information, in *Proceedings of the IEEE Conference on Robotics and Automation*, Los Alamitos, CA: IEEE Computer Society Press, pp. 2819–2826.

Robot Learning

Stefan Schaal

Introduction

Learning robot control, a subclass of the field of learning control, refers to the process of acquiring a sensorimotor control strategy for a particular movement task and movement system by trial and error. Learning control is usually distinguished from adaptive control in that the learning system is permitted to fail during the process of learning, while adaptive control emphasizes single-trial convergence without failure. Thus, learning control resembles the way that humans and animals acquire new movement strategies, while adaptive control is a special case of learning control that fulfills stringent performance constraints, such as may be needed in life-critical systems such as airplanes and industrial robots.

A key question in learning control is what it is that should be learned. In order to address this issue, it is helpful to assume one of the most general frameworks of learning control as originally developed in the middle of the twentieth century in the fields of optimization theory, optimal control, and in particular dynamic programming (Bellman, 1957; Dyer and McReynolds, 1970). Here, the goal of learning control was formalized as the need to acquire a task-dependent control policy π that maps the continuous-valued state vector \mathbf{x} of a control system and its environment, possibly in a time-dependent way, to a continuous-valued control vector \mathbf{u} :

$$\mathbf{u} = \pi(\mathbf{x}, \alpha, t) \quad (1)$$

The parameter vector α contains the problem-specific parameters in the policy π that need to be adjusted by the learning system. Figure 1 illustrates the generic control diagram of learning a control policy. Since the controlled system can generally be expressed as a nonlinear function

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (2)$$

in accordance with standard dynamical systems theory (Strogatz, 1994), the combined system and controller dynamics result in:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \pi(\mathbf{x}, t, \alpha)) \quad (3)$$

Thus, learning control means finding a (usually nonlinear) function π that is adequate for a given desired behavior and movement system. This formal viewpoint allows discussing robot learning in terms of the different methods that have been suggested for learning control policies.

Methods of Learning Robot Control

Learning the Control Policy Directly

It is possible to learn the control policy π directly, i.e., without splitting it into subcomponents, as explained in later sections. For this purpose, the desired behavior needs to be expressed as an optimization criterion $r(t)$ to be optimized over a certain temporal horizon T , resulting in a cost function

$$J(\mathbf{x}(t)) = \int_{t=0}^T r(\mathbf{x}(s), \mathbf{u}(s)) ds$$

or

$$J(\mathbf{x}(t)) = \int_{t=0}^{\infty} \frac{1}{\tau} e^{-(s-t/\tau)} r(\mathbf{x}(s), \mathbf{u}(s)) ds \quad (4)$$

The second formulation of Equation 4 illustrates the use of an infinite horizon time window by introducing a discount factor that

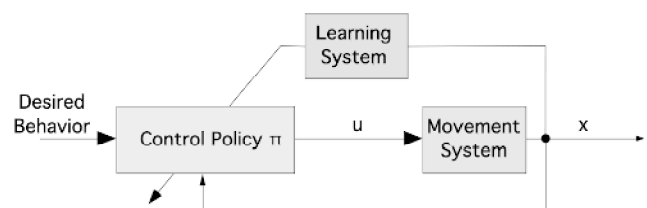


Figure 1. Generic control diagram for learning control.

reduces the influence of values of $r(t)$ in the far future. Note that $r(t)$ is usually a function of the state \mathbf{x} and command \mathbf{u} taken in \mathbf{x} ; i.e., $r(t) = r(\mathbf{x}(t), \mathbf{u}(t))$. Solving such kinds of optimization problems was developed in the framework of dynamic programming (Dyer and McReynolds, 1970) and its recent derivative, reinforcement learning (Sutton and Barto, 1998; see also REINFORCEMENT LEARNING). For reinforcement learning, $r(t)$ corresponds to the “immediate reward” and $J(\mathbf{x}(t))$ to the “long-term reward.” For instance, in the classical task of balancing a pole on a finger, the immediate reward could be +1 at every time step at which balancing was successful, and -1000 if the pole was dropped; the task goal would be to accumulate maximal long-term reward, equivalent to balancing without dropping.

The policy π is acquired with reinforcement learning by, first, learning the optimal function $J(\mathbf{x}(t))$ for every state \mathbf{x} , usually by a technique called temporal difference learning (see REINFORCEMENT LEARNING), and then deducing the policy π as the command \mathbf{u} in every state \mathbf{x} that leads to the best future payoff, i.e.,

$$\mathbf{u} = \max_{\mathbf{u}'} (r(\mathbf{x}(t), \mathbf{u}'(t)) + J(\mathbf{x}(t+1)))$$

where

$$\mathbf{x}(t+1) = \mathbf{x}(t) + f(\mathbf{x}(t), \mathbf{u}'(t))\Delta t \quad (5)$$

where Δt is an arbitrarily chosen constant time step for sampling the system’s behavior. Many variations of reinforcement learning exist, including methods that avoid estimating the optimization function $J(\mathbf{x}(t))$ (see REINFORCEMENT LEARNING).

Learning the Control Policy in a Modular Way

Theoretically, the techniques of reinforcement learning and dynamic programming would be able to solve any robot learning problem that can be formulated as sketched in the previous section. Practically, however, this is not true, since reinforcement learning requires a large amount of exploration of all actions and states for proper convergence of learning as well as appropriate representations for the function $J(\mathbf{x}(t))$. Traditionally, $J(\mathbf{x}(t))$ needs to be represented as a lookup table; that is, for every state \mathbf{x} a specific table cell holds the appropriate value, $J(\mathbf{x}(t))$. For continuous-valued states, discretization of the individual dimensions is needed. For high-dimensional systems, this strategy leads to an explosion of lookup table cells. For example, for a 30-dimensional movement system where each dimension is split into just two cells, an astronomical number of 2^{30} cells would be required. Exploring all these cells with a real robot would take forever, and even in computer simulations such problems are not tractable. Newer approaches employ special neural networks for learning and representing $J(\mathbf{x}(t))$ (e.g., Sutton and Barto, 1998), but problems with high-dimensional movement systems remain daunting.

A possible way to reduce the computational complexity of learning a control policy comes from modularizing the policy (Figure 2). Instead of learning the entire policy in one big representation, one could try to learn subpolicies that have reduced complexity and

subsequently build the complete policy out of such subpolicies. This approach is also appealing from the viewpoint of learning multiple tasks: some of the subpolicies may be reused in another task, which should strongly facilitate learning new tasks.

Motor control with subpolicies has been explored in various fields under the names of, e.g., schema theory (see SCHEMA THEORY), behavior-based robotics (see REACTIVE ROBOTIC SYSTEMS), pattern generators (see MOTOR PATTERN GENERATION), and movement primitives (Schaal, 1999). Robot learning with such modular control systems, however, is still in its infancy. Reinforcement learning has recently begun to formulate a principled approach to this problem (Sutton, Precup, and Singh, 1999). Another route of investigating modular robot learning comes from formulating subpolicies as nonlinear dynamical systems (Mussa-Ivaldi and Bizzi, 1997; Schaal and Sternad, 1998). However, all this research is still of a preliminary nature and not yet applicable to complex robot learning problems.

Indirect Learning of Control Policies

The previous sections assumed that motor commands are directly generated based on the information of the state of the world \mathbf{x} , i.e., from the policy function π . For many movement systems, however, such a *direct* control strategy is not advantageous because it fails to reuse modules in the policy that are common across multiple tasks. This view suggests that, in addition to a modularization of motor control and learning in form of a mixture of simpler policies (Figure 2), modularization can also be achieved in terms of a functional structuring within each control policy. A typical example is to organize the control policy into several processing stages, as illustrated in Figure 3 and also discussed as *indirect* control in MOTOR CONTROL, BIOLOGICAL AND THEORETICAL (q.v.). Most commonly, the policy is decomposed into a planning stage and an execution stage, a strategy that is typical for most robot controllers but also likely to be used in motor control of primates. Planning generates a desired *kinematic* trajectory, i.e., a prescribed way in which the state of the movement system is supposed to change in order to achieve the task goal. Execution transforms the plan into appropriate motor commands.

Separating planning and execution is highly advantageous. For instance, in reaching movements located toward a target, a *direct* approach to robot learning would require learning a new policy for every target location: the desired behavior is to minimize the distance between the hand and the target, and because of the complex dynamics of an arm, different target locations require very different motor commands for reaching. In contrast, an indirect control approach requires learning only the movement execution module, usually in the form of an inverse model (see below). The execution module is valid for any target location. For simplicity, movement plans can be kept the same for every target location, such as a straight line between the target and the starting position of the arm, with a bell-shaped velocity profile. Planning such movement kinematics requires only one-time learning of the robot kinematics model and using standard kinematic planning algorithms (Sciavicco and Siciliano, 1996) that can easily cope with any reachable target location. Thus, after the execution module has been acquired, the problem of reaching is a largely solved, no matter where the target is.

Depending on the task, planning can take place in external kinematic variables (e.g., Cartesian or end-effector space) or in internal kinematic variables (e.g., joint space for a human-like arm). If the planning space does not coincide with the space where motor commands are issued, coordinate transformations are required to map the external motor plan into intrinsic coordinates. This problem is commonly discussed as the inverse kinematics problem of

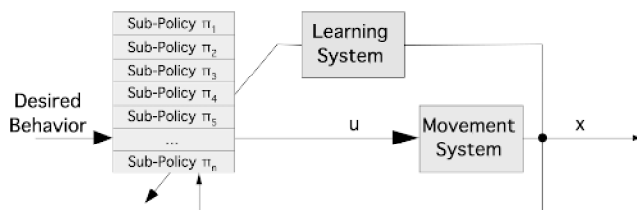
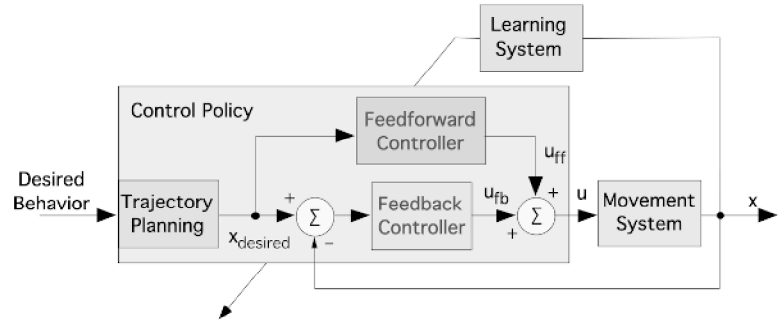


Figure 2. Learning control with subpolicies.

Figure 3. Learning control with functional decomposition.

ROBOT ARM CONTROL (q.v.), a problem that equally needs to be addressed by biological movement systems.

To transform kinematic plans into motor commands, standard methods from control theory can be employed (e.g., Sciavicco and Siciliano, 1996). Figure 3 illustrates a typical example that uses a feedforward/feedback mechanism, called a *computed torque controller*, that enjoys popularity in both robotic systems and models of biological motor control (Jordan, 1996). The feedback controller is of classical proportional derivative (PD) type, while the feedforward controller contains an inverse dynamics model of the movement system (see CEREbellum AND MOTOR CONTROL).

From the point of robot learning, functional modularity also decomposes the learning problem into several independent learning problems. The modules of the execution stage can be learned with supervised learning techniques (discussed later; see also CEREbellum AND MOTOR CONTROL). For various types of movements, kinematic movement plans can be highly stereotyped, as was described for the reaching example, such that no learning is required for planning. For complex movements such as a tennis serve, planning requires more sophistication, and the same reinforcement learning methods of *direct* control can be applied, the only difference being that the motor commands u are replaced with a desired change in trajectory \dot{x}_d . Applying reinforcement learning to kinematic planning is less complex than solving the complete *direct* control problem since the highly nonlinear transformation from kinematic plans to motor commands does not need to be acquired anymore, but how to perform reinforcement learning for high-dimensional movement systems still remains an open research problem.

Imitation Learning

A topic in robot learning that is of increasing interest is *imitation learning*. The idea of imitation learning is intuitively simple: a student watches the performance of a teacher and subsequently uses the demonstrated movement as a seed to start his or her own movement. The ability to learn from imitation has a profound impact on how quickly new skills can be acquired (Schaal, 1999). From the viewpoint of learning theory, imitation can be understood as a method to bias learning toward a particular solution, the teacher's. Motor learning proceeds afterward, as described in the other sections of this article. However, not every representation for motor learning is equally suited to be biased by imitation (Schaal, 1997). For instance, a robot using direct control can hardly profit from a demonstration, as motor commands are not perceivable, but a robot using indirect control could first extract a kinematic plan from the demonstration and then use it for starting its own learning. Imitation thus imposes interesting constraints on the structure of a learning system for motor learning.

Learning of Motor Control Components

Whether direct or indirect control is employed in a motor learning problem, the core of the learning system usually requires methods of supervised learning of regression problems, called function approximation in the neural network and statistical learning literature. *Function approximation* is concerned with approximating a nonlinear function $y = f(x)$ from noisy data, drawn according to the data-generating model:

$$y = f(x) + \varepsilon \quad \text{where } x \in \mathcal{R}^n, y \in \mathcal{R}^m, E\{\varepsilon\} = 0 \quad (6)$$

i.e., x is an n -dimensional continuous-valued vector, y is an m -dimensional continuous-valued vector, and ε a zero-mean random "noise" variable. By comparing the generic form of a policy in Equation 1 or a dynamics model in Equation 2 with Equation 6, it is apparent that learning such functions falls into the framework of function approximation.

Neural Network Approaches to Function Approximation

Many different methods of function approximation exist in the literature (see LEARNING AND STATISTICAL INFERENCE). For present purposes, it is sufficient to classify all these algorithms into two categories, spatially localized (*local*) algorithms and spatially non-localized (*global*) algorithms. The power of learning in neural networks comes from the nonlinear activation functions that are employed in the hidden units of the neural network. *Global* algorithms use nonlinear activation functions that are not limited to a finite domain in the input space (x -space) of the function. The prototypical example is the sigmoid function in Figure 4A that outputs a value of roughly one for any input greater than about one. In contrast, *local* algorithms make use of nonlinear activation functions that are different from zero only in a restricted input domain. The Gaussian function in Figure 4B exemplifies this class of functions. Even though both local and global algorithms are theoretically capable of approximating arbitrarily complex nonlinear functions, learning speed, convergence, and applicability to high-dimensional learning problems differ significantly between the two (Schaal and Atkeson, 1998).

To understand how global algorithms approximate nonlinear functions, it is helpful to think of them as an octopus whose tentacles stretch out and span the complex surface described by $y = f(x)$, except that the tentacles exist in an n -dimensional space. Global algorithms can work quite well for problems with many input dimensions, since their nonlocal activation function (i.e., their tentacles) can span even huge spaces quite efficiently. But global algorithms usually require very careful training procedures so that the tentacles learn how to stretch appropriately into all directions. In particular, if at some point in training, data are only provided for a restricted area in input space, the tentacles may focus too much on approximating this area and, while doing so, forget maintaining the "tentacle posture" in previously learned areas—a phe-

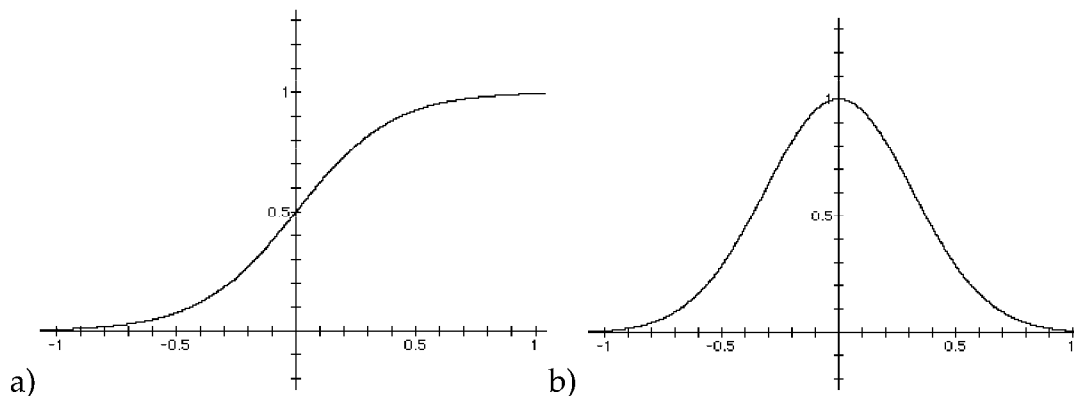


Figure 4. Nonlinear activation functions used in neural networks. *A*, The sigmoid function, a spatially global function. *B*, The Gaussian function, a spatially localized function.

nomenon called catastrophic interference (Schaal and Atkeson, 1998). Catastrophic interference is particularly pronounced in incremental learning problems, where training data come point after point and can only be used once for updating the algorithm. Unfortunately, this is the typical scenario in robot learning. Together with the problem of how to select the right number of hidden units (i.e., the right number of tentacles), it becomes quite complicated to train global algorithms for high-dimensional robot learning problems.

In contrast, local learning algorithms have quite different characteristics. The metaphor for local learning is simply that they approximate the complex regression surface with the help of small local patches, for instance locally constant or locally linear functions (Atkeson, Moore, and Schaal, 1997). Problems of how many patches need to be allocated, where to place them, how large they should be in input space, and how to learn them incrementally have largely been solved (Schaal and Atkeson, 1998). The biggest problem of local algorithms is the curse of dimensionality, i.e., the exponential explosion in the number of patches that are needed in high-dimensional input spaces. For instance, assume that we want to divide every input dimension of a function approximation problem into ten local regions. For two input dimensions, this strategy would result in 10^2 local regions, for three inputs in 10^3 regions, and for n inputs in 10^n regions. Even for only 12 input dimensions, this number reaches the number of neurons in the human brain. The only way to avoid this problem is to make the patches larger, but then the quality of function approximation becomes unacceptably inaccurate. There is theoretically no way out of the curse of dimensionality—but empirically, it turns out not to be a problem. The example demonstrating the curse of dimensionality can be turned around in our favor: How long would it take a robot system to generate all the data points to fill these big spaces? As an example, collecting 10^{12} data points at 100 Hz sampling frequency would take more than 300 years of uninterrupted movement! Thus, no actual robot will ever be able to generate enough data to fill these huge spaces. This argument triggers a most important question: What kind of data distributions are actually realized by robotic (or biological) movement systems? Vijayakumar et al. (2002) found that distribution had only about four to six dimensions locally in a robot learning problem that had 21 input dimensions, a finding that was also duplicated in other robot learning domains (Vlassis, Motomura, and Krose, 2002). Local learning can exploit this property by using techniques of local dimensionality reduction and can thus learn efficiently even in very high-dimensional spaces (Vijayakumar et al., 2002). Thus, for the time being, local learning algorithms seem to be better suited for robot learning.

Specific Function Approximation Problems in Robot Learning

Applying function approximation to problems of robot learning requires a few more considerations. The easiest applications are those of straightforward supervised learning, i.e., where a teacher signal y is directly available for every training point x . For example, learning a dynamics model of the form of Equation 2 usually falls into this category if the inputs x and u and the output \dot{x} can be measured directly from sensors. Many other problems of ROBOT ARM CONTROL (q.v.) are of a similar nature.

Learning becomes more challenging when instead of the teacher signal only an error signal is provided, and the error signal is just approximate. Assume we have such an error signal e when the network predicted a particular \hat{y} for a given input x . From this information, we can create a teacher signal $y = \hat{y} + e$ and train the network with this target. However, if e was only approximate, y is not the true target, and later on during learning another (hopefully more accurate) teacher signal may be formed for training the network. Thus, learning proceeds with “moving targets,” which is called a nonstationary learning problem. For such learning tasks, neural networks need to have an appropriate amount of plasticity in order to keep on changing until the targets become correct. On the other hand, it is also important that the network converge at some point and average out the noise in the data, i.e., that the network not have too much plasticity. Finding appropriate neural networks that have the right amount of plasticity-stability trade-off is a nontrivial problem, and so far, heuristic solutions dominate the literature.

Nonstationary learning problems are unfortunately quite common in robot learning. Learning the optimization function $J(x(t))$ in reinforcement learning is one typical example since the temporal difference algorithm (Sutton and Barto, 1998) can only provide approximate errors. Other examples include feedback error learning and learning with distal teachers (Jordan, 1996). Both of these methods address the problem that in learning control, we usually receive errors in only sensory variables, such as positions and velocities, but what is needed to train a control policy is an error in motor commands. Thus, feedback error learning creates an approximate motor command error by using the output of a linear feedback controller as the error signal. Learning with distal teachers accomplishes essentially the same goal, except that it employs a learned forward model to map an error in sensory space to an approximate motor error.

Applications

While the theoretical development of learning control has progressed significantly in recent years, applications in actual robots

have remained rather sparse because of the significant computational burden of most learning algorithms and the real-time constraints of actual robots. Reinforcement learning in actual robots remains largely infeasible, and only few examples exist in simplified setups (e.g., see references in Atkeson et al., 1997; Sutton and Barto, 1998; Schaal, 1999). Learning of internal models for robot control has found increasingly more widespread application owing to significant advances in the computational efficiency of supervised learning algorithms (e.g., see references in Atkeson et al., 1997; Vijayakumar et al., 2002; Vlassis et al., 2002).

Discussion

Robot learning is a surprisingly complex problem. It needs to address how to learn from (possibly delayed) rewards, how to deal with very high-dimensional learning problems, how to use efficient function approximators, and how to embed all the elements in a control system with real-time and robust performance. A further difference from many other learning tasks is that in robot learning, the training data are generated by the movement system itself. Efficient data generation, i.e., exploration of the world, will result in fast learning, while inefficient exploration can prevent successful learning altogether (see REINFORCEMENT LEARNING). Insofar as very few robots in the world are equipped with learning capabilities, research on robot learning is still in an early stage of development.

Road Map: Robotics and Control Theory

Related Reading: Imitation; Q-Learning for Robots; Reinforcement Learning in Motor Control; Robot Arm Control; Sensorimotor Learning

References

- Atkeson, C. G., Moore, A. W., and Schaal, S., 1997, Locally weighted learning for control, *Artif. Intell. Rev.*, 11:75–113. ♦
- Bellman, R., 1957, *Dynamic Programming*, Princeton, NJ: Princeton University Press.
- Dyer, P., and McReynolds, S. R., 1970, *The Computation and Theory of Optimal Control*, New York: Academic Press.
- Jordan, M. I., 1996, Computational aspects of motor control and motor learning, in *Handbook of Perception and Action* (H. Hever and S. W. Keele, Eds.), New York: Academic Press. ♦
- Mussa-Ivaldi, F. A., and Bizzi, E., 1997, Learning Newtonian mechanics, in *Self-Organization, Computational Maps, and Motor Control* (P. Morasso and V. Sanguinetti, Eds.), Amsterdam: Elsevier, pp. 491–501.
- Schaal, S., 1997, Learning from demonstration, in *Advances in Neural Information Processing Systems 9* (M. C. Mozer, M. Jordan, and T. Petsche, Eds.), Cambridge, MA: MIT Press, pp. 1040–1046.
- Schaal, S., 1999, Is imitation learning the route to humanoid robots? *Trends Cogn. Sci.*, 3:233–242. ♦
- Schaal, S., and Atkeson, C. G., 1998, Constructive incremental learning from only local information, *Neural Comput.*, 10:2047–2084.
- Schaal, S., and Sternad, D., 1998, Programmable pattern generators, in *Proceedings of the 3rd International Conference on Computational Intelligence in Neuroscience*, Research Triangle Park, NC: New York: Association for Computing Machinery, pp. 48–51.
- Sciavicco, L., and Siciliano, B., 1996, *Modeling and Control of Robot Manipulators*, New York: McGraw-Hill.
- Strogatz, S. H., 1994, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Reading, MA: Addison-Wesley.
- Sutton, R. S., and Barto, A. G., 1998, *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press. ♦
- Sutton, R. S., Precup, D., and Singh, S., 1999, Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning, *Artif. Intell.*, 112:181–211.
- Vijayakumar, S., D'Souza, A., Shibata, T., Conradt, J., and Schaal, S., 2002, Statistical learning for humanoid robots, *Auton. Robots*, 12:59–72.
- Vlassis, N., Motomura, Y., and Krose, B., 2002, Supervised dimension reduction of intrinsically low-dimensional data, *Neural Computat.*, 14:191–215.

Robot Navigation

José del R. Millán

Introduction

Mobile robots are gradually leaving the laboratories to undertake service tasks ranging from providing surveillance of buildings and supervision of plants, to transporting patients and delivery items, to cleaning houses and guiding people. Independently of the assigned task, the basic capability of a mobile robot is to move to its destination—or sequence of targets—efficiently (e.g., along short trajectories) and safely (i.e., without collisions). *Navigation* refers to the capability of selecting and performing a path from a current position to a desired location. Implicit in this definition is the ability to adapt the goal-oriented behavior to the complexity of the task. If a target location is either visible or identified by a landmark (or sequence of landmarks), a simple stimulus-response strategy can be adopted (REACTIVE ROBOTIC SYSTEMS). However, targets are often neither visible nor identified by any sequence of cues. In this case, for a robot to navigate it must first determine its position with respect to the target. This is the *localization* problem. Finally, to perform more flexible and sophisticated navigation (e.g., planning short cuts), the robot needs a model of the environment encoding the spatial relationships between locations. Acquiring such a model is the *map-building* problem.

Like any other robotic system (ROBOT ARM CONTROL), mobile robots must rely on on-line sensory information to take actions.

But, unlike most arm robots, sensory information cannot only be proprioceptive (i.e., an odometry process that gives the robot's coordinates based on internal encoders); it must also provide exteroception (i.e., information about the external environment). Indeed, odometry alone accumulates errors due to slippage, which will make the robot get lost and crash sooner or later. Mobile robots mainly use three types of sensors to perceive their surroundings, namely tactile sensors that inform about contacts, range sensors (lasers, ultrasounds, or infrareds) that return distances after appropriate transformations, and vision. Given the opposite strengths and weaknesses of the different sensors, an orthogonal issue not covered by this article is that of sensor fusion (SENSOR FUSION).

A common property shared by all types of sensors is their noisy responses. This sensor uncertainty, together with the inaccuracy of the robot's actuators and the unpredictability of real environments, makes the design of mobile robot controllers a difficult task. The complexity of the sensorimotor mapping underlying robot navigation yields two main consequences. First, simulations, though useful, are not enough to reproduce the actual agent-environment interaction. Second, robots must build their control strategies based on their own sensory perceptions of the real world (i.e., *embodiment*). Human-made controllers are, except for simple tasks and environments, inadequate because the designer must anticipate

every possible situation the robot might face and must tune the controller's parameters to achieve efficient performance. An alternative is to endow robots with *learning capabilities* in order to acquire autonomously their control system and to adapt their behavior to never-experienced situations (i.e., *generalization*).

Artificial neural networks (ANN) offer a suitable learning framework to model the basis of adaptive behavior. Indeed, their noise robustness and generalization capabilities allow robots to cope with the nature of their interaction with the world and to build appropriate sensorimotor mappings. The next three sections discuss ANN approaches to localization, map building, and navigation.

Localization

To solve complex navigation tasks, mobile robots must self-localize in the environment by relying on their exteroceptive and proprioceptive sensory inputs. In general, localization calls on *place recognition*. To localize itself, the robot can either simply memorize the sensory perceptions observed during exploration or can learn a more complex representation (map) encoding spatial relationships between experienced local perceptions. How the robot acquires a map is discussed in the next section.

Strictly speaking, only Thrun (1998a,b) uses an ANN for localization. In the remaining approaches, the robot's perception is matched against the model, which has an ANN organization, and its location is derived from that associated to the winning unit. Actually, the robot's perception can be the current sensory reading plus odometry information (Zimmer, 1996), sensory data averaged as the robot moves (Mataric, 1992), or egocentric views obtained from the sensory perceptions (Recce and Harris, 1996).

An alternative is to transform raw sensory data into a more reliable representation through ANN (Thrun, 1998a). In particular, a feedforward ANN is trained through *backpropagation* to generate a local *occupancy grid* from the current sensory perception. Such a grid is a discrete representation of the space around the robot, where each cell has a value that estimates the occupancy probability of the corresponding area of the world. After exploration, the localization algorithm searches for the previously stored grid that best matches the current local map. There are two advantages of using a neural sensor interpretation to build local occupancy grids: the ANN does not assume any noise distribution, and it interprets sensor readings simultaneously. On the other hand, the shortcoming of this approach is its computational cost, as building an $n*n$ grid requires $n*n$ calls to the ANN.

A totally different approach is to learn what environmental features are the most relevant landmarks for localization. Thrun (1998b) trained a feedforward ANN to optimize a Bayesian measure of probabilistic localization. Training was done on samples collected during an exploration phase in which each sample consisted of a sensory perception and its location. During operation, the robot averaged the ANN response for the k nearest neighbor samples to its estimated location. This approach demonstrated its superiority to hand-coded localization methods based on using doors and ceiling lights as landmarks.

Building

The representations a robot may learn are of two main types, namely *metric* and *topological*. In the former, maps quantitatively reproduce the geometric and spatial features of the environment. This is computationally expensive and vulnerable to errors that affect the metric information. Topological maps are more qualitative and consist of a graph, on which nodes represent perceptually distinct places (landmarks) and arcs indicate spatial relations between them. They are less vulnerable to sensory errors, and enable fast planning since the latter reduces to a simple search process in

a graph. However, topological representations rely on the existence of ever-recognizable landmarks.

One of the most popular approaches to building metric maps is based on the use of occupancy grids. Thrun (1998a) trained a feedforward ANN to create a local occupancy grid modeling the space surrounding the robot. Successive local grids generated as the robot explored its environment were then combined to produce an accurate global metric map. Once the global metric map was available, a topological graph could be abstracted off-line. This greatly reduces the cost of planning paths between different locations in the environment. This approach (in conjunction with the localization process discussed before) is implemented in museum tour-guided robots. An alternative is to use the same neural sensor interpretation but only for deriving coarse geometrical features from which to build up on-line a variable-resolution partitioning of the environment (Arleo, Millán, and Floreano, 1999). The environment is discretized in cells of different sizes, with a high resolution only on critical areas (i.e., around obstacles). The resulting map combines geometrical and topological aspects that are learned simultaneously.

Among topological approaches, Mataric's model (1992) builds a sparse graph in which each node represents a unique predefined landmark. Spatial relationships between landmarks are encoded by neighbor links in the graph, which produces a structure isomorphic to the topology of the environment. Disambiguation between similar sensory patterns is done by spreading expectations from the currently active unit to its neighbors (contextual discrimination) or by attaching metric information to the units. Self-organizing or organizing feature maps or Kohonen maps (SELF-ORGANIZING FEATURE MAPS) provide an alternative way of acquiring topological maps (Kurz, 1996; Zimmer, 1996). A self-organizing, or *Kohonen*, map clusters the sensory perceptions gathered during the exploration phase of the robot (Kurz, 1996). The dimensionality of the Kohonen map matches the robot's degrees of freedom, either two or three depending on whether or not the robot moves with a constant orientation. As a result, neighboring units in the learned Kohonen map correspond to neighboring areas in the sensory space. The problem is that there is no guarantee that neighbor areas in sensory space are also close in metric space. Still worse, due to the limitations of its sensors a robot may have similar sensory perceptions from two different metric locations. A possible solution is to include odometry information in the self-organizing process (Zimmer, 1996), which presumes a reliable localization process. Another solution is to use the temporal sequence of sensory perceptions and not just the current one. This can be achieved by means of a recurrent Kohonen map. Finally, instead of using a self-organizing map with a fixed structure (dimensionality and number of units), it is also possible to learn the topological map of the environment by means of a *dynamic self-organizing map*. This method adds a new unit whenever the current sensory perception is sufficiently different from any existing unit (Millán, 1997; Zimmer, 1996), or is using statistical measures (Zimmer, 1996). In this kind of network the topology of the environment is kept in the links between units. It is worth noting that the ADAPTIVE RESONANCE THEORY (q.v.) can yield quantizations of the environment similar to a dynamic self-organizing map, but the resulting map does not exhibit topological relationships between the units.

Map learning systems engineered so far are not as robust, flexible, and adaptable as biological spatial learning solutions. Neurophysiological findings suggest that the spatial memory of mammals is supported by location-sensitive neurons (*place cells*) in the *Hippocampus* (see HIPPOCAMPUS: SPATIAL MODELS). Recent research in robot navigation has moved toward biologically inspired approaches to developing autonomous systems that mimic mammalian spatial learning capabilities. For example, Recce and Harris (1996) put forward a map-building model that ascribes the spatial

memory function to the hippocampus. The authors assume that a place cell in the robot's hippocampus memorizes a complete ego-centric map of the environment. This is a strong requirement for both robots and animals, especially if operating in middle- and large-scale environments. Arleo and Gerstner (2000) propose a hippocampal model in which unsupervised *Hebbian* learning is applied to acquire a spatial map incrementally and on-line. The representation consists of a population of localized overlapping place fields that provide a stable coarse space code. The robot establishes place fields by extracting spatiotemporal properties of the environment from visual inputs and solves visual ambiguities by taking into account proprioceptive self-motion signals.

Navigation

If the robot has acquired (or is given) a topological map, then whenever it is requested to navigate to a given destination it simply searches for an optimal route in the graph and uses elemental *behaviors* to move from one node to the next along that route. However, building and maintaining consistent global maps of the environment is far from being a trivial problem, since noisy sensory data may introduce errors into the maps. Also, unless the robot is equipped with good exploration strategies, it may fail to model the whole environment and topological relationships. For example, most map building approaches rely on a wall-following (and obstacle avoidance) behavior that prevents the robot from visiting open or cluttered areas. Thus, while in operation, the robot will never take shortcuts. Alternatively, the robot can directly use behaviors to reach its destination without resorting to any map. In this section we discuss how the necessary behaviors can be learned by means of ANN.

A behavior is a set of perception-action rules that provide the robot with a given functionality such as obstacle avoidance (REACTIVE ROBOTIC SYSTEMS). Perception-action mappings can be learned off-line from representative training sets, mainly through *supervised* techniques (Pomerleau, 1993; Sharkey, 1998). Pomerleau's system (1993) is a paradigmatic example of the potentiality of the supervised approach. He trained a feedforward ANN to drive a car in a variety of roads. Training data were gathered by observing a human expert driving the vehicle. In particular, inputs corresponded to images of the road in front of the car and desired outputs corresponded to the driver's steering direction. After learning, the ANN controller made the vehicle follow the road by keeping it in the center of the lane.

Instead of requiring a human for data collection, an alternative is to use a preprogrammed controller as an initial teacher. Sharkey (1998) made feedforward ANN learn from an initial behavior-based controller to approach a target while avoiding obstacles. The final neural controller, obtained through a bootstrapping process, performed better than the original controller did. Nevertheless, the resulting network also inherited limitations of the initial controller. This illustrates one of the fundamental limitations of supervised learning and leads to the necessity of *autonomous* robot learning.

Autonomous robots must train themselves on-line in order to cope with weak and incomplete training examples. REINFORCEMENT LEARNING (q.v.) is an appropriate paradigm to achieve this. A reinforcement-based robot can improve its performance over time without needing extensive previous knowledge about the task. This is quite appealing but, on the other hand, makes the learning process very slow. The following section will describe several extensions to the basic reinforcement learning framework that considerably speed up the convergence to suitable sensorimotor mappings (or policies, as they are customarily called in the control and reinforcement learning fields), thus making it possible to build practical learning mobile robots.

Lin (1991) combined *Q-learning* (a widely studied reinforcement learning technique) and teaching. The controller had one feedforward network for each discrete action the robot can perform. The input to each ANN was the current sensory perception and the output was a prediction of the *Q-value* of that perception-action pair. The robot normally took the action with the highest Q-value (Q-LEARNING FOR ROBOTS). In Lin's experiment, a human teacher brought the robot to its goal several times along efficient paths. Then, the robot learned the appropriate Q-values, and hence good policies, from these examples. The taught navigation sequences helped reinforcement learning by biasing the search for suitable actions toward promising parts of the action space.

Thrun (1995) also used Q-learning, but integrated it with explanation-based learning (EBL). EBL requires a domain theory that is previously learned by a set of feedforward ANN (action models) in a supervised manner, one network for each discrete action the robot can perform. Each network receives the current sensory perception and predicts the next perception and Q-value. In addition to action models, the robot has also a Q network per action similar to that of Lin (1991). As before, the Q networks encode the control policies. Finally, for each actual sequence of actions taking the robot either to the goal or to a failure, the robot explains the observed example in terms of its domain theory by computing the derivatives of the policy with respect to the action model networks. These derivatives are used to bias the supervised learning of the policy (i.e., the Q networks). It is worth noting that since the action models are task independent, they are learned once and can be used across the different tasks faced by the robot.

The previously discussed reinforcement-based robots perform discrete actions, while for smooth operation they should take continuous actions. Millán (1996, 1997) has implemented *Actor-Critic* architectures instead of Q-learning for this purpose. Key components of his learning architecture are the use of *local networks* and the incorporation of *bias* into the network. Local networks make the robot learn incrementally new sensorimotor rules (or tune existing ones) without degrading the performance of other rules. The robots use built-in reflexes (basic domain knowledge) as bias. There are two benefits of bias. First, it accelerates the learning process since it focuses the search process on promising parts of the action space immediately. Second, it makes the robot operational from the very beginning and increments the safety of the learning process. The ANN is trained on-line by means of a combination of reinforcement learning and self-organizing rules. Every time the robot fails to generalize its previous experience to the current sensory perception, it uses the reflexes and adds a new unit to the network. The robot may also add a new unit whenever it receives an advice from humans. This unit is integrated into a dynamic self-organizing map and associates a region around the current perception to either the computed reflex or the advice. The resulting sensorimotor rule is then tuned by means of reinforcement learning and self-organizing rules. Experimental results show that a few minutes suffice for the robot to navigate efficiently in office environments of moderate complexity, in which the robot can easily get trapped inside concave areas.

Recently, reinforcement learning has been shown to be a suitable framework to model reward-based navigation in animals. For example, Arleo and Gerstner (2000) employed Q-learning in continuous space to drive action units one synapse downstream from their hippocampal place cells. Due to the coarse space code provided by the localized overlapping place fields, Q-learning converges in few trials, which is consistent with the rapid acquisition of goal-oriented behavior of animals. Several action modules share the same space representation and guide the robot to multiple targets.

Discussion

For mobile robots to undertake real-world tasks with unreliable sensors and actuators, whose responses greatly depend on the spe-

cific working environment, they must exhibit adaptive capabilities. Neural networks naturally cope with the learning task of analyzing the perception-action interactions for navigation, localization, and map building. Even though different ANN approaches have solved some instances of these three aspects of robot navigation, there does not exist a complete navigation system that is purely made of neural components. From an engineering standpoint, such a complete mobile robot must incorporate other types of learning techniques to generate more abstract models of its perceptions, actions, and sensorimotor rules (Kaiser et al., 1995). In addition, the engineering perspective calls for combining learning capabilities with alternative techniques to build successful mobile robots such as those of Thrun and co-workers (Thrun, 1998a). On the other hand, a different perspective looks at animals for inspiration to develop the necessary neural components of a complete robot navigation system. There are numerous efforts along this *bio-inspired* direction (BIOLOGICALLY INSPIRED ROBOTICS and NEUROETHOLOGY, COMPUTATIONAL) in which reinforcement-based learning (REINFORCEMENT LEARNING and Q-LEARNING FOR ROBOTS) and hippocampal models (COGNITIVE MAPS and HIPPOCAMPUS: SPATIAL MODELS) are keystones of this type of future intelligent (because adaptive) mobile robots.

Road Map: Robotics and Control Theory

Related Reading: Cognitive Maps; Embodied Cognition; Hippocampus: Spatial Models; Motion Perception: Navigation; Potential Fields and Neural Networks

References

- Arleo, A., Millán, J. del R., and Floreano, D., 1999, Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation, *IEEE Trans. Robot. Automat.*, 15:990–1000.
- Arleo, A. and Gerstner, W., 2000, Spatial cognition and neuro-mimetic navigation: A model of hippocampal place cell activity, *Biol. Cyb.*, 83:287–299.
- Kaiser, M., Klingspor, V., Millán, J. del R., Accame, M., Wallner, F., and Dillman, R., 1995, Using machine learning techniques in real-world mobile robots, *IEEE Expert*, 10:37–45. ♦
- Kurz, A., 1996, Constructing maps for mobile robot navigation based on ultrasonic range data, *IEEE Trans. Syst. Man Cybern.—Part B*, 26:233–242.
- Lin, L.-J., 1991, Programming robots using reinforcement learning and teaching, in *Proceedings of the Ninth National Conference on Artificial Intelligence*, Menlo Park, CA: AAAI Press/MIT Press, pp. 781–786.
- Mataric, M. J., 1992, Integration of representation into goal-driven behavior-based robots, *IEEE Trans. Robot. Automat.*, 8:304–312.
- Millán, J. del R., 1996, Rapid, safe, and incremental learning of navigation strategies, *IEEE Trans. Syst. Man Cybern.—Part B*, 26:408–420.
- Millán, J. del R., 1997, Incremental acquisition of local networks for the control of autonomous robots, in *Proceedings of the Seventh International Conference on Artificial Neural Networks*, Heidelberg, Germany: Springer-Verlag, pp. 739–744.
- Pomerleau, D. A., 1993, *Neural Network Perception for Mobile Robot Guidance*, Boston: Kluwer.
- Recce, M., and Harris, K. D., 1996, Memory for places: A navigational model in support of Marr's theory of hippocampal function, *Hippocampus*, 6:735–148.
- Sharkey, N. E., 1998, Learning from innate behaviors: A quantitative evaluation of neural network controllers, *Auton. Robots*, 5:317–334.
- Thrun, S., 1995, An approach to learning mobile robot navigation, *Robot. Auton. Syst.*, 15:301–319.
- Thrun, S., 1998a, Learning maps for indoor mobile robot navigation, *Artif. Intell.*, 99:21–71. ♦
- Thrun, S., 1998b, Bayesian landmark learning for mobile robot localization, *Machine Learn.*, 33:41–76.
- Zimmer, U. W., 1996, Robust world-modelling and navigation in a real world, *Neurocomputing*, 13:247–260.

Rodent Head Direction System

David S. Touretzky and William E. Skaggs

Introduction

The brain of the rat contains an inertial compass distributed across a collection of anatomical areas (Figure 1). Head direction (HD) cells in these areas fire maximally when the rat's head is pointed in a specific *preferred direction*, with a gradual falloff in firing as the heading departs from that direction. This directional specificity is independent of the rat's location or behavior. Within a population of HD cells, preferred directions are uniformly distributed around the circle. Although there is no topographic ordering of the cells in any known HD area, it is convenient to refer to a population of HD cells as if they were arranged around a ring according to their preferred directions. Thus, given a population of HD cells, for any direction the rat is facing, a "bump" of activity will be observed over the population, and the bump location will shift around the ring as the animal's heading changes.

Head direction is an internal, cognitive representation, not a simple reflection of sensory stimuli or motor activities. Maintenance of the HD activity bump is thought to be evidence for attractor networks operating in the rodent brain. Because of the HD system's interesting functional properties, it has been the focus of a very productive interaction between neurophysiologists and computational modelers.

Sensory Cues and Head Direction

In order to maintain an accurate heading estimate in the dark or in an unfamiliar environment, the rat must integrate its head angular velocity, which is sensed by the vestibular system. Other possible sources of angular velocity information include motor efference copy and optic flow, but lesions of the vestibular nuclei suffice to abolish stable HD responses (Blair, Cho, and Sharp, 1998). Experiments have shown that even passive rotations, if rapid, are compensated for by the HD system (i.e., the animal's heading estimate changes accordingly), probably because they are above the threshold of the vestibular system. Slow passive rotations are not compensated.

While the bump representation is maintained even in the absence of visual input, when rats forage in the dark in a radially symmetric environment such as a cylinder, the HD system drifts out of alignment after a few minutes, owing to cumulative integration error. Normally the HD system utilizes a correction mechanism based on visual landmarks—and perhaps other perceptual cues—to keep itself aligned with the environment.

In an often repeated experimental paradigm, HD cells are recorded while a rat forages for scattered food pellets in a gray cylindrical arena with a white card on the wall serving as a prominent

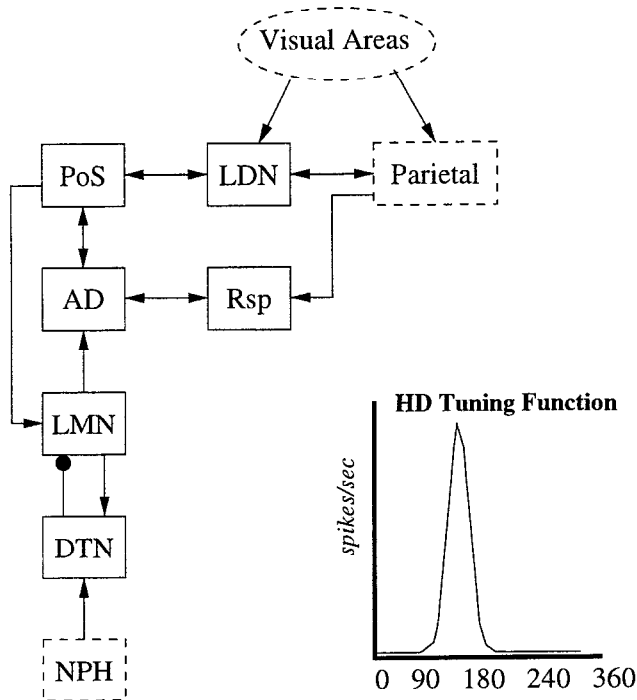


Figure 1. Organization of the rodent head direction system. AD, anterior dorsal nucleus of the thalamus; DTN, dorsal tegmental nucleus; LDN, lateral dorsal nucleus of the thalamus; LMN, lateral mammillary nucleus; NPH, nucleus prepositus hypoglossi; PoS, postsubiculum; Rsp, retrosplenial cortex. Graph at lower right shows the tuning curve of a typical head direction cell.

directional cue. When the cue card is rotated to a different location, the tuning curves of HD cells typically rotate by the same angle as the cue card was rotated, indicating that the cue card is exerting “control” over the HD system. The realignment is a gradual process and may require 1 to 2 minutes to complete (Knierim, Kudrimoti, and McNaughton, 1998).

The influence of visual cues on the HD system is a function of experience. After 1 minute of exposure to a novel environment, little or no control is exerted over the HD system by a cue card, but 8 minutes of exposure is enough for control to develop (Goodridge et al., 1998). The influence of visual landmarks also depends on their being perceived as stable. If rats are repeatedly disoriented before being placed in the cylinder, so that their HD system is randomly oriented, the cue card will appear at a different bearing on each trial, and its influence on the HD system will be substantially diminished (Knierim, Kudrimoti, and McNaughton, 1995).

Conflicts can be introduced between vestibular and visual cues by placing the rat in a cylinder whose floor and wall rotate independently. Under these conditions either the vestibular or the visual cues can win out, depending on circumstances.

The Head Direction Circuit

Head direction cells were first discovered in the postsubiculum (PoS, also called dorsal presubiculum) by Ranck, and studied in depth by Taube, Muller, and Ranck (1990a, 1990b). They were subsequently found in the lateral dorsal nucleus of the thalamus (LDN: Mizumori and Williams, 1993), anterior dorsal thalamic

nucleus (AD: Blair and Sharp, 1995; Taube, 1995), lateral mammillary nucleus (LMN: Blair et al., 1998; Stackman and Taube, 1998), dorsal tegmental nucleus (DTN: Sharp, Tinkelman, and Cho, 2001; Bassett and Taube, 2001), and several other areas. The firing properties of HD cells differ somewhat in the different areas. For example, PoS HD cells are best correlated with the rat's present heading, and their tuning curves are Gaussian shaped and independent of angular velocity. The tuning curves of AD HD cells distort at higher angular velocities, and the activity of these cells is best correlated with the rat's heading roughly 25 ms in the future. This quantity, the *anticipatory time interval* (ATI), was determined by plotting separate tuning curves for left versus right turns. PoS HD cells have an ATI of around zero, but when firing rate is plotted against present heading for an AD HD cell, the peaks of the two curves do not coincide. When firing rate is plotted against future heading, strong overlap of the two tuning curves is obtained for headings around 25 ms in the future. For example, an AD cell whose preferred direction is 120° when the rat is motionless will appear to have a different preferred direction when the rat is turning. During a turn at 320°/s, the cell will fire at its peak rate when the rat's head passes through 112° for a clockwise turn, or 128° for a counterclockwise turn. This can be interpreted as the cell having a fixed preferred direction of 120° but anticipating the rat's heading by 25 ms. The peak firing rates of AD HD cells, but not PoS HD cells, increase with angular velocity.

LMN HD cells have roughly Gaussian tuning curves modulated by angular velocity, but with a preferred turning direction. Cells that prefer clockwise turns increase their firing rate as angular velocity increases in the clockwise direction and decrease their firing rate as angular velocity increases in the counterclockwise direction, relative to the rate observed when the animal's head is not turning at all. Stackman and Taube (1998) observed an ATI of approximately 95 ms for LMN HD cells. But Blair et al. (1998) reported that LMN HD cells have an ATI of only 40 ms, and whereas their peak firing rate changes only slightly with velocity, the width of the tuning curve contracts as angular velocity increases in the preferred turning direction. The tuning curve width does not contract for turns in the opposite direction.

DTN HD cells have much broader tuning curves than PoS, AD, or LMN HD cells. Their firing rate is velocity modulated, and they have a preferred turning direction, as do LMN cells, but their tuning curve widths do not change with velocity (Sharp et al., 2001; Bassett and Taube, 2001). DTN also contains angular velocity cells that are not directionally tuned. These cells fire at a baseline rate when the rat is still, and increase their firing with increasing angular velocity for turns in the preferred direction, or decrease it for turns in the nonpreferred direction. DTN receives projections from the nucleus prepositus hypoglossi (one of the vestibular nuclei), and may be the place where angular velocity information first enters the HD system.

LDN HD cells have tuning curves similar to PoS cells, but they are dependent on visual input (Mizumori and Williams, 1993). If the rat is brought into the recording area in the dark, its LDN HD cells are quiescent. When the lights are turned on, LDN activity increases, resulting in normal HD responses in about 2 minutes. Since LDN projects to PoS, this may be a route by which visual information enters the HD system.

Lesions of AD disrupt the head direction signal in PoS, but lesions in PoS leave the AD HD signal largely intact, although the tuning curve broadens somewhat, and the control of visual landmarks over the HD system is impaired. The persistence of the HD signal may be because AD receives a projection from LMN, and from retrosplenial cortex, which also contains some HD cells. Bilateral but not unilateral LMN lesions abolish the AD HD signal (Blair et al., 1998).

Attractor Models of the Head Direction System

An attractor network is a recurrent neural network whose dynamics generate stable states (attractors) to which the system returns when perturbed. Skaggs et al. (1995) proposed that attractor dynamics could explain the shape of HD cell tuning curves, the stability of HD activity in the face of fluctuating sensory input, the control of the HD system by vestibular and visual cues, and the requirement for landmark stability. Zhang (1996) offered the first mathematical formulation of an attractor network as a model of an HD area. Goodridge and Touretzky (2000) modeled the interactions among PoS, AD, and LMN, including reproducing the distortions of AD tuning curve shapes with angular velocity. Sharp, Blair, and Cho (2001) proposed that the reciprocal connectivity between DTN and LMN could constitute an attractor network at the earliest stage of the HD system. This agrees with the observation that LMN cells have the largest ATI, AD cells anticipate less, and PoS cells, which are even further downstream, are hardly anticipatory at all.

The common mechanism in all bump attractor models is local excitation combined with global inhibition. Assuming for convenience that cells are arranged by preferred direction, each cell excites its nearby neighbors, as in Figure 2, with the strength of the excitatory connection falling off as a Gaussian function of the difference in their preferred directions. To prevent runaway excitation, HD cells also excite global inhibitory units (not shown), which provide uniform negative feedback onto the excitatory population. The stable states of this network consist of a bump of activation located somewhere on the ring. Noise in the network tends to be suppressed by the attractor dynamics, so the shape of the bump persists, and in the absence of external input, its location remains

fixed. In addition to accounting for the stability of HD tuning curves, this model also explains the observation of Knierim et al. (1995) that when the HD system drifts, all cells drift in unison: the difference in preferred directions of any two simultaneously recorded cells remains constant.

Applying an external input to a spot on the ring underlying one flank of an attractor bump causes the bump to shift until it is centered over the external input. This is the mechanism Skaggs et al. (1995) proposed to explain how visual landmarks exert control over the HD system. In their model, landmark feature detectors looking at specific egocentric bearings would learn excitatory connections onto the appropriate HD cells as the animal became familiar with its environment. For example, if while facing north the animal saw landmark X ahead and to its right, via Hebbian learning, a feature detector that responds to “X ahead and to the right” would develop excitatory connections onto active HD cells, whose preferred directions would be close to 360°. If the HD system later drifted as a result of accumulated integration error, when the animal attended to the scene, external input from feature detectors would pull the bump back into alignment with the environment.

The greater the magnitude of the external input applied to the flank of a bump, the faster the bump moves. This property can be used to integrate angular velocity if the magnitude of the external input scales with velocity. But if the external input is applied far from the peak, so that it does not lie on the flank of the bump, its effect will be suppressed by the attractor dynamics. If it is applied in equal amounts to both flanks of the bump, the bump will stay put.

Angular velocity integration in an attractor network therefore requires input from a population of velocity-sensitive cells that

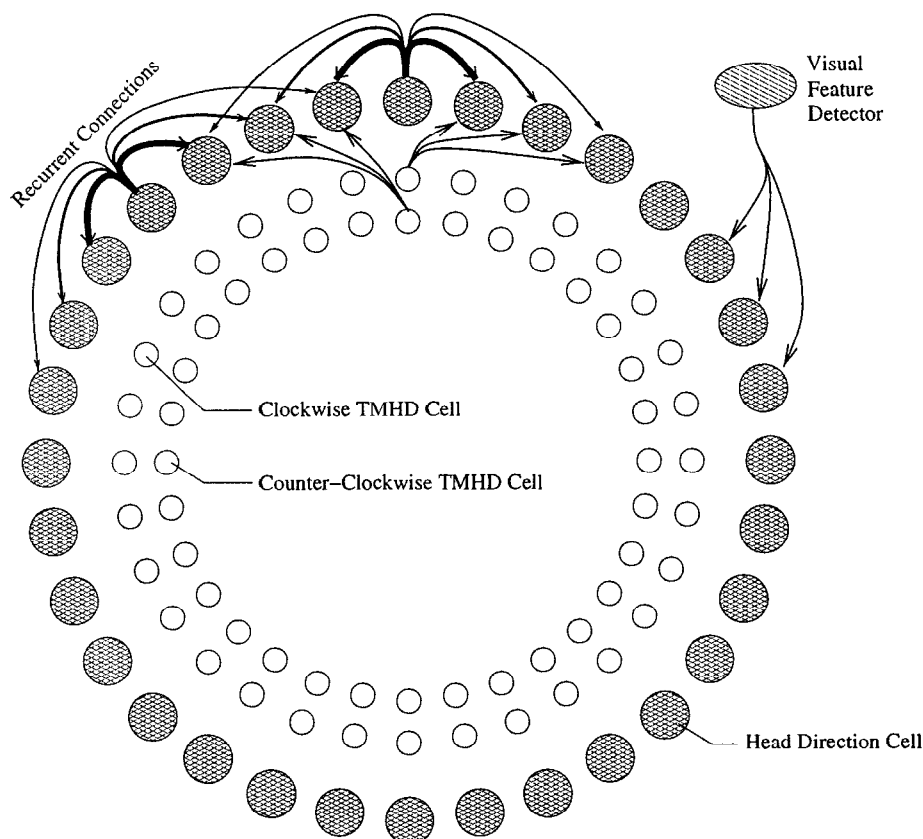


Figure 2. Architecture of an attractor-based head direction system model (after Skaggs et al., 1995).

exhibit HD tuning (so the input is focused on the correct flank of the attractor bump) and a turning preference. Blair and Sharp (1995) called these turn-modulated head direction (TMHD) cells. TMHD cells with a clockwise turning preference should make excitatory projections onto attractor HD cells with preferred directions slightly to the right of their own; TMHD cells with a counterclockwise turning preference should project to attractor HD cells slightly to the left. When the animal is still, both populations of TMHD cells should fire at the same rate, supplying identical input to both flanks of the attractor bump. When the animal is in a clockwise turn, the clockwise TMHD population should increase its firing rate, while the counterclockwise population should decrease its rate, in proportion to the angular velocity. Hence the bump would receive more input on its right flank, and begin shifting to the right. The opposite would occur for counterclockwise turns.

Cells in LMN seem to meet all the requirements for the hypothesized TMHD population: they show normal HD tuning, velocity modulation, and a turning preference. In the Goodridge and Touretzky (2000) model, the clockwise and counterclockwise LMN populations both project to AD. Since AD does not appear to have recurrent connections, it cannot function as an attractor network, and thus cannot exhibit all the stability properties of such a network. But AD HD cells integrate information from LMN, PoS, and perhaps retrosplenial HD cells, and AD projects to PoS. The attractor network is located in PoS, which does have recurrent connections. This model explains why the firing of PoS HD cells is independent of angular velocity, while AD HD cells are influenced by velocity.

In a complementary attractor model by Sharp et al. (2001), DTN cells make inhibitory projections onto LMN cells, which in turn make excitatory projections back onto the DTN population. DTN cells also receive angular velocity information from vestibular nuclei. HD cells in LMN and DTN have different tuning properties, but their interaction is hypothesized to produce the attractor dynamics required to integrate angular velocity.

HD cells in other areas, such as retrosplenial cortex and caudate nucleus, could also contribute to the functioning of the “core” HD system. Further work may explain why there are so many HD areas, and why the difference in ATI values for PoS, AD, and LMN cells is so much greater than can be accounted for by synaptic delays alone.

Road Map: Mammalian Motor Control

Related Reading: Dynamic Remapping; Hippocampus: Spatial Models; Vestibulo-Ocular Reflex

References

- Bassett, J. P., and Taube, J. S., 2001, Neural correlates for angular head velocity in the rat dorsal tegmental nucleus, *J. Neurosci.*, 21:5741–5751.
- Blair, H. T., Cho, J., and Sharp, P. E., 1998, Role of the lateral mammillary nucleus in the rat head-direction circuit: A combined single-unit recording and lesion study, *Neuron*, 21:1387–1397.
- Blair, H. T., and Sharp, P. E., 1995, Anticipatory head direction signals in anterior thalamus: Evidence for a thalamocortical circuit that integrates angular head motion to compute head direction, *J. Neurosci.*, 15:6260–6270.
- Goodridge, J. P., Dudchenko, P. A., Worboys, K. A., Golob, E. J., and Taube, J. S., 1998, Cue control and head direction cells, *Behav. Neurosci.*, 112(4):1–13.
- Goodridge, J. P., and Touretzky, D. S., 2000, Modeling attractor deformation in the rodent head-direction system, *J. Neurophys.*, 83:3402–3410. ♦
- Knierim, J. J., Kudrimoti, H. S., and McNaughton, B. L., 1995, Place cells, head direction cells, and the learning of landmark stability, *J. Neurosci.*, 15:1649–1659.
- Knierim, J. J., Kudrimoti, H. S., and McNaughton, B. L., 1998, Interactions between idiothetic cues and external landmarks in the control of place cells and head direction cells, *J. Neurophys.*, 80:425–446.
- Mizumori, S. J. Y., and Williams, J. D., 1993, Directionally selective mnemonic properties of neurons in the lateral dorsal nucleus of the thalamus of rats, *J. Neurosci.*, 13:4015–4028.
- Sharp, P. E., Blair, H. T., and Cho, J., 2001, The anatomical and computational basis of the rat head-direction cell signal. *Trends Neurosci.*, 24:289–294. ♦
- Sharp, P. E., Tinkelman, A., and Cho, J., 2001, Angular velocity and head direction cells recorded from the dorsal tegmental nucleus of Gudden in the rat: Implications for path integration in the head direction cell circuit. *Behav. Neurosci.*, 115:571–588.
- Skaggs, W. E., Knierim, J. J., Kudrimoti, H. S., and McNaughton, B. L., 1995, A model of the neural basis of the rat's sense of direction in *Advances in Neural Information Processing Systems 7* (G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds.), Cambridge, MA: MIT Press, pp 173–180.
- Stackman, R. U., and Taube, J. S., 1998, Firing properties of rat lateral mammillary single units: Head direction, head pitch, and angular head velocity, *J. Neurosci.*, 18:9020–9037.
- Taube, J. S., 1995, Head direction cells recorded in the anterior thalamic nuclei of freely moving rats. *J. Neurosci.*, 15:1953–1971.
- Taube, J. S., Muller, R. U., and Ranck, J. B., Jr., 1990a, Head direction cells recorded from the postsubiculum in freely moving rats: I. Description and quantitative analysis, *J. Neurosci.*, 10:420–435.
- Taube, J. S., Muller, R. U., and Ranck, J. B., Jr., 1990b, Head direction cells recorded from the postsubiculum in freely moving rats: II. Effects of environmental manipulations, *J. Neurosci.*, 10:436–447.
- Zhang, K., 1996, Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory, *J. Neurosci.*, 16:2112–2126. ♦

Schema Theory

Michael A. Arbib

Introduction

Schema theory complements neuroscience's well-established terminology for levels of *structural* analysis (brain region, neuron, synapse) with a *functional* vocabulary, a framework for analysis of behavior with no necessary commitment to hypotheses on the localization of each *schema* (unit of functional analysis), but which can be linked to a structural analysis whenever appropriate. Schemas provide a high-level vocabulary that can be shared by brain theorists, cognitive scientists, connectionists, ethologists, and even

kinesiologists, even though the implementation of the schemas may differ from domain to domain. This article presents a general perspective, notes but does not emphasize learning models for schemas, and focuses on two issues: structuring perceptual and motor schemas to provide an action-oriented account of behavior and cognition (as relevant to the roboticist as the ethologist); and how schemas describing animal behavior may be mapped to interacting regions of the brain. Schema-based modeling becomes part of neuroscience when constrained by data provided by, for example, human brain mapping, studies of the effects of brain lesions, or neu-

rophysiology. The resulting model may constitute an adequate explanation in itself or may provide the framework for modeling at the level of neural networks or below. Such a *neural* schema theory provides a functional/structural decomposition, in strong contrast with models that employ learning rules to train a single, otherwise undifferentiated, neural network to respond as specified by some training set.

Schemas: History and Comparisons

Central to our approach is the notion of the “active organism,” which seeks from the world the information it needs to pursue its chosen course of action. In *action-oriented perception*, current sensory input is itself a function of the subject’s active exploration of the world, which is directed by *anticipatory schemas*, which Neisser (1976) defines as plans for perceptual action as well as readiness for particular kinds of sensory structure. This view has resonances with that of Piaget (1971, pp. 6–7): “Any piece of knowledge is connected with an action . . . [T]o know an object or a happening is to make use of it by assimilation into an action schema . . . [namely] whatever there is in common between various repetitions or superpositions of the same action.” Acting on the basis of an action schema usually entails the *expectation* of certain consequences. Piaget talks of *assimilation*, the ability to make sense of a situation in terms of a stock of schemas, and *accommodation*, whereby the stock of schemas may change over time as the expectations based on assimilation to current schemas are not met. Piaget traces the cognitive development of the child from reflexive schemas through eye-hand coordination and object permanence all the way to schemas for language and abstract thought.

Head and Holmes introduced the term schema to neurology in 1911, speaking of the *body schema* (Frederiks, 1969, reviews relevant literature): “Anything which participates in the conscious movement of our bodies is added to the model of ourselves and becomes part of those schemata: a woman’s power of localization may extend to the feather of her hat.” A woman with unilateral damage to the parietal lobe may lose awareness that the body on the opposite side actually belongs to her—not only ignoring painful stimuli but even neglecting to dress that half of the body. Damage to the thalamus and the somatosensory system may also produce disorders of the body schema.

Bartlett (1932) carried the schema idea into cognitive psychology, with a schema being “an active organization of past reactions [or] experiences, which must always be supposed to be operating in any well-adapted organic response.” He stressed the constructive character of remembering. When people try to recall a story, they reconstitute it in their own terms—relating what they experience to a familiar set of schemas—rather than by rote memorization of details. Instead of thinking of ideas as impressions of sense data, schema theory posits an active and selective process of schema formation (cf. Piaget’s notion of assimilation), which in some sense constructs reality as much as it embodies it. More generally, cognitive psychology views schemas as cognitive structures built up in the course of interaction with the environment to organize experience. Not only is sensory input coded by instantiating certain schemas (we say a schema is instantiated when active copies are running, and refer to these copies as “schema instances”), as seeing a chair instantiates an instance of the “chair schema,” but the current stock of schema instances may also instantiate related action schemas such as “sitting” and general schemas such as “furniture” while inhibiting other competing schemas. Shallice (1988, p. 308n) stresses that the schema “not only has the function of being an efficient description of a state of affairs—as in, say, Bartlett’s usage—but also is held to produce an output that provides the immediate control of the mechanisms required

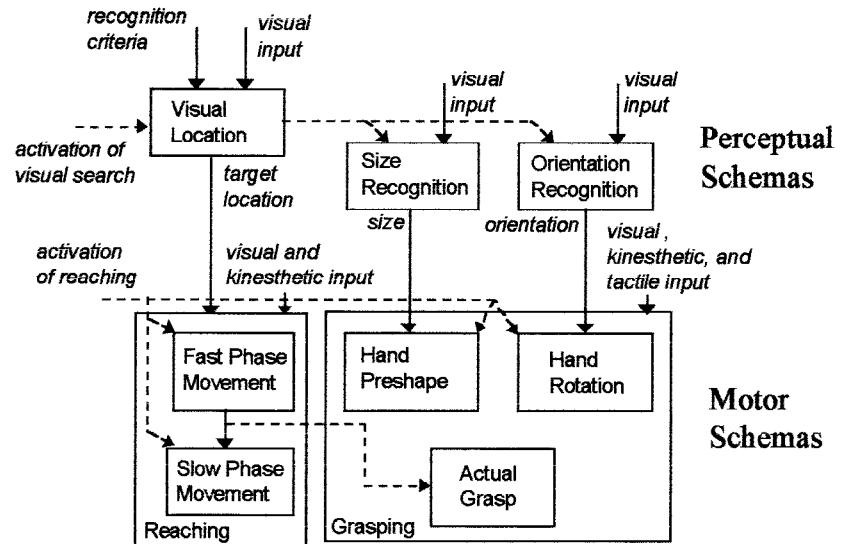
in one cognitive or action operation. The usage is thus more analogous to Piaget’s view than to Bartlett’s original concept.” In a connectionist vein, Rumelhart et al. (1986) suggest how schemas may be seen as emergent properties of adaptive, connectionist networks, but they neither relate schemas to action nor show how schemas may be combined to form assemblages (see discussion later in this article).

Schmidt (1976) offered a schema theory of discrete motor skill learning. Through experience, the subject builds up a *recall schema* that pairs the response specifications of a movement with the actual outcome. Later, this recall schema can be consulted to infer, from a desired outcome, the response specification that will produce it. Similarly, a *recognition schema* pairs the desired outcome with the expected sensory consequences of each movement. The recall schema is what is now known in the literature of motor control as an “inverse model” (SENSORIMOTOR LEARNING)—it goes from a desired response to a pattern of commands that achieves it, rather than the “direct” causal path from commands to action; while the recognition schema corresponds to Neisser’s anticipatory schema.

Arbib (1981; Arbib, Érdi, and Szentágothai, 1998, Chapter 3 for an exposition) offered a schema theory more tightly constrained by the need to explain the neural basis of behavior, stressing that a schema expresses a function that need not be co-extensive with the activity of a single neuronal circuit. (This view was foreshadowed in the work of Kilmer, McCulloch, and Blum (1969) who posed the general question of how the nervous system could set the organism’s “overall mode of behavior” through the cooperative computation [no executive control] of modules, each of which aggregates the activity of many neurons.) A *schema* is what is learned about some aspect of the world, combining knowledge with the processes for applying it; a *schema instance* is an active deployment of these processes. A *perceptual schema* not only determines whether a given “domain of interaction” (an action-oriented generalization of the notion of object) is present in the environment, but can also provide parameters concerning the current relationship of the organism with that domain. Each schema instance has an *activity level* that indicates its current salience for the ongoing computation. The activity level of a perceptual schema signals the credibility of the hypothesis that what the schema represents is indeed present, whereas other schema parameters represent other salient properties such as size, location, and motion of the perceived object. Given a perceptual schema, we may need several schema instances, each suitably tuned, to subserve our perception of several instances of its domain. *Motor schemas* provide the control systems that can be coordinated to effect a wide variety of actions. The *activity level* of a motor schema instance may signal its “degree of readiness” to control some course of action (thus enriching somewhat the related notion of motor pattern generators; see MOTOR PATTERN GENERATION).

Schema instances may be combined (possibly with those of more abstract schemas, including coordinating schemas) to form *schema assemblages*. For example, an assemblage of perceptual schema instances may combine an estimate of environmental state with a representation of goals and needs. A *coordinated control program* is a schema assemblage that processes input via perceptual schemas and delivers its output via motor schemas, interweaving the activations of these schemas in accordance with the current task and sensory environment to mediate more complex behaviors. Figure 1 shows the original coordinated control program. As the hand moves to grasp an object, it is *preshaped* so that when it has almost reached the ball, it is of the right shape and orientation to enclose some part of the object prior to gripping it firmly. Moreover (to a first approximation), the movement can be broken into a fast initial movement and a slow approach movement, with the transition from the fast to the slow phase of trans-

Figure 1. Hypothetical coordinated control program for reaching and grasping. Note that different perceptual schemas (at the top) are required to provide parameters for the motor schemas (at the bottom) for the control of “reaching” (arm transport \approx hand reaching) and “grasping” (controlling the hand to conform to the object). Note too the timing relations posited here within the “Hand Reaching” motor schema and between the motor schemas for “Reaching” and “Grasping.” Dashed lines—activation signals; solid lines—transfer of data. (Adapted from Arbib, 1981.)



port coming just before closing of the fingers from the preshape so that touch may take over in controlling the final grasp. The top half of Figure 1 shows three perceptual schemas: successful location of the object activates the schemas for recognizing the size and orientation of the object. The outputs of these perceptual schemas are available for the control of the hand movement by concurrent activation of two motor schemas, one controlling the arm to transport the hand toward the object and the other preshaping the hand, with finger separation and orientation guided by the output of the appropriate perceptual schemas. Once the hand is preshaped, it is only the completion of the fast phase of hand transport that “wakes up” the final stage of the grasping schema to shape the fingers under control of tactile feedback. (This model anticipates the much later discovery of perceptual schemas for grasping in a localized area [AIP] of parietal cortex and motor schemas for grasping in a localized area [F5] of premotor cortex. See GRASPING MOVEMENTS: VISUOMOTOR TRANSFORMATIONS.)

Neuroscience and cognitive psychology often view working memory as storing a single item (e.g., the location of a target, or a single phone number) for a short delay period between observation of the item and its use in some action, after which it is discarded. Here, we extend the notion to insist that working memory may hold a range of items relevant to upcoming actions, and these items may remain accessible for extended periods so long as they remain relevant. Schema-based modeling of action-oriented perception (VISUAL SCENE PERCEPTION) then views the *short-term memory* (STM) of an organism as a working memory that combines the schema instances encoding relevant aspects of, and plans for interaction with, the current environment. This assemblage is dynamic, as certain schema instances are discarded from memory (“de-instantiated”) while others are added (“instantiated”). *Long-term memory* (LTM) is provided by the stock of schemas from which STM may be assembled. New sensory input as well as internal processes can update STM. The internal state is also updated by knowledge of the state of execution of current *plans* that specify a variety of coordinated control programs for possible execution. Jeannerod (1997) surveys the role of schemas and other constructs in the cognitive neuroscience of action.

Schemas for *Rana Computatrix*

A schema model becomes a neural model, as distinct from a purely functional model, when explicit hypotheses are offered as

to how the constituent schemas are played over particular regions of the brain. To exemplify this, consider approach and avoidance in the frog (VISUOMOTOR COORDINATION IN FROG AND TOAD; and see Arkin et al., 2000, for a related discussion of behavioral models of the praying mantis as a basis for robotic behavior). A frog surrounded by dead flies will starve to death, but the frog will snap with equal “enthusiasm” at a moving fly or a pencil tip wiggled in a fly-like way. On the other hand, a larger moving object can trigger an escape reaction. Thus, a highly simplified model of the functioning of the brain of the frog has signals from the eye routed to two basic perceptual schemas, one for recognizing small moving objects (foodlike stimuli) and one for recognizing large moving objects (enemylike stimuli). If the small-moving-object schema is activated, it will in turn trigger the motor schema that gets the animal to approach what is apparently its prey. If the perceptual schema for large-moving-object is activated, it will trigger the motor schema for avoidance, causing the animal to escape an apparent enemy.

The biological model relates these schemas to anatomy. Each eye of the frog projects to regions on the opposite side of the brain, including the important visual midbrain regions called the *tectum* and the *pretectum* (in front of the tectum). If we hypothesize that the small-moving-object schema is in the tectum, while the large-moving-object schema is in the pretectum, the model (Figure 2A) predicts that animals with a pretectal lesion would approach small moving objects, but would not respond at all to large moving objects. However, Peter Ewert in Kassell studied toads with the pretectum removed and found that they responded with approach behavior to both large and small moving objects. This observation leads to the new schema-level model shown in Figure 2B. We replace the perceptual schema for *small* moving objects of Figure 2A by a perceptual schema for *all* moving objects and leave the right-hand column the way it was. The inhibitory pathway from the large-moving-object perceptual schema (in the pretectum) to the all-moving-object schema ensures that the model yields the normal animal’s response to small moving objects with approach but not avoidance. This model explains our small database on the behavior of both normal animals and those with a lesion of the pretectum.

We have thus shown how hypotheses about neural localization of subschemas may be tested and refined by lesion experiments. The important point is that models expressed at the level of a network of interacting schemas can really be testable biological

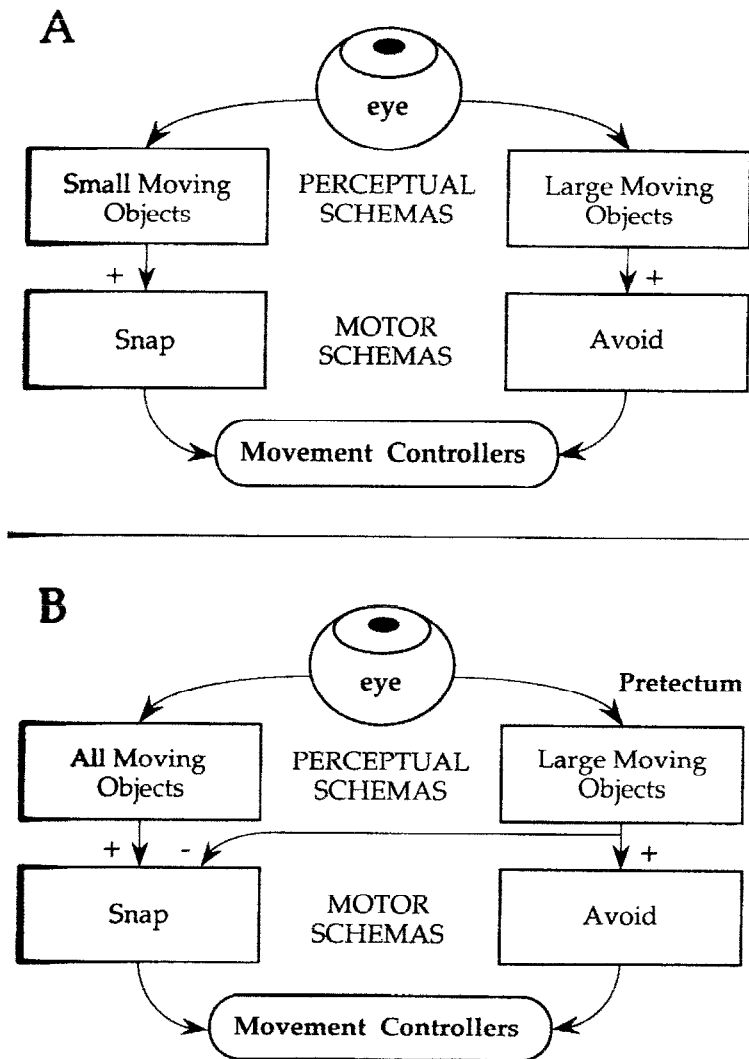


Figure 2. A, A "naive" schema program that represents the perceptual and motor schemas for frog approach behavior (snap at small moving objects) as completely separated from those for avoidance. B, A schema program for approach and avoidance that takes into account data on the effect of lesioning the pretectum. In particular, the "approach schema" is *not* localized in the tectum alone since it depends on pretectal inhibition for its integrity.

models. Subsequent work has extended work on *Rana computatrix* ("the frog that computes"; Arbib et al., 1998, for a partial review) at the level of both schemas and neural networks for phenomena such as detours and path planning, avoidance behavior sensitive to the trajectories of predators, and details of snapping behavior that link neural control to biomechanics. The work constitutes a grounding example of work in modeling neural mechanisms in overall animal behavior (NEUROETHOLOGY, COMPUTATIONAL).

Coordinated Control Programs and Motor Schemas

We have seen that schemas can be combined to form *coordinated control programs* that control the phasing in and out of patterns of schema co-activation and the passing of control parameters from perceptual to motor schemas. The notion of schema is thus *recursive*—a schema defined functionally may later be analyzed as a coordinated control program of finer schemas, and so on until such time as a secure foundation of neural localization is attained. The model of Figure 1 distinguished two phases of arm movement—a fast phase controlled by a ballistic schema (i.e., one that moves rapidly to completion, unaffected by feedback), followed by a slow phase controlled by a schema that does admit error-correction by use of sensory feedback. However, Jeannerod et al.

(1992) showed that reaching is subject to modification by sensory input even during the fast phase. If the target of a pointing task was perturbed at movement onset, the subject did not complete a ballistic movement toward the initial target before moving on to the new target. Rather, a smooth adjustment was made about 100 msec after target perturbation to a new trajectory terminating at the new target, without loss of accuracy. To address such data (and more), Hoff and Arbib (1993) extended the use of OPTIMIZATION PRINCIPLES IN MOTOR CONTROL (q.v.) by showing how to embed an optimality principle for arm trajectories into a controller that can use feedback to resist noise and compensate for target perturbations, and a predictor element to compensate for delays from the periphery. The result is a feedback system that can *act like* a feedforward system described by the optimality principle in "familiar" situations, where the conditions of the desired behavior are not perturbed and accuracy requirements are such that "normal" errors in execution may be ignored. However, when perturbations must be corrected for or when great precision is required, feedback plays a crucial role in keeping the behavior close to that desired, taking account of delays in putting feedback into effect.

Another claim embodied in Figure 1 is that the transition from preshaping to enclosing is controlled by, but does not influence,

the transition in the transport phase. However, data show that when the hand has unexpectedly to open wider (if object size is increased during reach) transport slows by about 200 msec, but if target location is perturbed, the hand temporarily closes so that maximum aperture is delayed as transport takes longer to reach the new target. Hoff & Arbib (1993) designed a controller for the preshape schema to tradeoff an optimality criterion needed to prevent discontinuous “jumps” in the preshape and a “cost” for having the hand open more than a certain amount. The latter yields the partial reclosing of the hand during prolonged movement caused by location perturbation. Their strategy for coordinating the motor schemas is set forth in Figure 3. Here, the Enclose schema is a replica of the Preshape schema with the only exception that its starting point is the maximum aperture achieved by the preshape schema (there seems to be a linear relation between the actual object size and the maximum aperture achieved). The coordinating schema receives from each of the constituent schemas—Transport, Preshape, Enclose—an estimate of the time that it needs to move for execution. Then, whichever schema is going to take longer, Transport or Grasp (Preshape + Enclose), is given the full time it needs, while the other schema will be slowed down to apply its optimality criterion over the longer time base. This yields a satisfactory match between data and simulation.

The implication (a truth better known in motor control than in other areas of neurophysiology) is that much is to be learned at the level of schema analysis prior to, or in concert with, the “lower level” analysis of neural circuitry. Although the hypotheses developed in this section allow us to gain insight into the interaction of a number of different processes, they also pose major challenges for further neurophysiological investigation.

Discussion

Perceptual and Motor Schemas

We have suggested that schemas provide some sort of action-oriented memory, yet have made a distinction between perceptual schemas and motor schemas. Why not combine these two con-

structs into a single notion of schema that integrates sensory analysis with motor control, as suggested in the earlier quote from Shallice? Indeed, there are cases in which such a combination makes sense. However, recognizing an object (an apple, say) may be linked to many different courses of action (to place the apple in one's shopping basket; to place the apple in the bowl at home; to pick up the apple; to peel the apple; to cook with the apple; to eat the apple; to discard a rotten apple, etc.). Of course, once one has decided on a particular course of action, then specific perceptual and motor subschemas may be invoked. But note that, in the list just given, some items are apple-specific whereas other invoke generic schemas for reaching and grasping. It was considerations like this that led me to separate perceptual and motor schemas—a given action may be invoked in a wide variety of circumstances; a given perception may, as part of a larger assemblage, precede many courses of action. Putting it another way, there is no one “grand apple schema” that links all “apple perception strategies” to “every act that involves an apple.” At the same time, however, note that, in the schema-theoretic approach, “apple perception” is not mere categorization—“this is an apple”—but may provide access to a range of parameters relevant to interaction with the apple at hand. The *Rana* example shows this in simplest form. In Figure 2A, the two schemas at left may be combined into a single unitary prey schema and the two at right into a single unitary predator schema. However, the lesion study suggested splitting perception from action since it is recognition of the large moving object that inhibits the prey-catching schema—based on the view that tectum and pretectum are “more perceptual” and the brainstem to which they project is “more motor.”

A detailed example of how schema theory extends to more “cognitive” realms than basic patterns of sensorimotor coordination is offered by the schema-based interpretation in the VISIONS computer vision system (see VISUAL SCENE PERCEPTION for references). This system shows the importance of schema theory within artificial intelligence (i.e., even when there is no claim to model the brain). Similarly, schemas have played an important role in the development of behavior-based robots (REACTIVE ROBOTIC SYSTEMS).

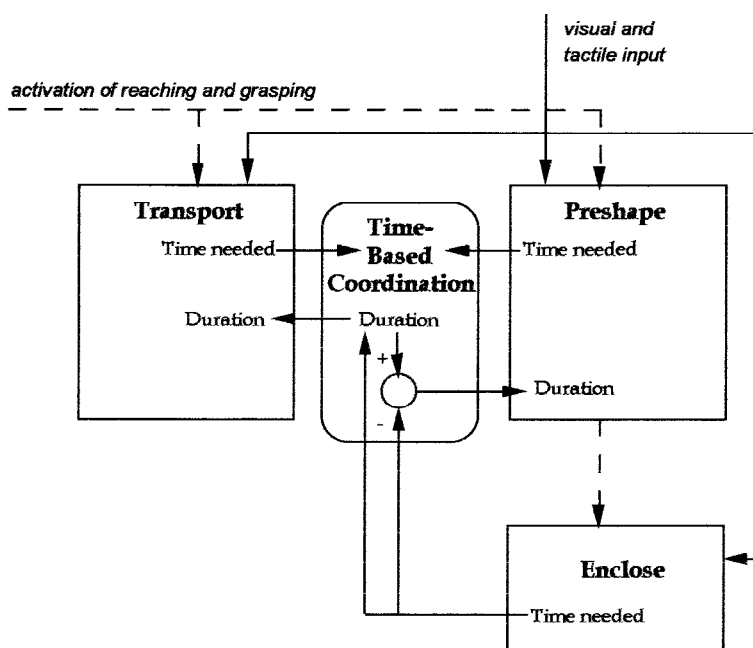


Figure 3. Feedback controllers for transport and preshape. “Cooperative computation” between subprograms is mediated by a coordinating schema ensuring that both reaching and grasping have adequate movement time (Hoff and Arbib, 1993).

Schemas and Their Assemblages Are Adaptable

Head himself considered schemas as plastic entities, which are subject to constant change, and adaptation is at the heart of Piaget's account of assimilation and accommodation. Although the examples of schemas given in the three figures above are fairly stable—as we explored the way in which schemas could be combined into coordinated control programs, and ways in which the psychological or neural correlates of such schema assemblages could be tested—it must be emphasized that schemas, and their connections with each other, change through the processes of accommodation. These processes adjust the network of schemas so that over time they may well be able to better handle a wide range of situations.

Work on HYBRID CONNECTIONIST/SYMBOLIC SYSTEMS (q.v.; see also Sun, 1995) has somewhat similar motivations to schema theory: In decomposing a cognitive model into a network of interacting processes, one may find at a given state of knowledge that quite different models will be appropriate for different components: some cognitive processes are best captured by symbolic models, some by connectionist models, and some by biologically realistic neural models. This leads to the development of hybrid systems. Schema theory is consistent with this in that it allows the schemas in an “assemblage” to be implemented in different ways so long as the input and output codes are compatible on any connection. However, it adds that what may appear to be disjoint schemas when implemented at one level may turn out to involve overlapping networks of subschemas when involved at a more detailed level (a simple example is given in Figure 2B). On the other hand, if we agree to the schema decomposition offered by high-level analysis, we may apply connectionist training procedures to adapt the initial schema structures if these are encoded by (artificial) neural networks.

Beyond Action and Perception

Through learning, a complex schema network arises that can mediate first the child's, and then the adult's, reality. Through being rooted in such a network, schemas are interdependent, so that each finds meaning only in relation to others. For example, a house is defined in terms of parts such as a roof, yet a roof may be recognized because it is part of a house that is recognized on the basis of other criteria such as “people live there.” Each schema enriches and is defined by the others (and may change when a formal linguistic system allows explicit, though partial, definition). Though processes of schema change may affect only a few schemas at any time, such changes may “cohere” to yield dramatic changes in the overall pattern of mental organization. There is change yet continuity, with many schemas held in common, yet changed because they must now be used in the context of the new network. Arbib and Hesse (1986) offer an epistemology rooted in this view of schema theory, and show how it may be expanded to link “schemas in the head” with the “social schemas” that form the collective representations (to use Durkheim's phrase) shared by a community. Schemas have also been used in a “computational, neo-Piagetian” approach to language

acquisition and may also be used in modeling language processing with special attention to the links between action, action and object recognition, and language (see LANGUAGE EVOLUTION: THE MIRROR SYSTEM HYPOTHESIS) and in relating these processes to neural schemas.

Road Maps: Artificial Intelligence; Psychology

Related Reading: Artificial Intelligence and Neural Networks; Compositionality in Neural Systems; Hybrid Connectionist/Symbolic Systems; Multiagent Systems

References

- Arbib, M. A., 1981, Perceptual structures and distributed motor control, in *Handbook of Physiology—The Nervous System II. Motor Control* (V. B. Brooks, Ed.), Bethesda, MD: American Physiological Society, pp. 1449–1480.
- Arbib, M. A., Érdi, P. and Szentágothai, J., 1998, *Neural Organization: Structure, Function, and Dynamics*, Cambridge, MA: MIT Press (see Chapter 3). ♦
- Arbib, M. A., and Hesse, M. B., 1986, *The Construction of Reality*, Cambridge University Press.
- Arkin, R. C., Ali, K., Weitzenfeld, A., and Cervantes-Pérez, F., 2000, Behavioral models of the praying mantis as a basis for robotic behavior, *Robotics and Autonomous Systems*, 32:39–60. ♦
- Bartlett, F. C., 1932, *Remembering*, Cambridge University Press.
- Frederiks, J. A. M., 1969, Disorders of the body schema, in *Handbook of Clinical Neurology*, 4, *Disorders of Speech Perception and Symbolic Behavior*, (P.J. Vinken and G.W. Bruyn, Eds.), North Holland, pp. 207–240. ♦
- Hoff, B., and Arbib, M. A., 1993, Simulation of interaction of hand transport and preshape during visually guided reaching to perturbed targets, *J. Motor Behav.*, 25:175–192.
- Jeannerod, M., 1997, *The Cognitive Neuroscience of Action*, Oxford, UK: Blackwell Publishers.
- Jeannerod, M., Paulignan, Y., MacKenzie, C., and Marteniuk, R., 1992, Parallel visuomotor processing in human prehension movements, in *Control of Arm Movement in Space* (R. Caminiti, P. B. Johnson, and Y. Burnod, Eds.), *Exp. Brain. Res. Ser.*, 22:27–44.
- Kilmer, W. L., McCulloch, W. S., and Blum, J., 1969, A model of the vertebrate central command system, *Int. J. Man-Machine Stud.*, 1:279–309.
- Neisser, U., 1976, *Cognition and Reality: Principles and Implications of Cognitive Psychology*, San Francisco: W.H. Freeman.
- Piaget, J., 1971, *Biology and Knowledge*, Edinburgh University Press, Edinburgh.
- Rumelhart, D. E., Smolensky, P., McClelland, J. L., and Hinton, G. E., 1986, Schemata and sequential thought processes in PDP models, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* vol. 2 (J. L. McClelland and D. E. Rumelhart, Eds.), Cambridge, MA: MIT Press, Chapter 14.
- Schmidt, R. A., 1976, The schema as a solution to some persistent problems in motor learning theory, in *Motor Control: Issues and Trends* (G. E. Stelmach, Ed.), New York: Academic Press, pp. 41–65.
- Shallice, T., 1988, *From Neuropsychology to Mental Structure*, Cambridge, MA: Cambridge University Press.
- Sun, R., 1995, On schemas, logics, and neural assemblies, *Applied Intelligence*, 5(2):83–102.

Scratch Reflex

Paul S. G. Stein

Introduction

An organism can scratch itself in response to a stimulus at a site on the body surface (Stein, 1983). Mechanical force is the stimulus that has been used in many studies of scratching. In frogs, chemical irritation has also been used to activate scratching. During successful scratching, a nearby limb moves toward and rubs against the site. If a different site is stimulated, different limb movements are required for a successful scratch. The stimulated site may be on a part of the body such as the shell of a turtle (Stein, 1989) or the back of a frog (Berkinblit, Feldman, and Fukson, 1989), or it may be on a part that can move with respect to the body such as the ear of a cat (Kuhta and Smith, 1990) or the elbow of a frog (see Stein, 1983). In each case, the organism can generate a successful scratch. (Since space in the reference list is limited, the reader is referred to the articles by Stein, 1989, Stein and Smith 1997, and other papers by the author for fuller references to the literature.)

Some organisms do not require the entire central nervous system to produce a successful scratch reflex. Some vertebrates with a complete spinal cord transection at the level of the neck or upper back, termed *spinal vertebrates*, can perform a successful hindlimb scratch in response to a stimulus delivered to a site on the body surface posterior to the complete transection. This stimulus excites neural networks in the spinal cord posterior to the complete transection (Stein, 1983, 1989). Scratching has been demonstrated in several spinal vertebrates: dog (Sherrington, 1906), cat (Arshavsky, Gelfand, and Orlovsky, 1986; Orlovsky, Deliagina, and Grillner, 1999), turtle (Stein, Mortin, and Robertson, 1986; Stein, 1989; Earhart and Stein, 2000), and frog. In frog, scratch reflex is also termed *wiping reflex* (Berkinblit et al., 1989; see also MOTOR PRIMITIVES).

Site specificity of scratch reflex in spinal vertebrates is impressive. The spinal frog's hindlimb can wipe off an acid-soaked piece of paper from its elbow. Successful wipes of the elbow by the hindlimb occur when the elbow is close to the neck as well as when the elbow is close to the midbody (see Stein, 1983). Spinal cord neuronal networks can generate complex sensorimotor transformations even when disconnected from supraspinal structures.

Strategies of Scratching: The Forms of a Scratch

The set of all successful scratches is constrained by the physical construction of the organism's limbs and body, i.e., its biomechanics. For some organisms, there may be a set of sites on the body surface, e.g., sites on the middle of the back of a turtle or a human, that cannot be rubbed directly by a limb of that organism. There may be some sites that can be scratched by using only one strategy of movement; for example, a human can scratch some sites on the upper back using only a strategy in which the elbow is placed over the shoulder. These sites belong to a set termed a *pure-form domain*. While the same motor strategy is used to rub against each site within a pure-form domain, parametric adjustment of limb movement is required to reach each specific site. There may be other sites that can be scratched by using either of several strategies; for example, a human can scratch a site on the side of the thorax using either the hand or the elbow. These sites belong to a set termed a *transition zone*. Each scratch movement strategy is termed a *form* of the scratch (Stein, Mortin, and Robertson, 1986). The concept of movement form can be applied to other motor acts. There are several forms of locomotion in mammals, e.g., walk, trot, and gallop (Stein and Smith, 1997; see also GAIT TRANSITIONS). Forward stepping and backward stepping are among the forms of stepping that humans produce.

The spinal turtle produces several scratch strategies. In each strategy, a distinct portion of the hindlimb is used to exert force against the stimulated site (Stein, Mortin, and Robertson, 1986). The turtle uses the dorsum of the foot for *rostral scratching*, the side of the knee for *pocket scratching*, and the side of the foot or heel for *caudal scratching*. Biomechanical constraints play a key role for each of these movement strategies. The rostral scratch is the only strategy that can be used to place a portion of the hindlimb against a site on the region that connects the upper shell and the lower shell in the middle of the body. The foot cannot reach sites in the pocket region just anterior to the turtle's hip; only the side of the knee can be used to generate force against a site in the pocket region. Thus, the spinal turtle can select the biomechanically appropriate form that produces successful scratches; selection of the proper scratch strategy does not require supraspinal structures, i.e., the brainstem and the brain. Experiments described in a later section establish form selection as an intrinsic property of spinal cord neural networks.

Coordinate System Transformations in the Scratch Reflex

Several transformations occur during scratching. First, a stimulus-to-sensory transformation takes place on the body surface when the stimulus activates primary afferent neurons. Second, a sensory-to-motor transformation takes place within the central nervous system (CNS). Third, a motor-to-mechanical transformation takes place in the limb. It is possible to examine all three transformations at the same time during actual scratching. For neuronal network studies, it is useful to examine the first and second transformations in the absence of the third transformation, i.e., in the absence of movement. Experiments that examine the response to a tactile stimulus that elicits scratch motor output while neuromuscular synapses are blocked are described in later sections.

Several coordinate systems help to describe the different transformations. First, the rectilinear orthogonal Cartesian coordinate system is useful in a scientist's description of the site on the body surface that receives the sensory stimulus as well as the position in space of the limb that rubs against the site. Second, the muscle/motor-pool coordinate system is useful for describing the output of the CNS. The set of motor neurons that synaptically activate a given muscle is termed a motor pool. Each muscle of the body and/or its motor pool can be viewed as a dimension of a coordinate system. Each dimension's amplitude is determined by the intensity of activation of each muscle/motor-pool (see MOTONEURON RECRUITMENT). During each movement, there is a distinct *motor pattern* of muscle/motor-pool activation that occupies a region of an abstract space whose dimensions are time and muscles/motor-pools. Third, the body degree-of-freedom, also termed joint-angle, coordinate system is useful for describing multijointed limb movements. Some body joints, e.g., the knee, have a single degree of freedom and involve only a single dimension. Other body joints, e.g., the hip, have several degrees of freedom and therefore require several dimensions. Each movement form has a distinct trajectory in time and degree-of-freedom space.

Coordinate System Analyses of the Turtle Scratch Reflex

Data obtained from studies of the scratch reflex in the spinal turtle (Stein, 1989) allow application of the concepts outlined in the pre-

vious section. These data are described in this section. Other data obtained from frog (Berkinblit et al., 1989; see also MOTOR PRIMITIVES) and from cat (Arshavsky et al., 1986) are also consistent with these concepts.

Cartesian Coordinate Description of Receptive Fields

If a stimulus applied to a site on the body surface elicits a specific form of scratch reflex in which a nearby limb reaches toward and rubs against the stimulated site, then that site is within the receptive field for that form of scratch. In the spinal turtle, there is a receptive field for rostral scratch, a receptive field for pocket scratch, and a receptive field for caudal scratch. There is also a rostral-pocket transition zone and a pocket-caudal transition zone (Stein et al., 1986).

Stimulation of most sites in one form's receptive field elicits only scratches of that form; these sites constitute the pure-form domain of that form's receptive field. The *pure-form domain* for each scratch form is the set of sites in which only one scratch form is biomechanically possible. There is also a set of transition-zone sites located between the pure-form domain for one form and the pure-form domain for another form. The *transition zone* is the set of sites in which more than one scratch form is biomechanically possible. Stimulation of a site in this transition zone can elicit either one scratch form, the other scratch form, or a blended response of both scratch forms. There are two types of blends: the switch response and the hybrid response. In a *switch response*, several cycles of one form are followed smoothly by several cycles of the other form. In a *hybrid response*, each of several successive cycles has two rubs per cycle; one rub utilizes one scratch form, and the other rub utilizes the other scratch form.

The occurrence of blends supports the notion that there is shared neural circuitry between the neural network that generates one form of scratch and the neural network that generates another form of scratch. In particular, analyses of blends support the concept that interneurons controlling the rhythm of hip movements are shared among the networks responsible for each of several forms of scratching (Berkowitz and Stein, 1994; Stein et al., 1995; Stein, McCullough, and Currie, 1998; Berkowitz, 2001).

Muscle/Motor-Pool Coordinate Description of Motor Patterns

The electromyographic (EMG) activity of individual muscles that play critical roles for each scratch form may be recorded during scratching in the spinal turtle (Earhart and Stein, 2000). The monoarticular knee extensor muscle is active during the rub against the stimulated site for all three forms of scratch. Scratching in the turtle is rhythmic; all three forms of scratching display rhythmic alternation between hip flexor muscle activity and hip extensor muscle activity. Timing of the monoarticular knee extensor muscle is distinct for each scratch form. The monoarticular knee extensor muscle is active (1) during the latter part of hip flexor muscle activity in a rostral scratch, (2) during hip extensor muscle activity in a pocket scratch, and (3) after the burst of hip extensor muscle activity in a caudal scratch. The motor pattern of muscle/motor-pool activation is distinct for each scratch form.

Body Degree-of-Freedom Coordinate Description of Movement

The time course of hip angle (angle of hip flexion/extension) and knee angle (angle of knee flexion/extension) has been studied during each of the three forms of turtle scratch reflex (see discussion in Earhart and Stein, 2000). Rhythmic alternation between hip flexion and hip extension occurs for all three scratch forms. Timing of

knee extension in the cycle of hip flexion and extension is distinct for each scratch form. The knee extends during the latter part of hip flexion in a rostral scratch; the knee extends during hip extension in a pocket scratch; the knee extends after hip extension is completed in a caudal scratch. A specific *movement pattern* in the joint-angle coordinate system is distinct for each form; that is, there is regulated timing of knee extension in the cycle of hip movement.

The movement pattern for each form in time and joint-angle space is similar to the motor pattern for each form measured in time and muscle/motor-pool space. In both spaces for each form of the scratch, there is a regulated timing of the knee with respect to the cycle of the hip. Similar changes of timing of a distal joint in the cycle of a proximal joint have been observed for other behaviors, e.g., forward versus backward stepping in humans.

Spinal Cord Networks for Scratch Reflex

Spinal cord networks responsible for producing the scratch reflex in the cat (Arshavsky et al., 1986) and in the turtle (Stein, 1989; Currie and Stein, 1992; Berkowitz and Stein, 1994; Berkowitz, 2001; Stein and Daniels-McQueen, 2002) are partially understood. This understanding relies upon the demonstration that the spinal cord can produce a motor pattern in the absence of actual movements. Movements are prevented by muscle acetylcholine receptor blockade with a specific antagonist, e.g., curare. The motor pattern is measured as the electroneurographic (ENG) activities of specific motor pools in response to a stimulation of a site in a scratch receptive field. The ENG motor patterns recorded in the absence of movements are termed *fictive* motor patterns.

The ENG motor pattern for each scratch form in the spinal immobilized turtle is generated in response to stimulation of a site in the receptive field for that scratch form. Each motor pool monitored in the immobilized turtle using ENG recording techniques innervates a muscle that was previously monitored using EMG recording techniques during actual movements. The ENG motor pattern is an excellent replica of the EMG motor pattern recorded during actual movements. These results establish that spinal cord neuronal networks select the appropriate scratch motor pattern in response to stimulation of a specific site on the body surface; thus, motor strategy selection is a property of spinal cord neuronal networks. These rhythmic motor patterns are produced in an open-loop condition without benefit of timing cues from movement-related sensory feedback; thus, rhythmic motor pattern generation is an intrinsic property of spinal cord neuronal networks.

Motor patterns produced in the absence of movement-related feedback are termed *central motor patterns*. The neuronal network responsible for generating a central motor pattern for a behavior is termed a *central pattern generator* (CPG) for that behavior (Stein et al., 1997). A goal of current research is to reveal the properties of the CPG for each scratch form. It is possible that the CPG for one scratch form shares no neural circuitry with the CPG for another scratch form; such a lack of overlap in circuitry is not likely to occur, however. Single-neuron recordings in the turtle support the hypothesis that the CPG for the rostral scratch may share neural elements with the CPG for the pocket scratch (Berkowitz and Stein, 1994; Berkowitz, 2001).

The scratch motor pattern is not independent of movement-related sensory input, however. Motor patterns are subject to important modulations due to sensory input (see MOTOR PATTERN GENERATION). For example, EMG recordings during actual scratching in the cat demonstrate amplitude and phase modulations of the motor pattern due to sensory feedback during paw contact with the stimulated site (Kuhta and Smith, 1990); similar modulations of the EMG motor pattern are also seen in the spinal turtle when the foot catches against the rod that is used to deliver the mechanical stimulus (Stein, 1983).

Localization and Distribution of Spinal Cord Neuronal Networks

The spinal cord is a segmental structure. Each segment receives sensory input from a specific region of the body surface termed the *dermatome* of that spinal segment. Each segment contains the cell bodies of motor neurons that innervate a specific set of muscles. The hindlimb enlargement is the set of spinal segments that contain the cell bodies of motor neurons that innervate hindlimb muscles.

Anterior segments of the hindlimb enlargement play an important role in scratch rhythm generation in the cat (Arshavsky et al., 1986) and in the turtle (Stein, 1989). In all limbed vertebrates, the anterior portion of the hindlimb enlargement contains hip flexor motor neurons and knee extensor motor neurons. A scratch motor rhythm is produced by the most anterior segment of the turtle hindlimb enlargement in response to stimulation of a site in that segment's dermatome. A rhythmic pocket scratch motor pattern is produced by the three most anterior segments of the turtle hindlimb enlargement in response to stimulation of a site in the dermatome of the most anterior segment. The spinal segment just anterior to the hindlimb enlargement also contributes to rhythmogenesis. Thus neuronal networks for scratching are contained in and distributed among a set of spinal segments.

Multisecond Excitability Changes in Scratch Neuronal Networks

The turtle scratch motor response can continue for several seconds after the termination of sensory stimulation (Currie and Stein, 1992). For an additional several seconds after the cessation of motor neuron activity, there is an increased excitability of spinal cord neuronal networks. This afterexcitability is form specific and is a physiological measure of spinal cord selection processes. NMDA receptors contribute to this afterexcitability (Currie and Stein, 1992; see also NMDA RECEPTORS: SYNAPTIC, CELLULAR, AND NETWORK MODELS). The long time constant of NMDA receptor activation is well suited for multisecond excitability changes.

Spinal cord neurons, termed long-afterdischarge interneurons, are activated by stimulation in a region of a scratch receptive field and are active for many seconds after the cessation of stimulation. Long-afterdischarge interneurons may play a role in motor pattern selection in scratch neuronal networks. NMDA receptors contribute to the excitability of long-afterdischarge interneurons (Currie and Stein, 1992).

Broad Tuning of Interneurons in Neuronal Networks for Scratching

There is broad tuning in the responses of individual turtle interneurons activated by stimulation of sites in the receptive fields for the rostral scratch and for the pocket scratch (Berkowitz and Stein, 1994; Berkowitz, 2001). Some of these interneurons are activated throughout the entire region of the scratch receptive fields for each of several forms. Many of these interneurons may be members of both the rostral scratch CPG and the pocket scratch CPG. For each interneuron, there is usually a site whose stimulation results in the highest frequency of action potentials; stimulation of other sites usually results in interneuron firing frequencies that decrease as the distance from the site that evokes the greatest response increases. These data are consistent with the hypothesis that motor pattern selection results from the summed activities of a population of broadly tuned interneurons that are shared by several CPGs.

Scratch CPG Rhythmogenesis

Reciprocal inhibition between hip flexor and hip extensor interneurons plays a major role in spinal cord scratch rhythmogenesis; additional mechanisms for rhythmogenesis must exist, however,

since hip flexor motor rhythms are generated in the absence of hip extensor motor activity (Stein et al., 1995; Stein, McCullough, and Currie, 1998). During quiescence of hip extensor motor neuron activity, there is also quiescence of hip extensor interneuron activity (Stein and Daniels-McQueen, 2002). This supports the concept that interneurons that are active during hip flexor motor output are rhythmogenic. Left-right interactions also play a key role in rhythmogenesis (Stein et al., 1995; Stein, McCullough, and Currie, 1998).

Discussion

Scratching can be used to uncover important characteristics of neuronal networks that perform sensory-to-motor transformations. Future experiments are required for a more complete understanding of the properties of these neuronal networks.

Road Maps: Motor Pattern Generators; Neuroethology and Evolution

Related Reading: Gait Transitions; Locomotion, Vertebrate; Motor Primitives

References

- Arshavsky, Y. I., Gelfand, I. M., and Orlovsky, G. N., 1986, *Cerebellum and Rhythmical Movements*, Berlin: Springer-Verlag. ♦
- Berkinblit, M. B., Feldman, A. G., and Fokson, O. I., 1989, Wiping reflex in the frog: Movement patterns, receptive fields, and blends, in *Visuomotor Coordination* (J.-P. Ewert and M. A. Arbib, Eds.), New York: Plenum Press, pp. 615–629.
- Berkowitz, A., 2001, Broadly tuned spinal neurons for each form of fictive scratching in spinal turtles, *J. Neurophysiol.*, 86:1017–1025.
- Berkowitz, A. and Stein, P. S. G., 1994, Activity of descending proprio-spinal axons in the turtle hindlimb enlargement during two forms of fictive scratching: phase analyses, *J. Neurosci.*, 14:5105–5119.
- Currie, S. N., and Stein, P. S. G., 1992, Glutamate antagonists applied to midbody spinal cord segments reduce the excitability of the fictive rostral scratch reflex in the turtle, *Brain Res.*, 581:91–100.
- Earhart, G. M., and Stein, P. S. G., 2000, Step, swim, and scratch motor patterns in the turtle, *J. Neurophysiol.*, 84:2181–2190.
- Kuhta, P. C. and Smith, J. L., 1990, Scratch responses in normal cats: hindlimb kinetics and muscle synergies, *J. Neurophys.*, 64:1653–1667.
- Orlovsky, G. N., Deliagina, T. G., and Grillner, S., 1999, *Neuronal Control of Locomotion from Mollusc to Man*, New York: Oxford University Press. ♦
- Sherrington, C. S., 1906, *The Integrative Action of the Nervous System*, New Haven, CT: Yale University Press.
- Stein, P. S. G., 1983, The vertebrate scratch reflex, *Symp. Soc. Exp. Biol.*, 37:383–403. ♦
- Stein, P. S. G., 1989, Spinal cord circuits for motor pattern selection in the turtle, *Ann. NY Acad. Sci.*, 563:1–10. ♦
- Stein, P. S. G., and Daniels-McQueen, S., 2002, Modular organization of turtle spinal interneurons during normal and deletion fictive rostral scratching, *J. Neurosci.*, 22:6800–6809.
- Stein, P. S. G., Grillner, S., Selverston, A. I., and Stuart, D. G., Eds., 1997, *Neurons, Networks, and Motor Behavior*, Cambridge, MA: MIT Press. ♦
- Stein, P. S. G., McCullough, M. L., and Currie, S. N., 1998, Reconstruction of flexor/extensor alternation during fictive rostral scratching by two-site stimulation in the spinal turtle with a transverse spinal hemisection, *J. Neurosci.*, 18:467–479.
- Stein, P. S. G., Mortin, L. I., and Robertson, G. A., 1986, The forms of a task and their blends, in *Neurobiology of Vertebrate Locomotion* (S. Grillner, P. S. G. Stein, D. G. Stuart, H. Forssberg, and R. M. Herman, Eds.), London: Macmillan, pp. 201–216. ♦
- Stein, P. S. G., and Smith, J. L., 1997, Neural and biomechanical control strategies for different forms of vertebrate hindlimb motor tasks, in *Neurons, Networks, and Motor Behavior* (P. S. G. Stein, S. Grillner, A. I. Selverston, and D. G. Stuart, Eds.), Cambridge, MA: MIT Press, pp. 61–73. ♦
- Stein, P. S. G., Victor, J. C., Field, E. C., and Currie, S. N., 1995, Bilateral control of hindlimb scratching in the spinal turtle: Contralateral spinal circuitry contributes to the normal ipsilateral motor pattern of fictive rostral scratching, *J. Neurosci.*, 15:4343–4355.

Self-Organization and the Brain

Christoph von der Malsburg

Introduction

There is a fundamental difference between brain and computer. The computer is based on the algorithmic division of labor. It relies on the separate existence of minds that program and interpret. The brain is a physical, dynamical system to which the algorithmic scheme cannot be applied except in a metaphorical sense. The brain is not digital, not deterministic in any operational sense, and in its inner workings it never gets help from a separate interpreting and planning mind. Instead, the brain is organized on the basis of physical interactions between its elements and of the nested processes of evolution, ontogenesis, and state organization. Unfortunately, our thinking about brain and mind is still very much in the grips of the algorithmic scheme, and we are always in danger of seeing in evolution a programmer and in the genome a program, of taking perception as a mere preparation of material for a "higher level" always just outside our field of view, or of subscribing to a model of learning that relies on the presence of an experimenter to select and prepare the learning patterns and their encoding.

Self-organization is the process by which ordered structures arise in dynamical systems (see PATTERN FORMATION, BIOLOGICAL; COOPERATIVE PHENOMENA; Haken, 1978; Ball, 1999). The theory of evolution has always been a theory of self-organization (EVOLUTION OF THE ANCESTRAL VERTEBRATE BRAIN; EVOLUTION OF GENETIC NETWORKS; EVOLUTION OF ARTIFICIAL NEURAL NETWORKS), and the perspective of self-organization is gaining growing acceptance as well at the level of morphogenesis (not treated here) and the organization of neural connectivity (see DEVELOPMENT OF RETINOTECTAL MAPS; OCULAR DOMINANCE AND ORIENTATION COLUMNS) and brain state (see DYNAMIC LINK ARCHITECTURE). But our ideas about self-organization will have to be developed much further before they can get algorithmic thinking totally out of the way.

Self-Organization in General

Self-organization is the phenomenon by which simple mechanisms of growth and interaction lead to the establishment of complex structures of global order. The phenomenon dominates our world, from the crystalline structure of minute snowflakes to the flow patterns in our coffee cup, atmosphere, and oceans to the formation of stars, planetary systems, and galaxies. Its most awesome expression, however, is found in the structures of the living world. Elementary arrangements cooperate and compete to form more complex patterns, these in turn conspiring to create structures of higher and higher order. What distinguishes winning patterns is the degree of harmony of their elementary arrangements within their smaller and larger contexts. The most interesting aspect of the phenomenon is the tremendous contrast between the simplicity of elementary interactions and the complexity and beauty of the structural order they create. Although a silent revolution in favor of self-organization as a fundamental principle has swept the intellectual world within the past two or three decades, we are still far from realizing the full impact of the phenomenon on our comprehension of things.

Self-organized structures are characterized by inherent order—symmetries, self-similarity, repetition, and many other kinds of regularity: the mechanisms of self-organization are not free to create arbitrary arrangements and can only select from a relatively narrow universe of ordered patterns. For example, although no two snowflakes are ever alike, they realize a vanishing subset of all arbitrary arrangements of water molecules. The amount of information

needed to specify a self-organized pattern is therefore very modest, tremendously smaller than the (logarithm of) number of all random arrangements of elements. In consequence, describing phenomena in terms of self-organized structures is a very powerful tool, and focusing attention on those structures is of great scientific importance.

Although science still has to struggle to find and formulate much of the universe of structured patterns that can be realized by self-organization, this universe is preexisting in the mathematical sense. (To give a specific example, the set of all possible periodic crystal lattices can be completely specified mathematically.) It is impressive to see to what extent the structure of this universe is determined by laws and relationships operating at higher levels and independent of the detail of elementary mechanisms and interactions. Structure is determined to a large extent from the top down, supporting the hope that even systems as complex as the brain can ultimately be understood in relatively simple terms.

In an elementary process of self-organization, a system makes the transition from a simple, relatively unstructured initial state that is in the process of becoming unstable to a more structured final state in which the dynamic forces reach equilibrium again. The transition starts with random *fluctuations*, which are small deviations from the initial state. The process is governed by three types of interactions: fluctuations self-amplify, fluctuations cooperate, and fluctuations compete. In the interplay between these interactions, certain constellations of fluctuations collude to form patterns on a larger scale.

A simple and well-understood example of self-organization, originally proposed by Bénard, is a flat pan filled with liquid and heated from below. (Other examples of self-organizing systems are discussed in COOPERATIVE PHENOMENA.) In an initial state, the liquid is at rest. When the temperature gradient is strong enough, the initial state becomes unstable, and fluctuations in the form of small regions of upwelling or downwelling liquid start to grow in amplitude. These fluctuations are self-amplifying, as upward motion draws more hot liquid from the bottom, creating a column of fluid that is hotter and lighter than the surrounding fluid, and similarly for downward motion. Fluctuations cooperate, with vertical movement in one small column dragging along movement in neighboring columns. And they compete, as upward movement in one place must be made up by downward movement in other places. These interactions establish convection cells, regions of coherent motion, which typically take the shape of hexagons or rolls (the latter being akin to the parallel bands of clouds sometimes seen in the sky), often of impressively regular arrangement. In a given system, fluctuations can be defined on several levels. In the Bénard system, for instance, "fluctuations" may be identified with droplets of moving liquid, as I just did, but also with whole coordinated patterns of mutually consistent up-and-down movement, which may have the form of sinusoidal waves fluctuating in amplitude.

The dynamic properties of interacting fluctuations, and consequently the selection of particular final structures, usually is subject to so-called control parameters (among the control parameters of the Bénard system are the temperature gradient, the viscosity, and the depth of the liquid). Moreover, a pattern resulting from self-organization may set the stage for more self-organization, sometimes making a final result dependent on the detailed history of a long chain of processes. This aspect is especially important in biology, whose structures are the result of phylogenetic and ontogenetic history.

Mathematical Methods

There are three types of theoretical tools to describe and understand self-organizing systems: computer simulation, statistical mechanics, and dynamical differential equations and the attendant fields of catastrophe theory and stability analysis. Statistical mechanics is a method to derive relations between global properties of physical systems from the statistics of the constituent atoms or molecules. Typical global parameters are temperature, energy, volume, and pressure. From a very simple basic idea ("all detailed configurations of the system compatible with global parameters are attained with equal probability") all interesting quantities can be derived mathematically (see *STATISTICAL MECHANICS OF NEURAL NETWORKS* and Hertz, Krogh, and Palmer, 1991). Important applications of statistical physics are phase transition systems. They fall into "universality classes." It is remarkable to what extent the qualitative behavior of systems in a given universality class is independent of detailed material properties. What above was called "harmony" materializes in statistical physics as a quantity called free energy (great harmony is low free energy).

Many self-organizing systems can be completely characterized by continuous variables and their deterministic interactions (in Bénard's system these are the local velocity of the flowing liquid and the forces acting on it). Such systems are conveniently described as systems of coupled differential equations (see Haken, 1978; see also *COOPERATIVE PHENOMENA*). These are always of nonlinear type and can be solved analytically (that is, with pencil and paper) only in the rarest cases. Computer simulation is a convenient way to explore the behavior of such systems. However, since the behavior of a system can depend in unexpected ways on the precise settings of parameters, it is profitable to analyze this behavior using stability analysis (see *DYNAMICS AND BIFURCATION IN NEURAL NETS*; *COOPERATIVE PHENOMENA*; Haken, 1978). As a result, a simpler description of the system is derived in the form of a small number of superimposable patterns ("linear modes") and differential equations describing the behavior of their amplitudes. Stability analysis is thus a powerful conceptual tool to make the transition from a description at the level of microscopic building elements to macroscopic modes of behavior.

Self-Organization in the Brain

The brain is often cited as the most complex entity in the universe. And indeed it is an awesome array of structure that spans many orders of magnitude. Although this can also be said of a simple rock, what distinguishes the brain is the purposeful arrangement of its detail. Pathological alteration at any level—molecular, synaptic, cellular, or macroscopic—easily leads to serious functional deficit. On the other hand, much of the variation from individual to individual, or even from second to second in a given individual, seems to be fully compatible with physiological function. This raises the great puzzle of how the myriad important functional relationships are created and maintained in the brain. According to an old mode of thinking, a Creator, or Mother Nature, or a genetic program is in control of every molecular reaction with exquisite algorithmic foresight. Although this thought pattern of "hetero-organization" under control of a preexisting detailed plan still governs many a thought subconsciously, it is quickly losing dominance, giving way to models of self-organization at all levels—evolution, ontogenesis, learning, and functional organization. As a result, instead of blindly accepting any arbitrary brain structure as of equal *a priori* likelihood, the explicit examination of mechanisms of organization in evolution and ontogenesis yields powerful constraints on what to expect, reducing the search space of experimental science by many orders of magnitude.

A potent argument against the brain as irreducible heap of structure is the gross mismatch between the amount of information con-

tained in its detail as measured naively and the amount of available genetic information. The direct specification of an arbitrary wiring pattern of the 10^{14} connections between the 10^{10} neurons in the human cerebral cortex would, for instance, take more than 10^{15} bits of information. By comparison, there are little more than 10^9 bits in the human genome. This mismatch of more than six orders of magnitude vanishes if the genes are required only to specify the rules of the game and some control parameters for ontogenetic self-organization. As a consequence, the resulting connectivity patterns, far from random, have to be highly regular. Some of this regularity is known to us and has, for instance, the form of topographic organization. The bulk of the connective architecture of the brain, however, is unknown and very difficult to determine experimentally. The most parsimonious way to describe the brain (including individual variations) would be in terms of generative rules and appropriate theoretical tools to retrace ontogeny. Intimate knowledge of the laws of organization is also a prerequisite for analysis and cure of many ailments of the brain, much more than mere knowledge of static structure (even if it could be had).

The evolution of the brain cannot possibly be imagined as a progression of mutation and selection at the level of individual cells or synapses ("evolution by rote"). This search space would just be too large, even on the time scale of hundreds of millions of years, and the resulting information could not be transmitted from generation to generation through the limited genome. In reality, however, as the genes determine the laws of self-organization and its control parameters, the search space is much smaller. Evolution restricts itself to finding fruitful architectures and tuning them to the particular biological needs of the species by alterations at the control parameter level.

The neurosciences are making extensive use of inductive generalization from a few strategic observations. This is possible only on the basis of the regularities that are the hallmark of self-organization. Progress could be accelerated and many a false start and expensive investigation avoided if we had a clearer picture of the control structures and regularities of the brain as a self-organizing entity. So far the surface has hardly been scratched.

An immense amount of experimental and theoretical work will have to be done. The perspective of self-organization is not a magic bullet that reduces the brain problem to armchair theorizing and a few simple equations. Our nervous system is made up of a bewildering variety of cell types of different anatomy, molecular makeup, and behavioral repertoire, and our brain has an immensely convoluted gross anatomy of interconnected nuclei and areas. Even if the details of neural connectivity could be derived from general laws, these general laws are conditioned from below and above by the underlying behavioral repertoire of the constituent elements and the gross boundary conditions. The amount of information in our genome may be much smaller than any naively calculated information content of the brain, but it is still a very large quantity.

In order for the brain to guide the animal safely through life, it must be in tune with the environment. How can it be so? One important answer evidently is that the brain learns through examples. Unfortunately, this process is far from being fully understood. Each situation with which a brain has to cope is unique. Only appropriate analysis and coding makes information from one scene applicable to others. These mechanisms of analysis and proper representation are unlikely to be learned themselves from example. They exist prior to experience (as postulated by Immanuel Kant) and rely to a large extent on self-organization. Although this sounds like a daring proposition, requiring the universe of self-organized brain structures to be generally in tune with the structure of the environment by "preestablished harmony," no other solution to the epistemology riddle is in sight. The self-organized ontogenesis of brain structures constitutes a natural language, and all evolution had to do was use this language to write the particular text that defines us.

Activity Self-Organization

In the formulation of McCulloch and Pitts (1943), neural networks have binary signals and synchronized switching and are thus equivalent to digital machines. Just as in digital machines, these networks and their activity have no inherent tendency to fall into organized patterns. McCulloch and Pitts do not discuss any mechanism for the generation of networks or the underlying logical functions to be realized. These are based entirely on the algorithmic schema and must be structured on the basis of a separate programmer's insight. With appropriate changes, however, neural networks can be made to self-organize. For this, some combination of giving up synchrony of switching, introducing graded instead of binary signals, and specializing connectivity patterns to near symmetry (between the forward and backward connection between two neurons) seems to be important. Essential for self-organization is a tendency to iteratively approach stationary states (or nearly stationary states).

The self-organization of activity patterns is most simply discussed and modeled in homogeneous two-dimensional sheets of neurons with short-range excitation and longer-range inhibition, assumptions that are consistent with what is known about cortical anatomy. A homogeneous activity distribution in a sheet of neurons becomes unstable when the slope of the input-output function of the tissue grows beyond a critical value (see *DYNAMICS AND BIFURCATION IN NEURAL NETS*). Small fluctuations in signal strength then start to grow and self-amplify. Due to short-range excitation, they cooperate locally; through longer-range inhibition they compete. As a consequence, organized patterns grow.

Under idealized conditions these patterns may have the form of regular arrays of blobs or of parallel stripes. Under more realistic conditions one observes more or less regular waves or arrays of blobs, closely corresponding in character to ripples in sand under irregularly moving water, and many other systems.

Such activity patterns are found in the visual cortex of cat and monkey with optical or other recording techniques. These patterns may play a role in the generation and expression of the regular columnar arrangement of cellular properties such as *OCULAR DOMINANCE AND ORIENTATION COLUMNS* (q.v.). They have also been proposed as an explanation for visual hallucinations during migraines or states of intoxication (see *PATTERN FORMATION, BIOLOGICAL*).

If, as is the reigning opinion, neural connections are the repository of knowledge in the brain, activity self-organization is the mechanism by which elementary knowledge (in the form of connections) is combined into useful thought patterns. An admittedly very simple first model of this process is associative memory (Hertz et al., 1991; see also *COMPUTING WITH ATTRACTORS*). Activity states are stored by *HEBBIAN SYNAPTIC PLASTICITY* (q.v.), that is, by strengthening connections between neurons that are simultaneously active in a state. Later, the state can be dynamically recreated by self-organization, starting from an incomplete version of it. The model thus treats self-organization at two levels, the formation of connections and the formation of activity states. The model has been fully analyzed by the methods of statistical physics, which helped to identify important parameters and gave useful information on memory capacity. It may be hoped that the obvious weaknesses of the associative memory model—it only recreates states established previously and cannot generalize to new scenes—can be overcome to create a more realistic model of brain state organization.

The main reason for associative memory's weakness is the monolithic character of its activity states. Lately, much experimental and theoretical study has been devoted to spatiotemporal patterns as a means to segregate the mass of neurons simultaneously active in a given brain state into separate chunks (corresponding to

separate objects in a scene) by synchronizing the signals of neurons within chunks and desynchronizing them between chunks (see *SYNCHRONIZATION, BINDING AND EXPECTANCY*).

Network Self-Organization in Ontogenesis

Network self-organization conforms to a simple scheme (von der Malsburg, 1995). A network and its inputs create activity patterns. These are characterized by statistical correlations between pairs of neural signals. In response to these correlations, individual synapses are changed by *HEBBIAN SYNAPTIC PLASTICITY* (q.v.). In terms of the general scheme of self-organization discussed above, elementary fluctuations are deviations of synaptic strengths from their initial state. Fluctuations self-amplify as result of the positive feedback between excitatory synapses and the correlations they create. Cooperation rules between synapses that converge on the same neuron if their activity is correlated, as a consequence of which they help each other establish favorable postsynaptic activity. Competition rules between synapses if the activity states they favor are not compatible with each other. The altered network creates modified patterns of signal correlations, which in turn modify synaptic strengths. This feedback loop may create a runaway situation that drives the network state away from the initial, unstructured state. Because of their elementary interactions, natural alliances exist between synapses, and the network that finally wins is a constellation of connections that is optimal in terms of cooperation and competition.

The ontogenesis of fiber projection systems has been studied in great detail in the example of the retinotectal system (see *DEVELOPMENT OF RETINOTECTAL MAPS*; von der Malsburg, 1995). There seems to be a consensus that three major mechanisms are guiding fibers: gross routing, fiber positioning, and fiber sorting. Apparently, the first two mechanisms are based on gradients of morphogens that guide fibers first to the target structure and then to their target position within that structure. These mechanisms are simple extensions of general themes of morphogenesis (see the chapter on bodies in Ball, 1999). The retinotectal system shows considerable flexibility in dealing with unexpected variations imposed experimentally or by anomalous development. The projection can, for instance, adjust to changed relative size of retina and tectum. The adaptability of the retinotectal system is explained by the third mechanism, fiber sorting. This mechanism has been shown to be dependent on neural activity. It has been modeled with great success and is now a prime paradigm for neural network self-organization. Other instances of network self-organization deal with the ontogenesis of *OCULAR DOMINANCE AND ORIENTATION COLUMNS* (q.v.; see also von der Malsburg, 1995) and of structures in the barrel field of the rodent cortex.

According to recent evidence, synapses can change their weight on a very rapid time scale (for a review see Hempel et al., 2000). It might be surmised that these changes are controlled by network-created short-term signal fluctuations in a rapid version of *HEBBIAN SYNAPTIC PLASTICITY* (q.v.). If that is the case, a process of network self-organization may be active, adapting the connectivity pattern of the brain to the current context in which it is operating on time scales of fractions of seconds to minutes (see *DYNAMIC LINK ARCHITECTURE*).

Conclusion

The theme of self-organization currently receives surprisingly little attention in the brain theory community. And yet there is a tremendous, largely uncharted territory to be explored. The easily accessible geometrical patterns of activity and connectivity have been modeled in hundreds of publications. However, the material with which neural self-organization plays has the form of a *net-*

work, in which neighborhood is defined in terms of neural connectivity and not (or at least not directly) in terms of two- or three-dimensional geometry. We therefore must prepare ourselves for a totally new universe of network and activity patterns in which order is defined in terms of abstract relationships and quantities, and not in terms of pretty pictures. Second, the theory of self-organization has hitherto focused on the establishment of static structures. The nervous system, however, is about the generation of purposeful, nested processes evolving in time. Third, getting the simulation of a self-organizing system to work, that is, to create the intended type of patterns, is still an art. There are usually quite a few control parameters in a system, and they must all be put in the right ballpark. Blind search is usually out of the question for quantitative reasons. In view of the variability of the physiological state of the nervous system, evolution cannot simply have developed fixed values for these parameters. Nature must have developed general mechanisms to actively and autonomously regulate its systems, such as to produce interesting self-organized processes and states.

Finally, it will be important to understand the process of brain organization as a cascade of steps, each one taking place within the boundary conditions established by the previous one, and the theory of such cascades is as yet nonexistent. The tremendous creativity inherent in our brain's state organization cannot be well understood without knowing the arena set up by ontogenesis and learning, and ontogenesis cannot well be understood without understanding the genetic control structure set up by evolution. The perspective of self-organization is more of a program for future

work than an accomplished success. There is a new continent to be discovered. It is the continent where our mind lives. It is time to set sail.

Road Map: Dynamic Systems

Related Reading: Cooperative Phenomena; Development of Retinotectal Maps; Ocular Dominance and Orientation Columns; Self-Organizing Feature Maps

References

- Ball, P., 1999, *The Self-Made Tapestry: Pattern Formation in Nature*, New York: Oxford University Press. ♦
- Haken, H., 1978, *Synergetics: An Introduction. Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry and Biology*, 2nd enlarged ed., New York: Springer-Verlag. ♦
- Hempel, C. M., Hartman, K. H., Wang, X.-J., Turrigiano, G. G., and Nelson, S. B., 2000, Multiple forms of short-term plasticity at excitatory synapses in rat medial prefrontal cortex, *J. Neurophysiol.*, 83:3031–3041.
- Hertz, J., Krogh, A., and Palmer, R. G., 1991, *Introduction to the Theory of Neural Computation*, Redwood City, CA: Addison-Wesley.
- McCulloch, W. S., and Pitts, W. H., 1943, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.*, 5:115–133.
- von der Malsburg, C., 1995, Network self-organization in the ontogenesis of the mammalian visual system, in *An Introduction to Neural and Electronic Networks*, 2nd ed. (S. F. Zornetzer, J. Davis, and C. Lau, Eds.), New York: Academic Press, pp. 447–463. ♦

Self-Organizing Feature Maps

Helge Ritter

Introduction

A first and very important step in many pattern recognition and information processing tasks is the identification or construction of a reasonably small set of important features in which the essential information for the task is concentrated.

The *self-organizing feature map* (SOFM) is a nonlinear method by which such features can be obtained with an unsupervised learning process. It is based on a layer of adaptive units ("neurons," Figure 1) that gradually develops into an array of feature detectors

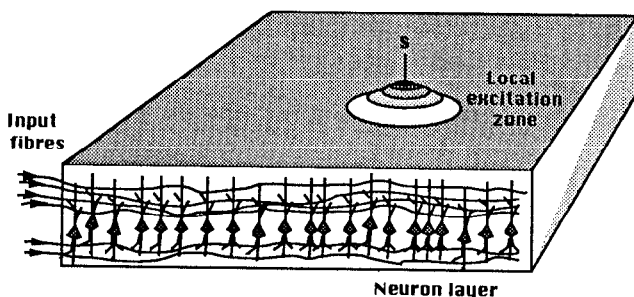


Figure 1. Schematic representation of a feature map. Nerve fibers providing the input signal excite the neurons via synaptic connections. Lateral interactions restrict the neural responses and the synaptic adaptation to a local excitation zone. Each possible position s of the excitation zone can be viewed as a compressed image in a two-dimensional space of the original stimulus features.

that is spatially organized in such a way that the location of the excited units becomes indicative of statistically important features of the input signals.

The linking of input signals to response locations in the map can be viewed as a nonlinear projection from a signal or input space to the (usually) two-dimensional (2D) map layer. The resulting "compressed image" of the (usually higher-dimensional) input space has the property of a topographic map that reflects important metric and statistical properties of the input signal distribution: distance relationships in the input space (expressing, e.g., pattern similarities) are approximately preserved as distance relationships between corresponding excitation sites in the map, and clusters of similar input patterns tend to become mapped to coherent areas whose size varies in proportion to the frequency of the occurrence of their patterns.

This resembles in many ways the structure of topographic feature maps found in many brain areas, for which the SOFM offers a neural model that bridges the gap between microscopic adaptation rules postulated at the single neuron or synapse level and the formation of experimentally better accessible, macroscopic patterns of feature selectivity in neural layers. From a statistical point of view, the SOFM provides a nonlinear generalization of principal component analysis and has proved valuable in many application contexts, ranging from pattern recognition and optimization to robotics (Ritter, Martinetz, and Schulten, 1992; Kohonen, 1995).

The Basic Feature Map Algorithm

The SOFM algorithm provides an unsupervised learning rule by which the adaptive units (neurons) can tune their response prop-

erties in such a way that the described, topographic map structure arises. The process is iterative and is driven by a sequence of activity patterns on some shared set of (afferent) input lines (Figure 1), at each step activating and adapting some local group of neurons.

Successful self-organization of the map requires the following: (1) the neurons must be exposed to a sufficient number of different inputs; (2) for each input, only the synaptic input connections to the excited group are affected; (3) similar updating is imposed on many adjacent neurons; and (4) the resulting adjustment is such that it enhances the same responses to a subsequent, sufficiently similar input (Kohonen, 1982).

A standard formulation of this process models the input patterns as vectors \mathbf{x} from some pattern space V (whose dimensionality n is given by the number of input lines) and the neural layer as a set of formal neurons occupying discrete sites r in a (e.g., planar) grid A . The response of a neuron at site r is determined by the dot product $\mathbf{x} \cdot \mathbf{w}_r$, where \mathbf{w}_r is the vector of the neuron's synaptic connection strengths with the n afferents carrying the input pattern \mathbf{x} .

For each input \mathbf{x} , the adaptive changes in the layer are confined to a local group of neurons centered at the site s for which $\mathbf{x} \cdot \mathbf{w}_s$ is maximal. This is achieved by modulating the adaptive changes for other sites r with an activity profile h_{rs} that has its peak at $r = s$ and that decays to smaller values with increasing distance $\|r - s\|$ from s ("neighborhood cooperation"; it is essential that $\|r - s\|$ be taken in the space of the lattice A , not in the original signal space V):

$$\mathbf{w}_r^{(new)} = (1 - \epsilon \cdot h_{rs})\mathbf{w}_r^{(old)} + \epsilon \cdot h_{rs} \cdot \mathbf{x} \quad (1)$$

Equation 1 can be justified by assuming the traditional Hebbian law for synaptic modification together with an additional nonlinear, "active" forgetting process for the synaptic strengths (Kohonen, 1995).

A rather realistic modeling choice for h_{rs} is, e.g., a Gaussian

$$h_{rs} = \exp\left(-\frac{\|r - s\|^2}{\sigma^2}\right) \quad (2)$$

whose variance $\sigma^2/2$ will control the radius of the group.

Consequently, neurons that are close neighbors in A will tend to specialize on similar patterns. After learning, this specialization is used to define the mapping from the space V of patterns onto the (discretized) space A : each pattern vector $\mathbf{x} \in V$ is mapped to one of the discretized neuron sites of A . The image of \mathbf{x} under this mapping is defined to be the location $s = s(\mathbf{x})$ associated with the neuron for which $\mathbf{x} \cdot \mathbf{w}_s$ is largest ("winner neuron").

A frequently useful modification of this basic algorithm is to replace the winner criterion for the site $s(\mathbf{x})$ by a different one. A frequent choice is minimizing the Euclidean difference $\|\mathbf{x} - \mathbf{w}_s\|$ (which is equivalent to maximizing $\mathbf{w}_s \cdot \mathbf{x}$ when the vectors are normalized), but many other similarity measures can also be used to obtain SOFMs.

Visualization of Feature Maps

There are two main ways to visualize a feature map. The first approach labels each neuron in A by the test pattern that excites this neuron maximally (best stimulus). It yields a partitioning of a map into coherent regions of similarly specialized neurons and resembles the experimental procedure by which sites in a brain area are labeled by those stimulus features that are most effective in exciting neurons at this site. In the example in Figure 2, each training and test pattern consisted of 13 simple binary-valued features describing one of 16 animals (Ritter and Kohonen, 1989) whose similarity relationships are reflected in the resulting partitioning of the SOFM.

duck	duck	horse	horse	zebra	zebra	cow	cow	cow	cow
duck	duck	horse	zebra	zebra	zebra	cow	cow	tiger	tiger
goose	goose	goose	zebra	zebra	zebra	wolf	wolf	tiger	tiger
goose	goose	hawk	hawk	hawk	wolf	wolf	wolf	tiger	tiger
goose	owl	hawk	hawk	hawk	wolf	wolf	wolf	lion	lion
dove	owl	owl	hawk	hawk	dog	dog	dog	lion	lion
dove	dove	owl	owl	owl	dog	dog	dog	dog	lion
dove	dove	eagle	eagle	eagle	dog	dog	dog	dog	cat
hen	hen	eagle	eagle	eagle	fox	fox	fox	cat	cat
hen	hen	eagle	eagle	eagle	fox	fox	fox	cat	cat

Figure 2. Visualization of a 10×10 feature map for a set of pattern vectors describing binary features of 16 animal species. The spatial arrangement of the labeled map regions reflects the similarity relationships between the animals.

In the second approach, the feature map is visualized as a "virtual net" in the original pattern space V . The virtual net is the set of weight vectors \mathbf{w}_r displayed as points in the pattern space V and connected by lines between those pairs $(\mathbf{w}_r, \mathbf{w}_s)$, whose neuron sites (r, s) are nearest neighbors in the lattice A . While the virtual net is very well suited to display the topological ordering of the map, its use is limited to continuous and at most three-dimensional (3D) spaces. Figures 3A and 3B show the development of the virtual net of a $2D$ 20×20 lattice A from a disordered initial state (Figure 3A) into an ordered final state (Figure 3B) when the stimulus density is concentrated along the surface $z = x \cdot y$ in the cube V given by $-1 < x, y, z < 1$.

Discussion

The main characteristics of the feature map algorithm. Geometrically, the adaptive process (Equation 1) can be viewed as a sequence of local deformations of the virtual net in the space of input patterns so that it *approximates the shape of the stimulus density* $P(\mathbf{x})$ in the space V . A good approximation is possible if the topology of the virtual net (which is the same as the topology of the lattice A) and the topology of the stimulus density are the same (as, e.g., in Figures 3A and 3B). Otherwise, e.g., if the dimensionalities of the stimulus manifold and the virtual net differ, the resulting approximation can only partially fulfill the goal of matching spatially close points of the stimulus manifold to points that are neighbors in the virtual net. An example is depicted in Figure 3C, where the map manifold is 1D (a chain of 400 units), the stimulus manifold is 2D (the surface $z = x \cdot y$), and the embedding space V is 3D (the cube $-1 < x, y, z < 1$). In this case, the dimensionality of the virtual net is lower than the dimensionality of the stimulus manifold (this is the typical situation), and the resulting approximation resembles a "space-filling" fractal curve.

Properties of the features that are represented in a feature map. The geometric interpretation of the previous section suggests that a good approximation of the stimulus density by the virtual net requires that the virtual net be oriented tangentially at each point of the stimulus manifold. Therefore, a d -dimensional

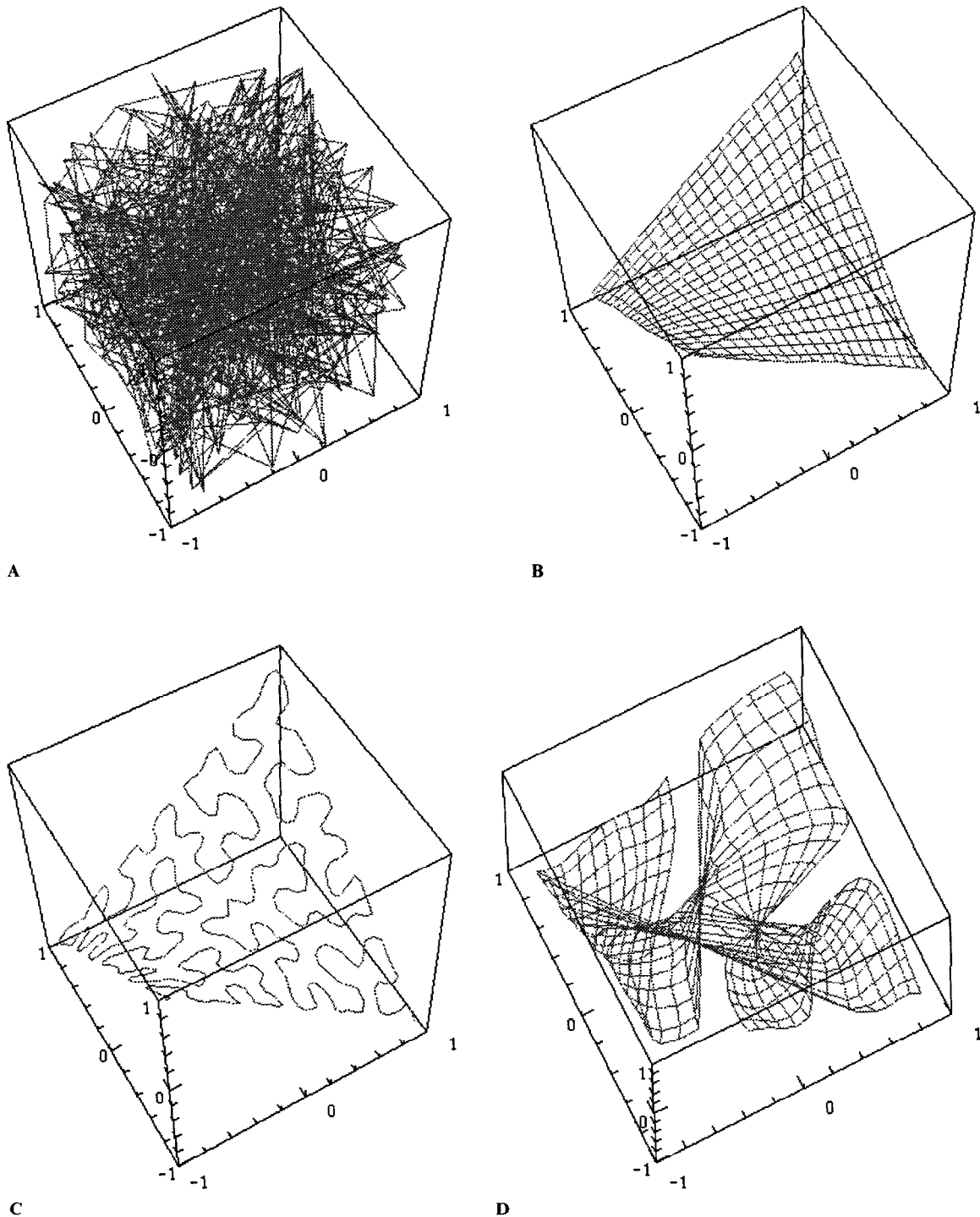


Figure 3. Top: Initial, disordered (A) and final, ordered (B) configuration of the “virtual net” for a two-dimensional feature map, developing under the influence of a two-dimensional stimulus density that is embedded in a three-dimensional signal space. C, *Dimension conflict*: when replacing the

two-dimensional feature map of parts A and B with a one-dimensional chain, a space-filling fractal curve results. D, A *topological defect* is characterized by several patches of globally conflicting local orderings.

feature map will select a (possibly locally varying) subset of d independent features that capture as much of the variation of the stimulus distribution as possible. This is an important property that is also shared by the method of PRINCIPAL COMPONENT ANALYSIS

(PCA) (q.v.). An important difference, however, is that the selection of the “best” features can vary smoothly across the feature map and can be optimized locally. Therefore, the SOFM can be viewed as a nonlinear extension of PCA.

Relationship of the feature map to more traditional approaches. The SOFM shares with a number of other approaches the goal of forming lower-dimensional, distance-preserving mappings of data, which is a general task in data analysis known as *multidimensional scaling* (see DATA CLUSTERING AND LEARNING). Usually, these methods are based on a direct minimization of some *projection index* that quantifies, e.g., the degree of distance distortion, or some measure of “interestingness” (e.g., non-Gaussianity of the resulting image distribution). An example of the first type is the nonlinear *Sammon mapping*, while examples of the latter type are the *projection pursuit methods* that lead to various linear projections (of which PCA can be viewed as a special case when the interestingness measure is just the data variance).

There is also the following relationship with *vector quantization*, which considers the task of finding discrete encodings for continuous signals satisfying certain optimality requirements, such as minimal reconstruction error. The determination of the site $s(\mathbf{x})$ of the winner neuron in a SOFM can be considered as assignment of a code s to a data vector \mathbf{x} . From this code, \mathbf{x} can be reconstructed (decoded) approximately by taking $\mathbf{w}_s(\mathbf{x})$ as its reconstruction, with an average (mean square) reconstruction error

$$E = \int P(\mathbf{x})(\mathbf{x} - \mathbf{w}_{s(\mathbf{x})})^2 d^d \mathbf{x} \quad (3)$$

It can be seen that in the special case $h_{rs} = \delta_{rs}$ (absence of neighborhood cooperation), the adaptation rule (Equation 1) is a stochastic minimization procedure for E , and becomes equivalent to a standard algorithm for finding an optimized set of codebook vectors \mathbf{w}_s , in vector quantization terminology. However, a further source of reconstruction errors may be the *confusion* of two codes s' , s'' with some probability $p(s', s'')$. This situation requires the minimization of a more general error measure, now given by

$$E = \sum_{s', s''} p(s', s'') \int_{s(\mathbf{x})=s'} P(\mathbf{x})(\mathbf{x} - \mathbf{w}_{s''})^2 d^d \mathbf{x} \quad (4)$$

It can be shown that the (approximate) minimization of this error measure, and therefore the construction of an optimal code for this more general situation, is closely related to the feature map formation, provided one takes $h_{ss'} = p(s, s')$; i.e., the neighborhood cooperation in the map is derived from the “confusion matrix” $p(s, s')$ (Luttrell, 1994).

The stationary states of the feature map algorithm. A *stationary state* is a configuration for which the average change of weights per adaptation step vanishes. A stationary state is stable if all sufficiently small perturbations from it decay on the average. However, for a given stable stationary state, this still leaves the possibility that there may exist “very unlikely” sequences of adaptation steps that lead to a different stationary state. If such sequences exist, a stable stationary state is called *metastable*; otherwise it is called *absorbing*. Only the absorbing states are the “true” final states of the time evolution of a feature map. However, if the system is driven into a metastable state, it may become “trapped” for such a long time that the state cannot be distinguished from a true absorbing state within realistic observation times. Usually, the metastable states are those that occur as only partially ordered, while those states that intuitively appear as fully ordered seem to have the property of being absorbing (Erwin, Obermeyer, and Schulzen, 1992a, 1992b).

Speed and reliability of the ordering process in relation to the various parameters of the model. The convergence process of a feature map can be roughly subdivided into a first ordering phase, in which the correct topological order is produced, and a subsequent fine-tuning phase. So far, rigorous convergence proofs for

the full process have only been obtained for the 1D case, and for the 2D case when the correct topological order is specified along the border of A . Since for many geometrically intuitive definitions of order in dimensions greater than 1 one can show that they do not lead to absorbing states for the SOFM process, a very general ordering proof for higher dimensions is unlikely to exist. For a good overview on these and other theoretical aspects of the SOFM, see Cottrell, Fort, and Pagès (1998).

A very important role for the ordering phase is played by the neighborhood kernel h_{rs} . Usually, h_{rs} is chosen as a function of the distance $\|\mathbf{r} - \mathbf{s}\|$; i.e., h_{rs} is translation invariant. The algorithm will work for a wide range of different choices (locally constant, Gaussian, exponential decay) for h_{rs} , but for fast ordering the function h_{rs} should be convex over most of its support. Otherwise, the system may get trapped in partially ordered, metastable states and the resulting map will exhibit “topological defects,” or conflicts between several locally ordered patches (Figure 3D shows a typical example) (Erwin et al., 1992a, 1992b).

Another important parameter is the distance up to which h_{rs} is significantly different from zero. This range sets the radius of the adaptation zone and the length scale over which the response properties of the neurons are kept correlated. The smaller this range, the larger the effective number of degrees of freedom of the network and, correspondingly, the harder the ordering task from a completely disordered state. Conversely, if this range is large, ordering is easy, but finer details are averaged out in the map. In addition, for Gaussian neighborhood kernels, a sufficiently large range also makes the function h_{rs} convex over the entire network. Therefore, formation of an ordered map from a very disordered initial state is favored by a large initial range (a sizable fraction of the linear dimensions of the map) of h_{rs} , which then should decay slowly to a small final value (on the order of a single lattice spacing or less). Although statistical considerations seem to dictate a $1/t$ decay law, the faster exponential decay is suitable in many cases.

It should be noted that the absence of a general ordering proof is mainly of theoretical concern. In practical applications, one usually will start with an already ordered configuration (oriented, e.g., along the major principal axes of the data distribution) and apply the SOFM process for the fine-tuning phase.

Quantitative characterization of the ordering achieved with a feature map. The difficulty of characterizing the achieved ordering is partly due to the difficulty of defining order for embeddings of higher-dimensional lattices in higher-dimensional spaces. A more modest goal is the construction of some cost or energy function that is minimized by the algorithm and that then can serve as a measure of “progress” toward a converged state (without, however, necessarily telling much about the geometric ordering properties of such a state). A good candidate is the function E in Equation 4; however, it has been rigorously shown that the SOFM algorithm does not admit an exact energy function, rendering Equation 4 only an approximation. Remarkably, the approximation can be made exact by changing the SOFM winner criterion for the determination of $s(\mathbf{x})$ (Heskes and Kappen, 1993; Luttrell, 1994):

$$s(\mathbf{x}) = \arg \min_r \sum_{r'} h_{rr'} \|\mathbf{x} - \mathbf{w}_{r'}\| \quad (5)$$

This winner criterion is more costly to compute, and the modified process seems to resemble the SOFM sufficiently closely to warrant the substitution of Equation 5 with the simpler but theoretically less satisfying original winner criterion in practical applications.

In applications, the data topology often is unknown, and the use of a SOFM can only be considered in an attempt to find out how well the (usually) 2D map can approximate the unknown data topology. Various measures of the faithfulness of the resulting maps have been suggested, but their practical usefulness has been limited

by their complexity, which usually far exceeds that of the SOFM algorithm itself.

Relationship between stimulus density and weight vector density. During the second, fine-tuning phase of the map formation process the density of the weight vectors becomes matched to the signal distribution. Regions with high stimulus density in V lead to the specialization of more neurons than regions with lower stimulus density. As a consequence, such regions appear magnified on the map; that is, the map exhibits a locally varying *magnification factor*. In the limit of no neighborhood, the asymptotic density of the weight vectors is proportional to a power $P(\mathbf{x})^\alpha$ of the signal density $P(\mathbf{x})$ with exponent $\alpha = d/(d + 2)$. For a nonvanishing neighborhood, the power law remains valid in the 1D case, but the original SOFM algorithm leads to a different exponent α that now depends on the neighborhood function (Ritter, 1991). For higher dimensions, the relation between signal and weight vector distribution is more complicated, but the monotonic relationship between local magnification factor and stimulus density seems to hold in all cases investigated so far. Here, too, the modified winner criterion (Equation 5) leads to a simplification and yields a density exponent $\alpha = d/(d + 2)$, independent of the neighborhood function.

The effect of a "dimension conflict." In many cases of interest, the stimulus manifold in the space V is of higher dimensionality than the map manifold A , and as a consequence, the feature map will display those features that have the largest variance in the stimulus manifold. These features have also been termed *primary*. However, under suitable conditions, further, *secondary* features may become expressed in the map. The representation of these features is in the form of repeated patches, each representing a full "miniature map" of the secondary feature set. The spatial organization of these patches is correlated with the variation of the primary features over the map: the gradient of the primary features has larger values at the boundaries separating adjacent patches, while it tends to be small within each patch. The conditions for the occurrence of secondary features have been analyzed for simplified situations (Ritter et al., 1992; Obermayer, Blasdel and Schulen, 1992). These results show that the stability of a 2D map with only two primary features requires that the ratio of the variance of the primary features across the range of the neighborhood function and of the variance of the signal manifold perpendicular to the directions of the primary features be below a certain threshold. If the signal distribution is such that no such map configuration exists, additional features will become expressed in the map, with the role of the secondary features assigned to the features with the lower variance.

Modeling the properties of observed brain maps. Many regions in the brain are known to be topographic representations of sensory surfaces. It has been shown that the qualitative structure, including, e.g., the spatially varying magnification factor, and certain experimentally induced reorganization phenomena can be reproduced with the Kohonen feature map algorithm (Ritter et al., 1992). A more stringent test is provided by the more complex maps that are found in the visual cortex. In V1, there is a hierarchical representation of the features: retinal position, orientation, and ocular dominance, with retinal position acting as the primary feature. The secondary features, orientation and ocular dominance, form two correlated spatial structures: (1) a system of alternating bands of binocular preference and (2) a system of regions of orientation-selective neurons arranged in parallel iso-orientation stripes such that orientation angle changes monotonically in the perpendicular direction. The iso-orientation stripes are correlated with the binocular bands such that both band systems tend to intersect perpen-

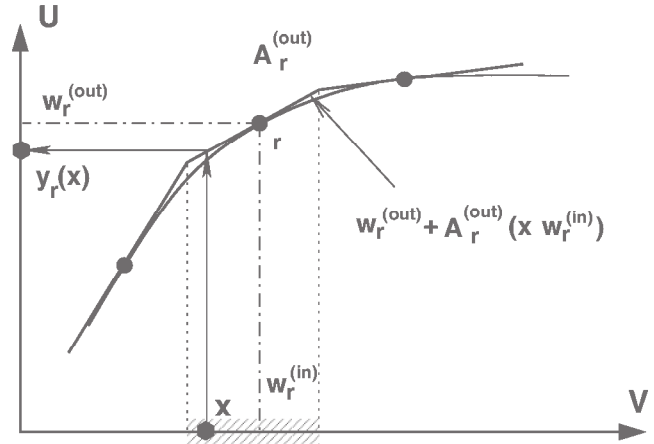


Figure 4. Extended feature map for learning nonlinear input-output mappings. The input-output mapping $V \rightarrow U$ is defined by a collection of locally valid linear mappings, one for the vicinity (shaded subregion in V) of the (input) weight vector $\mathbf{w}_s^{(in)}$ of each neuron. Defining these mappings requires an additional weight vector $\mathbf{w}_s^{(out)}$ and a matrix \mathbf{A}_s (not shown) per neuron. These additional parameters are adapted with a perceptron error correction rule.

dicularly. In addition, the orientation map exhibits several types of singularities that tend to cluster along the monocular regions between adjacent stripes of opposite binocularity. It turns out that all these features, including even many quantitative aspects, can be remarkably well reproduced by the Kohonen feature map algorithm (Erwin, Obermayer, and Schulen, 1995). One may conclude that despite its computational simplicity, the Kohonen feature map algorithm can successfully model a striking range of features of observed brain maps.

Variants of the basic SOFM algorithm. The basic SOFM algorithm is an on-line learning method that admits the formulation of an analogous batch variant (batch SOFM; Kohonen, 1995) in which adaptation steps occur only after entire epochs, during which all (or a larger number of) patterns have been presented. When T_s denotes the subset of patterns for which neuron s was selected as winner in the current epoch, the batch SOFM adaptation step at the end of the epoch is given by

$$\mathbf{w}_r^{(new)} = \frac{\sum_{\mathbf{x} \in T_s} h_{rs} \cdot \mathbf{x}}{\sum_{\mathbf{x} \in T_s} h_{rs}} \quad (6)$$

The advantages of the batch algorithm are the absence of any learning rate and its enlarged possibilities for various optimizations to allow the training also of very large maps and for very large data sets (for details, see Kohonen, 1995).

One major possibility for reducing the number of nodes, as required, e.g., for higher dimensional maps, is the use of *linear interpolation*. In particular, the extension of the basic feature map to produce a vectorial output value from a *locally valid linear map*

$$\mathbf{y}_r(\mathbf{x}) = \mathbf{w}_r^{(out)} + \mathbf{A}_r(\mathbf{x} - \mathbf{w}_r^{(in)}) \quad (7)$$

(Ritter et al., 1992) has proved useful in many contexts. Here, the additional quantities are *output weights* $\mathbf{w}_r^{(out)}$ and matrices \mathbf{A}_r . Selection of a winner neuron s and adaptation of the input weights $\mathbf{w}_r^{(in)}$ proceed as in the original SOFM algorithm, while the output weights $\mathbf{w}_r^{(out)}$ and \mathbf{A} can be trained with a perceptron rule.

A Bayesian generalization of the SOFM has been proposed and analyzed by Luttrell (1994). Here, both the winner selection $\mathbf{x} \rightarrow$

$s(\mathbf{x})$ and the decoding $s \rightarrow \mathbf{w}_s$ are replaced by probabilistic mappings, leading to generalized SOFM versions that can be connected with channel coding theory. The resulting formalism has been used to formulate a SOFM variant for the mapping of *distance data* (Graepel and Obermayer, 1999), for which no explicit pattern vectors but only pairwise distances are given as training information. Another idea from statistics, the use of a generative model as a departure point for the generation of topographic maps, has led to the GTM method (Bishop, Svensén, and Williams, 1997). Here the map is derived from a probability density that is concentrated along a low-dimensional, parametrized manifold whose parameters are estimated by the maximum likelihood approach.

A good collection of articles on major recent developments on the SOFM is contained in Oja and Kaski (1999).

Applications of the feature map algorithm. Artificial feature maps have proved useful in many pattern recognition applications. The classic example is the *Neural Typewriter* of Kohonen, where a feature map is used to create a map of phoneme space for subsequent speech recognition. Subsequent work has demonstrated the possibility of creating feature maps of language data that are ordered according to higher-level, semantic categories (Ritter and Kohonen, 1989) and of deriving very large-scale maps to support navigation in very large databases, such as the Internet. Other applications of the feature map include process control, image compression, time series prediction, optimization, generation of noise-resistant codes, synthesis of digital systems, and robot learning. Kohonen's *Self-Organizing Maps* (1995) contains a good survey of SOFM applications, with additional comments on SOFM uses in an Internet database.

Road Maps: Grounding Models of Networks; Learning in Artificial Networks

Related Reading: Competitive Learning; Data Clustering and Learning; Hebbian Synaptic Plasticity; Learning Vector Quantization; Principal Component Analysis

References

- Bishop, C. M., Svensén, M., and Williams, C. K. I., 1997, GTM: The generative topographic mapping, *Neural Computat.*, 10:215–234.
- Cottrell, M., Fort, J., and Pages, G., 1998, Theoretical aspects of the SOM algorithm, *Neurocomputing*, 21:119–138.
- Erwin, E., Obermayer, K., and Schulten, K., 1992a, Self-organizing maps: Stationary states, metastability and convergence rate, *Biol. Cybern.*, 67:35–45.
- Erwin, E., Obermayer, K., and Schulten, K., 1992b, Self-organizing maps: Ordering, convergence properties and energy functions, *Biol. Cybern.*, 67:47–55.
- Erwin, E., Obermayer, K., and Schulten, K., 1995, Models of orientation and ocular dominance columns in the visual cortex: A critical comparison, *Neural Computat.*, 7:425–468.
- Graepel, T., and Obermayer, K., 1999, A stochastic self-organizing map for proximity data, *Neural Computat.*, 11:139–155.
- Heskes, T., and Kappen, B., 1993, Error potentials for self-organization, in *Proceedings of an International Conference on Neural Networks*, vol. 3, New York: IEEE, pp. 1219–1223.
- Kohonen, T., 1982, Self-organized formation of topologically correct feature maps, *Biol. Cybern.*, 43:59–69.
- Kohonen, T., 1990, The self-organizing map, *Proc. IEEE*, 78:1464–1480. ♦
- Kohonen, T., 1995, *Self-Organizing Maps*, 2nd ed., Berlin: Springer-Verlag. ♦
- Luttrell, S. P., 1994, A Bayesian analysis of self-organizing maps, *Neural Computat.*, 6:767–794.
- Obermayer, K., Blasdel, G., and Schulten, K., 1992, A statistical mechanical analysis of self-organization and pattern formation during the development of visual maps, *Phys. Rev. A*, 45:7568–7589.
- Oja, E., and Kaski, S., Eds., 1999, *Kohonen Maps*, Amsterdam: Elsevier.
- Ritter, H., 1991, Asymptotic level density for a class of vector quantization processes, *IEEE Trans. Neural Netw.*, 2:173–175.
- Ritter, H., and Kohonen, T., 1989, Self-organizing semantic maps, *Biol. Cybern.*, 61:241–254.
- Ritter, H., Martinez, T., and Schulten, K., 1992, *Neural Computation and Self-Organizing Maps*, Reading, MA: Addison-Wesley. ♦

Semantic Networks

John A. Barnden, Mark G. Lee, and Manuela Viezzer

Introduction

Semantic networks (SNs) are a way of representing information abstractly by means of a graphical notation that makes use of interconnected nodes and arcs. They are commonly used in symbolic cognitive science. There are interesting similarities and differences between SNs and neural networks (NNs). Some attempts have been made to implement or emulate SNs in NNs and to form hybrid SN-NN systems.

SNs were originally developed for couching “semantic” information, either in the psychologist’s sense of static information about concepts or in the semanticist’s sense of the meanings of natural language sentences. However, they are also used as a general knowledge representation tool. The more elaborate types of SNs are similar in their representational abilities to sophisticated forms of symbolic logic. (For more information on SNs, see Sowa, 1984; Rich and Knight, 1991, chaps 4, 9–11; Lehmann, 1992.)

The Nature of Semantic Networks

As with NNs, there are many different styles of SNs. However, all SNs share the following general features:

- The nodes are to be interpreted (by us) as standing for physical or nonphysical entities in the world, classes or kinds of entities, relationships, or concepts.
- The links, which are almost always directed, encode specific relationships between entities, concepts, etc., where the type of the relationship is specified by a symbolic label on the link.

We roughly characterize SNs as being either *restricted* or *general*, although there are cases in between. The restricted class is typified by the small fragment shown in Figure 1. Nodes represent kinds, activities, or individuals (all named in the diagram by lowercase labels). IS-A links and INST links are the characteristic links of restricted SNs. Other links, like MAIN-LOCOMOTION in the fragment, are usually called roles and used to represent properties of the kinds. IS-A is the subkind relationship, whereas INST is the instance (i.e., individual-to-kind) relationship. IS-A links are often labeled as A-KIND-OF or AKO links. The individuals in the fragment are Canny and Osten, whereas the kinds are canary, ostrich, bird, and animal.

The central purpose of a restricted network is to organize concepts, kinds, or classes into a taxonomy, which may or may not be

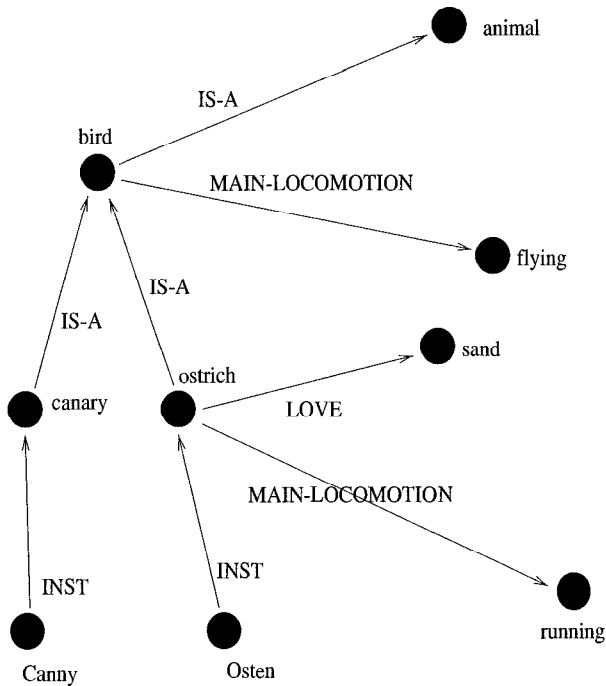


Figure 1. Fragment of a semantic network, restricted class (see text).

strictly hierarchical, and to state properties of the included entities. A general principle is the following:

- Information general to a class, kind, or concept should be held at the node representing it, and not repeated at nodes for sub-classes, subkinds, or subconcepts (or particular instances of them); instead, these subentities implicitly *inherit* the information at the former node (inheritance principle).

An allied fundamental principle is the following:

- Information attached to a node can contradict information attached to an ancestor node; in that case the former information overrides the latter (default-overriding principle).

Here, an ancestor node of a node *N* is a node that can be reached from *N* by a succession of IS-A links, or by an INST link and then possibly some IS-A links. Thus, the fragment in Figure 1 says that birds (in general) have flying as their main method of locomotion, but that for ostriches (in general), the main method of locomotion is running. On the other hand, canaries in general, and Canny in particular, implicitly inherit the flying because no node between those nodes and the bird node has a MAIN-LOCOMOTION link contradicting the flying. Osten inherits the running of ostriches in general, and their love of sand.

The general type of SN is typified by the fragment shown in Figure 2. The *q* node represents the proposition that every pig loves rain. Alternatively, it represents the situation of every pig loving rain. The fragment shown is a direct analogue of the logic formula

$$(\forall x) \text{ is-pig}(x) \Rightarrow \text{loves}(x, \text{rain})$$

Although general SNs often contain taxonomic information, this information is less important to the overall purpose of the network.

Rather, the focus is on representing entire sentences, propositions, or situations using a network format instead of an algebraic one. The idea goes back to Frege's *Begriffsschrift* and Peirce's relational graphs, and has been exploited to express the semantics of natural language, as, for example, in discourse representation theory (Kamp and Reyle, 1993).

In contrast to restricted networks, general networks usually have only a small selection of link labels, typified by those in Figure 2. Often, labels have a close connection to deep case relationships in natural language semantic theories (see the AGENT and OBJECT links in Figure 2). Relationships such as loving are now represented by nodes (see the *loves* node in Figure 2) rather than by link labels (see the LOVE link in Figure 1). Also, the more elaborate general SNs have facilities analogous to the connectives and quantifiers of formal logic (see the *implication* and *forall* nodes in Figure 2). The QUANT link to the *forall* node says that the *q* node represents a universally quantified proposition. The VAR link points to the *x* node, playing the role of the variable *x* in the formula above. The BODY link points to the body of the proposition, which has the form of an implication proposition (*i* node). ANTE stands for antecedent, and CONSE represents consequent. PRED links point from nodes representing simple propositions to the nodes for the predicates in the propositions. ARG stands for argument.

One of the problems to be handled when SNs are used to represent propositions is to express correctly the scope of logical operators. This can be achieved by including explicit nodes to represent propositions, like node *q* in Figure 2, thus turning the SN into what is often called a *propositional semantic network*. The first propositional semantic network to be implemented was the MIND system, by Stuart Shapiro, which then evolved into the SNePS system (see Shapiro and Rapaport, 1992).

Thus, formal logic expressions can be recast as SNs. However, the two forms of representation are not equivalent, partly because SNs implicitly involve implementational assumptions that facilitate some particularly important operations (see below), whereas formal

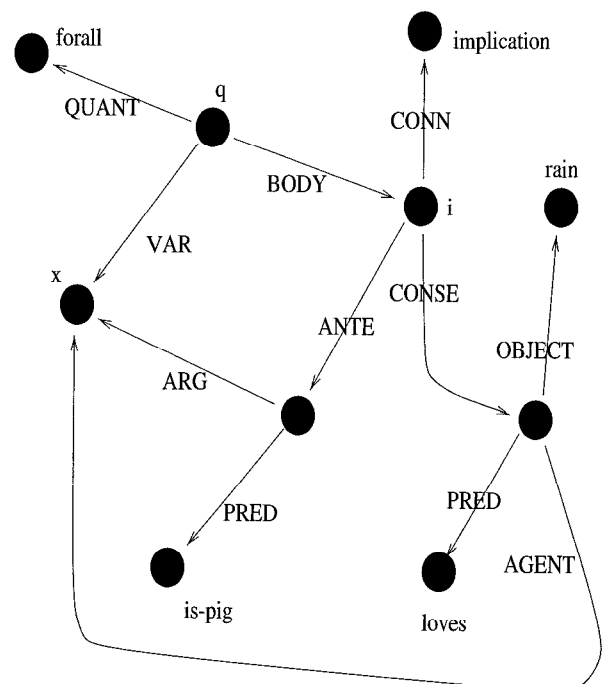


Figure 2. Fragment of a semantic network, general class (see text).

logic is devoid of implementational assumptions. Also, to parallel the inheritance-blocking features of SNs in logic, one has to turn to special, advanced forms of logic (for an introduction, see Rich and Knight, 1991, chap. 7; see also Brachman et al., 1991). Moreover, several SN approaches have made radical claims as to what should and what should not be represented. For example, the SNePS family of SNs argues for a totally intensional representation of knowledge (Shapiro and Rapaport, 1992).

Among current logical systems, description logics capture the features of restricted SNs and are used to reason about taxonomic information. Some versions of description logic support nonmonotonic reasoning and belief revision to cope with the conflicting information that may result from applications of the overriding defaults principle (see Brachman et al., 1991).

Processing in Semantic Networks

Sometimes SNs are used as static data structures that change only when external programs manipulate them. However, it is possible to embed some processing mechanisms in the networks themselves, thus giving SNs a self-modifying ability similar to that of NNs.

One common and basic type of processing in SNs is *spreading activation*. During the course of spreading activation, if one node is active at some moment, then the nodes directly connected to it can become activated. A given episode of activation spreading might only pass activation across links of specific types. Activation here is often a yes-no matter, as opposed to the graded activation typically used in NNs. However, graded activation can be used in SNs.

A common variant of activation spreading is *marker passing*. Markers are symbolic objects, usually of several different types. The markers that a node send out depend in some possibly complex way on the markers it receives from other nodes. Markers are often simple symbolic objects, bearing no information except their type. However, more complex markers are used in some SNs; for instance, a marker can contain information on its node of origin or the nodes or links it has passed through. Complex markers are used in the system of Charniak (1986), for example. A marker can have a graded energy level that affects its ability to cause a receiving node to produce a marker in response (see the system of Hendler in Barnden and Pollack, 1991).

The main motivation for activation spread or marker passing is to allow *intersection search*. For instance, consider an SN system used in understanding natural language text. Marker passing might start with nodes representing the word senses of various words in the text. The collisions of markers at nodes means that paths have been found between word senses. This information can be used to help disambiguate the words (see Yu and Simmons, 1990). Marker passing can also be used to manage simple inheritance reasoning. For instance, in Figure 1, Osten's main locomotion method could be found by sending out a marker from the Osten node and constraining it to travel only along a chain of IS-A links, possibly preceded by a single INST link, and followed by one MAIN-LOCOMOTION link. If the system prefers answers found first to answers found later, the *running* answer is preferred over the *flying* answer, thereby effecting the desired handling of exceptions (blocking of inheritance).

Marker passing is important, but is generally used only for restricted purposes. In general networks that handle more elaborate reasoning, the brunt of the inferencing is typically done by sub-network matching and construction processes (see Cravo and Martins, 1993, for an advanced implemented system).

In any SN with INST and IS-A links, inheritance reasoning is simpler and quicker than it would otherwise be. In SNs without the special links, or in ordinary logical frameworks, traversal of taxo-

nomic relationships is a more elaborate process. These observations rely on certain strong implementational assumptions that are generally made about SNs, though usually only tacitly. The assumptions apply both to computer implementations of SNs and to hypothesized realizations of SNs in the brain. One of the assumptions is that, if a processing mechanism is attending to a given node N, then it can efficiently transfer attention to nodes connected by single links to N.

Inheritance reasoning can also be given a probabilistic interpretation where every arc between nodes in the SN is given a probability from 1.0 to 0.0 of being traversed. Such networks are termed Bayesian. Bayesian networks are used in reasoning with uncertainty. The probabilities associated with individual arcs can be automatically learned from data. Pearl (2000) gives an extensive overview of such networks.

Finally, SNs can have *attached procedures or rules*. These are algorithms, expressed in some symbolic format, that are attached to nodes or links, and they typically have localized effects supporting particular inference functions. (For a brief introduction to this topic, see Rich and Knight, 1991, chap. 4 and section 10.3.2.)

Marker passing and attached procedures can also be combined, as in Petri nets, where there are passive nodes, called places, active nodes, called transitions, and sets of rules for marking places with tokens and for executing, or firing, transitions.

Contrasts to Neural Networks

In NNs, in contrast to the case of general SNs, activation spread is almost always the only mechanism for short-term computation (as opposed to long-term computation, such as slow adaptation). Moreover, the types of activation spread and marker passing in SNs are typically more complex than activation spread in NNs, especially when complex markers are used. In strong contrast to almost all NNs, the topology of an SN, especially a general SN, can be changed by processing mechanisms in arbitrarily extensive and complex ways in the short term.

NN links do not have type labels. As a result, NNs do not treat different input links differently for the most part. Instead, a weighted sum of the input values is formed, submerging the identities of different links. The most salient exception to this is the differentiation of input links into excitatory and inhibitory links, but this distinction is crude compared with the link differentiation by labels in SNs. Nevertheless, NN links that impinge on other links and modulate them (see NEUROMODULATION IN INVERTEBRATE NERVOUS SYSTEMS) can be used to obtain many of the effects of link typing in SNs.

When an NN link is directed, activation can spread in only one direction across the link. It is less common to impose this constraint in SNs. It is therefore easier in SNs to express asymmetric relationships between nodes without consequently constraining the accessibility of the nodes from each other. As an example of the problem raised for NNs, consider the use of symmetric links in some artificial NNs, notably Hopfield nets (see STATISTICAL MECHANICS OF NEURAL NETWORKS). These allow activation spread in both directions, but do not directly support conceptually asymmetric relationships. More indirect methods are needed to handle such relationships (see Barnden and Srinivas, 1991).

As a result of all these differences, SNs cope much more readily with many of the representational and processing needs that arise in high-level cognitive activities such as commonsense reasoning and natural language understanding. The needs in question are listed in ARTIFICIAL INTELLIGENCE AND NEURAL NETWORKS (q.v.). NNs have difficulty in rapidly creating new, complex bodies of information, and in structurally matching complex, and possibly highly temporary, bodies of information. Certainly, any direct par-

allel of subnetwork matching and subnetwork creation in SNs is difficult to achieve in typical NNs (but see DYNAMIC LINK ARCHITECTURE). However, some NN systems have been developed that can directly implement SNs (see the next section). They depart from mainstream NNs in that they have more elaborate and specialized structure.

On the other hand, NNs appear to have advantages with respect to other needs posed by high-level cognition (see ARTIFICIAL INTELLIGENCE AND NEURAL NETWORKS). These other needs include context sensitivity, graceful degradation, and automatic adaptation. A contrastive factor here is that link weights are fundamental in NNs but relatively uncommon in SNs. This difference confers a graded associativity quality on NNs that has no direct parallel in most SNs, although the length of a path between two SN nodes has often been assumed to encode the strength of association between them: the shorter the path, the stronger the association.

Bringing Semantic and Neural Networks Together

Several researchers have overtly used NNs to implement or approximately emulate SNs. Other work has aimed at mixed systems that combine NNs with SNs or are compromises between the two. Major examples of the implementation/emulation type of work can be found in Hinton (1989) and in the chapters by Barnden, by Bookman and Alterman, and by Diederich, Dyer, and Shastri in Barnden and Pollack (1991). Touretzky (1990) describes a framework that is not aimed specifically at semantic networks but that could readily implement them. Because of the complexity of the systems that implement or emulate SNs, we omit description of them here.

For examples of combined SN-NN systems, see the chapters by Hendler and Lehnert in Barnden and Pollack (1991). In the Hendler case, nodes in the SN can also serve as input-output nodes in an NN. Markers passed to such SN nodes cause them to inject neural activation into the NN, and vice versa. The symbolic markers have a numeric strength, and this affects the level of neural activation instigated by a marker. The NN acts as an intermediary between SN nodes, allowing an SN node representing one concept to stimulate an SN node encoding a similar concept. The NN is trained by backpropagation to associate concepts with sets of low-level features. Concepts are thereby viewed as similar according to the extent to which they share features. Hendler's system makes symbolic reasoning less rigid by enriching it with similarity-based reasoning.

For compromises between SNs and NNs, see the chapters by Eskridge and Kokinov in Holyoak and Barnden (1994). Unlike Hendler's system, these systems are not divided into separate neural and semantic parts. In Eskridge's system, nodes communicate by means of numerical activation signals, much as in NNs, but symbolic markers travel along with the numerical activation, the links have labels, activation can be constrained to spread only over links with specified labels, links can be dynamically created, and specialized SN processing rules can be invoked. The system performs complex analogical processing, with the symbolic aspects coping with complex structure manipulation and the neural aspects helping with retrieval of source analogues and with weighing of alternatives.

Discussion

NNs and SNs have much in common and should be regarded as two points in a rich, quasi-continuous space of computational architectures rather than as radically different types of network. There are important differences in the nature and usage of links and in the degree to which computation can be thought of as local to individual nodes (although in restricted SNs the computation can be as local as it is in NNs). There are various ways of implementing or emulating SNs in NNs, and of forming hybrid SN-NN systems.

Road Map: Artificial Intelligence

Related Reading: Artificial Intelligence and Neural Networks; Bayesian Networks; Competitive Queuing for Planning and Serial Performance; Connectionist and Symbolic Representations; Graphical Models: Probabilistic Inference; Structured Connectionist Models

References

- Barnden, J. A., and Pollack, J. B., Eds., 1991, *Advances in Connectionist and Neural Computation Theory*, Vol. 1: *High Level Connectionist Models*, Norwood, NJ: Ablex.
- Barnden, J. A., and Srinivas, K., 1991, Encoding techniques for complex information structures in connectionist systems, *Connect. Sci.*, 3:263–309.
- Brachman, R. J., McGuinness, D. L., Patel-Schneider, P. F., Resnik, A. L., and Borgida, A., 1991, Living with Classic: When and how to use a KL-ONE-like language, in *Principles of Semantic Networks: Explorations in the Representation of Knowledge* (J. F. Sowa, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 401–456.
- Charniak, E., 1986, A neat theory of marker passing, in *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, Los Altos, CA: Morgan Kaufmann, pp. 584–588.
- Cravo, M. R., and Martins, J. P., 1993, SNePSwD: A newcomer to the SNePS family, *J. Exp. Theoret. Artif. Intell.*, 5:135–148.
- Hinton, G. E., 1989, Implementing semantic networks in parallel hardware, in *Parallel Models of Associative Memory*, updated edition (G. E. Hinton and J. A. Anderson, Eds), Hillsdale, NJ: Erlbaum, pp. 191–221.
- Holyoak, K. J., and Barnden, J. A., Eds., 1994, *Advances in Connectionist and Neural Computation Theory*, Vol. 2: *Analogical Connections*, Norwood, NJ: Ablex.
- Kamp, H., and Reyle, U., 1993, *From Discourse to Logic*, Dordrecht: Kluwer.
- Lehmann, F. W., Ed., 1992, *Semantic Networks in Artificial Intelligence*, New York: Pergamon Press.
- Pearl, J., 2000, *Causality: Models, Reasoning and Inference*, Cambridge, Engl.: Cambridge University Press.
- Rich, E., and Knight, K., 1991, *Artificial Intelligence*, 2nd ed., New York: McGraw-Hill. ♦
- Shapiro, S. C., and Rapaport, W. J., 1992, The SNePS family, in Lehmann, F. W., 1992, *Semantic Networks in Artificial Intelligence*, New York: Pergamon Press.
- Sowa, J. F., 1984, *Principles of Conceptual Structures: Information Processing in Mind and Machine*, Reading, MA: Addison-Wesley.
- Touretzky, D. S., 1990, BoltzCONS: Dynamic symbol structures in a connectionist network, *Artif. Intell.*, 46:5–46.
- Yu, Y.-H., and Simmons, R. F., 1990, Truly parallel understanding of text, in *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, Menlo Park, CA: AAAI Press, pp. 996–1001.

Sensor Fusion

Allen M. Waxman

Introduction

Sensor fusion is not the sole purview of the modern brain but rather an old and common strategy in nature, and of great relevance to a variety of applications. It is a means by which multiple sensing modalities and interacting modality-specific processing streams can cue one another, enhance or depress cross-modality responses to stimuli, generate cross-modality expectations, and provide complementary evidence supporting or negating a decision (e.g., the detection of prey, or the recognition of an object). Studies of sensory interactions in flatworms, crustaceans, insects, and fishes (see references in Stein and Meredith, 1993), snakes (Newman and Hartline, 1982), cats and monkeys (Stein and Meredith, 1993; Stein, Wallace, and Stanford, 1999), and humans (Stein and Meredith, 1993; Giard and Peronnet, 1999; Shimojo and Shams, 2001) point to multiple levels of sensory integration in brain. From multimodal sensory receptors to midbrain organs (i.e., tectum or superior colliculus) to corticotectal feedback pathways and temporal-prefrontal lobe activation, it is clear that sensor fusion impacts an animal's ability to detect and track objects (i.e., its attentive and orienting behaviors) as well as its decision-making abilities (i.e., to recognize an object from its multimodal signature). Thus, the many sensory pathways that have developed in living systems to process chemical, mechanical, thermal, visual, and auditory stimuli have generally formed two classes of pathways that also project to one another, one that is modality specific and one that is integrated across sensory modalities.

Sensor fusion is of direct relevance to the military defense (Hall and Llinas, 1997), intelligence, remote sensing (Pohl and van Genderen, 1998), medical, robotics, and manufacturing communities. Multiple specialized sensors have been developed to provide the data necessary for decision making in each of these application areas. Yet it is often acknowledged that today's information shortcomings can be overcome, not necessarily with still more and better sensors, but rather with tomorrow's sensor fusion systems. Witness the multitude of methods, systems, and applications for sensor fusion that are reported annually at conferences such as the MSS (formerly IRIS) National Symposium on Sensor and Data Fusion 1988–2002, the International Conference on Information Fusion 1998–2002, and the SPIE conference on Sensor Fusion: Architectures, Algorithms, and Applications 1997–2002. However, approaches motivated by or derived from biological sensor fusion strategies have been embraced by relatively few (Toet, van Ruyven, and Valetton, 1989; Anastasio, Patton, and Belkacem-Boussaid, 2000; Fay et al., 2000; Fisher et al., 2001; Waxman et al., 1995, 1997, 2001).

Biological Insights

There is an extensive and growing literature on sensor fusion in biological systems. Methods of observation in the living animal include electrophysiological recording from single cells, event-related potential (ERP) monitoring from arrays of scalp electrodes, functional magnetic resonance imaging (fMRI) and position emission tomography (PET) of the living brain, and direct psychophysical experiments on humans. Some of the most significant studies lead to the following insights.

Snakes

Newman and Hartline (1982) have studied the integration of visual and thermal infrared signals in the optic tectum of the rattlesnake

and python. Both visual signals from the retina and infrared signals from the pit organs project onto the tectum in orderly maps of sensory space (i.e., retinotopically) that are in register with one another. Newman and Hartline identified six classes of bimodal neurons that display highly nonlinear multimodal interactions, including AND cells, OR cells, response-enhanced cells (e.g., cells in which the response to a visual stimulus is enhanced in the presence of an infrared stimulus), and response-depressed cells (e.g., cells in which the response to an infrared stimulus is depressed in the presence of a visual stimulus). These single-cell responses indicate that both cooperative and competitive nonlinear processes are at play, all in the service of orienting the snake toward its potential prey. They are, in fact, reminiscent of opponent-color multispectral interactions observed in primate retinal ganglion cells (Waxman et al., 1997).

Cats and Monkeys

In mammals, the tectum has evolved into the superior colliculus (SC). A midbrain organ involved in attentive and orienting behavior, it aims to *keep your eye on the ball*. The definitive works of Stein and Meredith (1993) and Stein et al. (1999) make clear that the seven-layered SC receives direct sensory input to its deep layers from visual, auditory, and somatosensory neurons, as well as significant feedback from single-modality cortical areas, and that it projects up to higher centers via thalamus and down to brainstem and spinal cord motor pathways in order to control the pointing of sensory organs (see Stein and Meredith, 1993, for anatomical drawings of the SC). These sensory and cortical inputs all form topographically registered maps in the deep layers, in a coordinate system tied to the moving retina (requiring the existence of learned transformations between retinal, head, and body coordinates). Moreover, one also finds motor maps (of vector displacement) in registration as well, providing all sensory modalities common access to the machinery that turns the eyes, ears, and head in order to find and track that target object. More than 50% of deep-layer neurons are multisensory (bimodal and trimodal), exhibiting both cross-modal response enhancement and response depression. These neurons have receptive fields, often with center-surround spatial profiles and noncoincident but overlapping temporal profiles. These cells are well designed (in fact, they develop postnatally, from experience) to detect complementary evidence supporting or negating the detection and localization of targets from their multimodal signatures.

An interesting and significant phenomenon observed among multisensory neurons in SC is that of *inverse effectiveness* in cross-modal response enhancement. Multisensory neurons that respond weakly to unimodal stimuli can be greatly enhanced in their response to multimodal stimuli, in contrast to the modest enhancement observed among multisensory neurons that respond strongly to unimodal stimuli. This effect has been shown to be consistent with the interpretation that multisensory neurons in SC compute the conditional probability (in the sense of Bayesian estimation theory) that a target is present, given the unimodal or multimodal stimulus (Anastasio et al., 2000). However, it is clear from electrophysiological studies that response enhancement in multisensory neurons is due to cortical feedback to the SC that develops after birth (Stein et al., 1999).

Humans

It thus seems likely that the human SC is responsible for our being fooled by both the ventriloquist and the television set. Interactions

between visual and auditory activations on registered maps in SC convince us that the sound is actually coming from the mouths of those dummies! Even our perception of human articulated sounds is strongly influenced by the visual shape of moving lips (i.e., the McGurk effect). And what we think we see is easily altered by the sounds we hear, even for very simple stimuli such as flashes and beeps. Shimojo and Shams (2001) review this and many other cross-modality target detection and tracking experiments in which cortical activation is monitored via fMRI and ERP measurements. They note the significance of transient/discontinuous stimuli, regardless of modality, and the apparent ability of these stimuli to influence multimodal perceptions.

Beyond orienting behaviors devised to detect, localize, and track objects, when it comes to the task of object recognition from multimodal signatures, we humans also employ a good deal of cortex while preserving old successful strategies. Observations of ERPs (Giard and Peronnet, 1999) reveal signs of cross-modality response enhancement and response depression as well as new neural activity (not seen with single-modality stimuli derived from the object) in the right frontotemporal area, generally considered an associative area of brain.

Technological Implementations

A great deal of work has been done on the military problem of tracking multiple targets from multiple sensors using Bayesian statistical inference methods, and cast in the context of higher levels of information fusion (Hall and Llinas, 1997). The information-theoretic concept of mutual information, in conjunction with perceptron neural networks, has recently been applied to audiovisual fusion and speaker localization (Fisher et al., 2001).

With application to remote sensing and environmental monitoring, it is commonplace to utilize imagery derived from multiple airborne and space-based sensors (Pohl and van Genderen, 1998). However, most image fusion methods employed are based on global statistical methods (e.g., PRINCIPAL COMPONENT ANALYSIS, q.v.) and false color overlay of separate modalities. These approaches do not derive or benefit from biological approaches to sensor fusion. Early methods for fusion of two complementary imaging modalities, in particular visible (reflected) and thermal (emitted) infrared imagery, also did not utilize biological fusion strategies, but did build on known multiresolution representations in visual processing (Toet et al., 1989). A successful approach to visible-infrared image fusion based on single-opponent color processing in the retina was developed by Waxman et al. (1997) and enabled a color night vision technology. This approach was extended to accommodate three and four imaging modalities by means of double-opponent color processing (as in primary visual cortex), and a target learning and recognition capability based on the fused sensor signature was incorporated (Fay et al., 2000). Biological architectures for visual 3D object learning and recognition, which fuse evidence over multiple views in an *aspect network* and apply to multiple imaging modalities, are summarized in Waxman et al. (1995).

Real-Time Night Vision Enhancement

Figure 1 illustrates an opponent-color architecture for fusing multiple images in real time. It requires that the imagery be collected in an optically registered manner, or separately registered to a common reference frame. Shown here are three modalities of wide dynamic range imagery, low-light visible (VIS), short-wave infrared (SWIR), and thermally emitted long-wave infrared (LWIR), that provide different views into the night. These distinct modalities are fused into a single natural color image (shown here in gray scale,

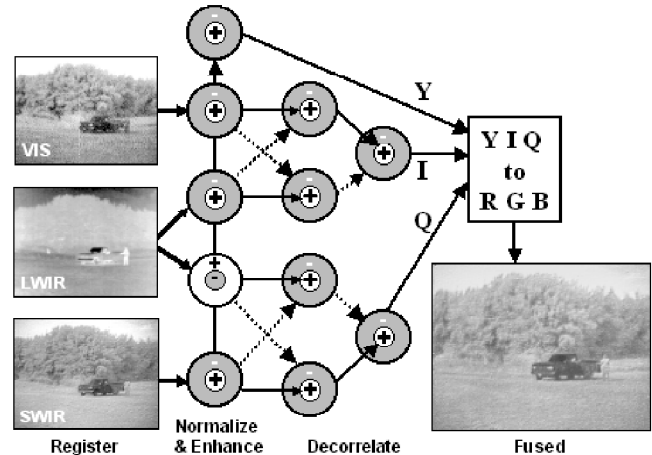


Figure 1. Color night vision through the real-time fusion of multiple imaging sensors (visible, short-wave, and long-wave infrared). Contrast enhancing *on-* and *off-channel* and *opponent-color* receptive fields are built from center-surround shunting neural networks. (See Fay et al., 2000, for color rendering of the fused output image.)

but see Fay et al., 2000, for several color renderings). Multiple center-surround shunting neural networks are used to create enhanced *on-* and *off-center* (e.g., hot and cold LWIR) responses, as well as *single-* and *double-opponent* cross-modality contrasts that serve to decorrelate the input imagery. The outputs of the network are mapped to the human opponent-color YIQ representation and converted to RGB color for display. Similar networks comprised of a variety of such opponent-color fields have been used to fuse two, three, and four imaging modalities, and have been further combined with 3D imagery obtained from a lidar in real-time. Using fuzzy ARTMAP neural networks (Carpenter et al., 1992) in conjunction with a target designation interface, we incorporated field-adaptive target learning and recognition based on fused image data.

Multisensor Fusion and Exploitation

Imagery and signals collected by multiple platforms over a common geospatial area can be brought into registration by means of a 3D site model (i.e., terrain data and building models), as indicated in Figure 2 (Waxman et al., 2001). Forming a layered database of multiple registered modalities, opponent-sensor image fusion (see Figure 1) and spatial feature extraction (oriented edges, extended boundary contours, texture measures, and 3D slopes and curvatures) support interactive 3D visualization and also augment the database. The layered data are well organized for interactive pattern learning and search, for each pixel now corresponds to a complete feature vector. With the aid of a graphical user interface, an analyst selects example and counterexample (context) pixels of the object of interest, and, treating these as training data, a *search agent* is created using fuzzy ARTMAP neural networks. Additionally, the agent sorts the many elements of the feature vector and identifies those features that are salient to the target in context. The agent then conducts a first-pass search for the object of interest (e.g., roads, buildings, vehicles) in the fused imagery. Subsequent searches will use these detections as context in guided search. A separate search agent is trained for each object of interest in the fused data set. These same methods for image fusion and mining have been successfully applied to spectral MRI and MRI/SPECT in medical imaging.

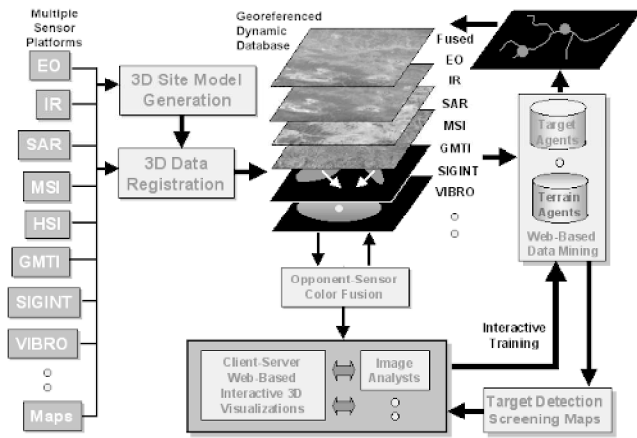


Figure 2. Architecture for fusion and exploitation (i.e., data mining) of multisensor data in a geospatial context. (A prototype fusion system is described in Waxman et al., 2001.)

Discussion

Biological systems, from simple to complex, all demonstrate the capacity for fusing multiple sensing modalities. In humans, auditory-visual fused localization underlies the daily experience of watching television, in which speech seems to emanate from the visual image of the actor on the screen and not from the speakers beside the TV. Lessons learned from the rattlesnake visible-infrared fusion pathway and its similarity to opponent-color processing in the mammalian retina have led to real-time architectures for color-fused night vision. The existence of registered multimodal maps and bimodal neurons in mammalian SC, as well as the cortical feedback pathways to the SC, has led to an architecture and prototype system for fusion and exploitation of multisensor surveillance data in a geospatial context. Observed interactions in human fronto-temporal cortices reflect associative processes underlying multi-sensor object recognition.

Statistical, information-theoretic, and neural network methods of sensor fusion provide complementary tools for turning data into decisions in the context of knowledge. It remains to be seen if neural methods can rise to the challenge of higher levels of sensor and information fusion, for which knowledge-based systems are required. However, in our experience the unique insights obtained from biological system design yield distinct advantages in the development of adaptive sensor fusion systems for real-world problems.

Road Maps: Other Sensory Systems; Vision

Related Reading: Collicular Visuomotor Transformations for Gaze Control

References

- Anastasio, T. J., Patton, P. E., and Belkacem-Boussaid, K., 2000, Using Bayes' rule to model multisensory enhancement in the superior colliculus, *Neural Comput.*, 12:1165–1187.
- Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., and Rosen, D. B., 1992, Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Trans. Neural Netw.*, 3:698–713.
- Fay, D. A., Waxman, A. M., Aguilar, M., Ireland, D. B., Racamato, J. P., Ross, W. D., Streilein, W. W., and Braun, M. I., 2000, Fusion of 2-/3-/4-sensor imagery for visualization, target learning and search, *Enhanced Synthet. Vision 2000*, SPIE-4023:106–115.
- Fisher, J. W. III, Darrell, T., Freeman, W. T., and Viola, P., 2001, Learning joint statistical models for audio-visual fusion and segregation, in *Advances in Neural Information Processing Systems 13* (T. K. Leen, T. G. Dietterich, and V. Tresp, Eds.), Cambridge, MA: MIT Press, pp. 772–778.
- Giard, M. H., and Peronnet, F., 1999, Auditory-visual integration during multimodal object recognition in humans: A behavioral and electrophysiological study, *J. Cognit. Neurosci.*, 11:473–490.
- Hall, D. L., and Llinas, J., 1997, An introduction to multisensor data fusion, *Proc. IEEE*, 85:6–23. ♦
- Newman, E. A., and Hartline, P. H., 1982, The infrared vision of snakes, *Sci. Am.*, 246:116–127.
- Pohl, C., and van Genderen, J. L., 1998, Multisensor image fusion in remote sensing: Concepts, methods and applications, *Int. J. Remote Sens.*, 19:823–854. ♦
- Shimojo, S., and Shams, L., 2001, Sensory modalities are not separate modalities: Plasticity and interactions, *Curr. Opin. Neurobiol.*, 11:505–509. ♦
- Stein, B. E., and Meredith, M. A., 1993, *The Merging of the Senses*, Cambridge, MA: MIT Press. ♦
- Stein, B. E., Wallace, M. T., and Stanford, T. R., 1999, Merging sensory signals in the brain: The development of multisensory integration in the superior colliculus, in *The New Cognitive Neurosciences*, 2nd ed. (M. S. Gazzaniga, Ed.), Cambridge, MA: MIT Press, pp. 55–71. ♦
- Toet, A., van Ruyven, L. J., and Valetton, J. M., 1989, Merging thermal and visual images by a contrast pyramid, *Opt. Engn.*, 28:789–792.
- Waxman, A. M., Gove, A. N., Fay, D. A., Racamato, J. P., Carrick, J. E., Seibert, M. C., and Savoye, E. D., 1997, Color night vision: Opponent processing in the fusion of visible and IR imagery, *Neural Netw.*, 10:1–6.
- Waxman, A. M., Seibert, M., Gove, A. N., Fay, D. A., Bernardon, A. M., Lazott, C., Steele, W. R., and Cunningham, R. K., 1995, Neural processing of targets in visible, multispectral IR and SAR imagery, *Neural Netw.*, 8:1029–1051.
- Waxman, A. M., Verly, J., Fay, D. A., Liu, F., Braun, M. I., Pugliese, B., Ross, W. D., and Streilein, W. W., 2001, A prototype system for 3D color fusion and mining of multisensor/spectral imagery, in *Proceeds. of the Fourth International Conference on Information Fusion*, I-WeC1, pp. 3–10.

Sensorimotor Interactions and Central Pattern Generators

Avis H. Cohen and David L. Boothe

Introduction

Movement through the world requires that the environment be integrated and understood. An organism must navigate obstacles, seek sustenance, and avoid predators. Movement requires further that the organism integrate the visual, auditory, and other environ-

mental information within its machinery for locomotion. Each of us in our daily interactions with the world is familiar with the seamless way in which our nervous system combines sensory and motor information. Thus, we know in a simplistic sense that sensory and motor systems are integrated. How this integration is performed by a nervous system is currently poorly understood. Much

work has been done on motor and sensory systems in isolation. Less work has been done on how these systems interact. In this article we offer an interpretation of currently existing empirical evidence and argue from this evidence for some very basic properties of the biological systems performing sensorimotor integration.

A considerable amount of the experimentation on sensorimotor integration has focused on how sensory information modulates motor system activity. The flow of information commonly explored is from sensory to motor. Two main types of experiment have been performed to elucidate this relationship: reflex-type experiments, and studies of how sensory inputs modulate central pattern generators (CPGs). Peripheral feedback, including that from tactile receptors, muscle afferents, and other proprioceptive feedback, passes up to the somatosensory cortex, but first synapses within the spinal cord, where there are reflex circuits that respond to such inputs. Additionally and perhaps more functionally relevant, sensory input is filtered and processed locally, where the spinal pattern generator circuitry fits its response into the ongoing locomotion as appropriate. Thus, although the brain receives much of the sensory input, the responses to spinal inputs are first the responsibility of the local spinal circuitry. The first section of this article discusses the impact of sensory information on CPGs.

The influence of motor systems on sensory activity is less well characterized. Recent evidence shows that vertebrate motor systems can strongly influence sensory input. In the second section we show that the interplay between sensory feedback and motor output is more complex than is often suggested, and that behavior is strongly affected by movement and movement-related activity.

Studying motor and sensory systems in isolation, one can easily miss the massive interaction between these systems. An understanding of this interaction is crucial for explaining the simplest behaviors of an organism in its environment. What is most surprising about this interaction is not that it exists, but its pervasiveness. Interaction between motor and sensory systems exists from the first steps of sensory detection to the highest levels of processing. In the third section of this article we show that this type of two-way interaction also exists between the cortex and spinal cord.

The older and more usual view is that the spinal cord performs rhythmic and rapid reflexive-type behaviors that require the spinal circuitry and its speed locally, whereas more complex activities are the province of cortex or at least supraspinal centers. In this view, the descending systems send commands to lower-level systems, which respond appropriately. However, this distinction may not be so marked (Jankowska and Lundberg, 1981). Cortical commands now are viewed as integrated with CPG circuitry to produce volitional movements. We present evidence that the integration is considerably richer than the old view or even than the new view would suggest.

There is no doubt that cortical systems contribute to sensorimotor integration. What is in doubt is that motor cortex sends commands to a passively responsive spinal cord. Motor commands are acted on only as spinal circuits independently and thoroughly process all incoming information. In the view that we present, spinal cord and cortex are highly integrated. Movements result from an ongoing interactive process. The actual form that movements take is largely a product of spinal circuitry and its synergies, while the cortex is expected to perform other, more complex tasks that integrate somatosensory and motor systems with other inputs to which the cortex has unique access, such as vision, audition, and olfaction.

In such a limited review, it is inevitable that the material be selective. This review is not intended to be complete but rather to suggest the different types of evidence available.

Sensory to Motor Flow of Information

The most common view of sensory feedback is that it provides information regarding the position of the organism as it moves through space. Appropriate to this role, the feedback can correct a CPG on a cycle-by-cycle basis to maintain the organism in proper relationship to its environment (Rossignol, Lund, and Drew, 1988). For example, the hip joint angle of the cat can trigger a new step cycle as the body is propelled over its respective limb on the ground. Whelan and Pearson (1997) have also found that stretch of muscle spindles can trigger a new step cycle through contractions of appropriate muscles. In these ways, the cycle periods are able to accommodate changes in the velocity of the animal. Similarly, the bending of the tailfin in dogfish or lamprey entrains the swimming so that swim cycles are appropriate lengths for the environmental conditions. Thus, if the body is not able to adequately bend against a strong current, this will be compensated for by a longer cycle. This type of sensory regulation is accomplished at the level of the spinal cord and requires no descending input, although descending input will influence and weaken the responses if present (Whelan and Pearson, 1997). All CPGs must have some stimulus that can trigger or prolong a new cycle as necessary in order to guarantee that the CPG's movements are adaptive.

Another well-documented role for sensory feedback is to elicit reflexive responses to environmental perturbations. This is also accomplished at the spinal level, where sensory inputs are gated through the CPG during ongoing activity (Rossignol et al., 1988). A sensory stimulus can elicit phase-dependent responses that are quite unlike the reflex responses that such stimuli would induce in the absence of CPG activity. For example, an obstacle encountered by the paw dorsum will produce an enhanced flexion during the flexion phase of the step cycle, but it will produce an enhanced extension during the extension phase. This guarantees that the limb is properly supported at the moment it is raised to avoid the obstacle. The contralateral limb is also integrated with such responses. That is, the limb opposite the stimulus must be positioned to support the responsive limb before it will flex over such an obstacle (Hiebert et al., 1994). Such phase-dependent responses to perturbations are very common across CPGs. For each rhythmic movement there are classes of stimuli that elicit such responses (Rossignol et al., 1988). In the case of locomotion, the response requires no input from descending systems and is seen in spinal animals as well as intact animals.

Motor to Sensory Flow of Information

More recently, new evidence has accumulated to indicate that the relationship between the CPG and its sensory feedback is not unidirectional. Dubuc and his colleagues (Nussbaum et al., 1996) first found that there is a feedforward signal coming from the CPG that can be recorded as a dorsal root potential. This is a phasic presynaptic modulation of the sensory terminals in the spinal cord. The modulation is sufficient in some cases to trigger action potentials in the sensory fibers even during fictive locomotion, that is, locomotor pattern generator activity in the absence of movement (reviewed in Nussbaum et al., 1996). Thus, the signals cannot be coming from any source other than the CPG. A similar phenomenon has been found in cockroach flight, in locust walking, and in lamprey swimming. This phenomenon thus seems to be quite general, occurring across a wide range of animals from invertebrates and vertebrates and from cats to lampreys (reviewed in Cohen, 1992).

In the cockroach, the role of phasic modulation seems quite straightforward to interpret. It apparently provides the cirral fibers with activity that is phase locked to the flight as a kind of preemphatic sensory input during the movement. The situation in cat is

more problematic, and the role of presynaptic modulation remains somewhat uncertain. Nussbaum et al. (1996) postulate three possible mechanisms by which the presynaptic modulation, acting as presynaptic inhibition, could influence the activity of primary sensory afferents. First, presynaptic inhibition could reduce the amount of transmitter being released presynaptically. This presynaptic inhibition would reduce the strength of the postsynaptic signal. Second, presynaptic activity could elicit antidromic action potentials in the primary sensory afferent axon. These antidromic action potentials could interfere with orthodromic discharges directly in the axon, or they could desensitize the axon at the first node of Ranvier. This antidromic stimulation would therefore also reduce the influence of primary sensory afferents on postsynaptic cells. Third, presynaptic inhibition could conceivably cause orthodromic action potentials in other axonal collaterals. These orthodromic action potentials could then cause postsynaptic response elsewhere in the spinal cord. The existence of this third possibility is not well confirmed, nor is the overall functional impact of these effects clear.

Evidence has been accumulating for additional roles that sensory feedback may play. We and others have found that movement and the sensory feedback generated by that movement can change the state of the CPG and significantly alter the pattern of the behavior. In some cases the changes are on a time scale that outlasts a single cycle and have effects that are not predicted by any of the above examples. Two examples will be presented, one invertebrate and one vertebrate.

In the case of locust flight, removal of the hindwing tegulae results in an immediate change in the motor pattern. The wingbeat frequency decreases, and the interval between the activity of depressor and elevator muscles increases. Over a period of 2 weeks, the motor pattern can return to normal even without regeneration of the tegulae (Buschges, Ramirez, and Pearson, 1991). These changes indicate that the intact frequency and phase structure of the movement is normally a function of the CPG as well as the input from the wing sensors.

In the lamprey, slow changes evoked by sensory stimuli can continue over a time period that outlasts the sensory stimulation by varying numbers of cycles (Kiemel and Cohen, 2001). A small-amplitude bending movement of the isolated spinal cord is known to entrain the rhythm of fictive swimming, but more recently it has been found that the bending also leads to a speeding of the bursting that outlasts the bending for one or more cycles. The bending required for this effect can be very low amplitude and can last for as little as one cycle. Consequently, the longer-lasting, slowly decaying excitatory effect of the movement is apt to be seen during intact locomotion. In the latter, the sensory feedback would be a direct consequence of the muscle activation pattern generated by the CPG during its normal activity.

This type of slowly decaying excitation seen in lamprey could be interpreted as an example of positive feedback. That is, the CPG generates movement that in turn causes the CPG to go faster. However, the relationship between the frequency of the bending and the increase in the frequency is weak. If this is indeed positive feedback, then the gain is apparently less than 1, as the effects are limited (Prochazka, Gillard, and Bennett, 1997). The role for this type of excitation is unclear. A modeling study of the slowly decaying excitation (Verschure and Cohen, unpublished) offers some insights into possible roles the excitation might play. The model consists of the six-element neural network model for the lamprey segmental oscillator coupled to a reticulospinal element and receiving tonic drive to initiate bursting. This model was first proposed by Buchanan and Grillner (Grillner, Wallén, and Brodin, 1991). To it was added slowly decaying positive feedback from stretch receptors connected to the motor neuron output. The bursting remained controlled and stable over a wide range of parameter values. Rather than being destabilizing, this type of slowly decay-

ing excitation seemed to offer a new potential control mechanism. Interestingly, the system became autonomous, with no need for tonic drive and with the frequency of bursting highly sensitive to the gain and the duration of the stretch receptor feedback. This type of response may well depend on the balance of excitation and inhibition in a particular network. There was no attempt to test the effect of input with fewer or weaker inhibitory connections. However, it seems apparent that the well-studied lamprey CPG has no shortage of inhibitory neurons (Grillner et al., 1991). While such control mechanisms remain conjectural, the model does suggest that such types of excitation might perhaps be examined in new ways. One fairly definite role that this excitation could play is to change the gain on the system. That is, it seems likely that the excitation could increase the responsiveness of the CPG to other inputs.

In the lamprey (Guan, Kiemel, and Cohen, 2001) the changes induced by the movement have now additionally been implicated in altering the intersegmental coupling of the CPG. In the lamprey, the phase delays among the segments as well as the frequency of the movement are also altered by the movement. This was found in lamprey spinal cords induced to swim with the muscle present. There was no brain or tail, with the spinal cord kept as it is normally during the generation of fictive swimming; the spinal cord was exposed to the bath by removal of the dorsal musculature. The other muscle was left intact, and the preparation was pinned with a single pin at either the rostral or caudal end of the body to provide lateral stability (Figure 1). The burst pattern of the muscle was monitored with electromyography.

The cycle frequencies in such a preparation were consistently faster than the bursting of the respective isolated spinal cord. The phase delays were also consistently shorter. Thus, there is an alteration in the basic parameters of the movement in the presence of the muscle and movement. Using methods developed by Kiemel and Cohen (1998), the characteristics of the intersegmental coupling were measured in spinal cords with the long-range coupling reduced by lesions. Reducing the coupling was necessary since without the reduction, the values for the total coupling are all too near the asymptotic limit to be differentiated. In the presence of movement, the total coupling strength was found to be greatly enhanced in all spinal cords tested in this way. There were also changes in the ratio of ascending to descending short-range coupling. It has been shown in the isolated spinal cord of the lamprey

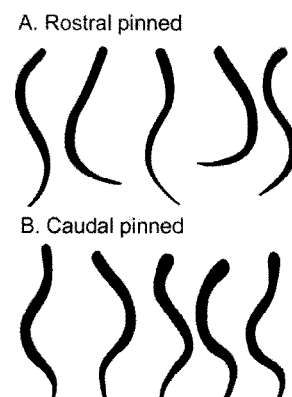


Figure 1. Reduced lamprey preparation, with body and muscle largely intact, swimming in response to bath applied D-glutamate (see Guan et al., 2001, for details of the preparation). Shown are the movements during one cycle of swimming when the body was pinned at the rostral end (A) and at the caudal end (B). Aside from the pinning, there was no difference in treatment between the two episodes of movement.

that after long-range coupling has been almost eliminated through lesions, the ascending short-range coupling is stronger than the descending short-range coupling (Guan et al., 2001). However, in the presence of movement and movement-related feedback, short-range coupling is found to be predominantly descending. Thus, movement and movement-related feedback can alter not only the movement parameters but also the underlying functional properties of the spinal circuitry. The movement could also be strikingly different, depending on which end of the body was allowed to be free (Cohen et al., unpublished) (Figure 1). It is important to note that this complexity is seen in the absence of the brain, and thus depends solely on the spinal cord–body interface.

Spinal Cord-Cortical Interactions

As discussed elsewhere in this *Handbook* (see CHAINS OF OSCILLATORS IN MOTOR AND SENSORY SYSTEMS), many studies of spinal cord treat it as a series of bidirectionally coupled oscillators. The coupling between the segmental oscillators in lamprey has been found to be very strong and is likely to be strong in other vertebrates as well. As such, the removal of one such oscillator or input to any oscillator, will alter the behavior of the other oscillators, regardless of their location within the informational stream. One can conclude from this that as sensory input travels up the spinal cord, it is shared with local circuits before it reaches the cortex. However, additional evidence is needed to truly make the case that motor and sensory systems are integrated across all levels, and not just across the spinal cord.

Interactions between cortex and spinal cord appear to be highly integrated. Indeed, there is evidence that there are neurons in the primary motor cortex of cats that are phasically active during locomotion of the animal. Thus, the strongest descending cortical outputs are most likely to be influenced by preexisting motor activity. This could mean that as the cortical neurons send motor commands to the limbs, the signals are apt to be properly timed to fit within the context of the ongoing locomotion. This is reasonable, as descending neurons must act in the proper context of the animal's movement or the commands will be inappropriate and maladaptive.

Neurons in the posterior parietal cortex of awake, unrestrained rats have also recently been found to exhibit activity linked to motor activity (McNaughton et al., 1994). Many of these cells responded strongly to specific types of locomotion. However, often these same cells would not respond to locomotor movements such as left turns or right turns. If the cells were responding to somatosensory or environmental information alone, one would expect them to respond in either the left turn or right turn case or to some specific position in space. In some animals a specific motion such as turning was compared with an attendant somatosensory state, such as passive bending of the animal. Less than 20% of the movement-related activity seemed to be explained by possible somatosensory or environmental inputs. Thus, the majority seemed to be predominantly motor in their responses.

Recently, it has been shown that the presence of cortical systems can speed up and enhance the responses over that of spinal animals (Hiebert et al., 1994). In some ways, this observation is counter-intuitive, as the spinal cord should be faster alone. There are a number of possible explanations for this phenomenon. One is that cortical neurons can alter the response properties of the spinal neurons, making them more excitable, or perhaps the cortex is more adept at responding to unexpected events. Whatever the explanation, while the spinal cord has the capacity to respond by itself, in intact animals the descending systems participate to make the responses more adaptive. This more rapid response in intact animals to certain types of stimuli is still consistent with integration taking place locally, as spinalized animals do respond to the stimuli, but

much more slowly. What the cortex could be providing is some expectation or change in gain such that it picks up that something is novel more quickly than the spinal cord alone.

In conjunction with the discussion in the previous section, the three cases discussed in this section show that sensorimotor interactions across all levels are bidirectional, and that CPG-generated activity has influences on cortical areas thought only to be influenced by or involved in the processing of sensory activity.

Conclusion

Sensory and motor systems are integrated across all levels of the nervous system, from the spinal cord to the cortex. When one considers CPGs and their interactions with sensory input, one sees a remarkable range of phenomena. The simple stimulus-response reflex is a minimal component. At the spinal cord level, responses to sensory stimuli are modulated by the CPG to produce adaptively altered responses, and sensory stimuli adjust the length and amplitude of the CPG cycle. However, sensory input is also filtered by the CPG itself, while sensory input can also alter the state of the spinal activity on a slower time scale and in more ways than is often described. Although sensory input performs the traditionally considered corrective and monitoring functions, in interaction with the CPG it can also change the overall frequency and relative timing of the muscles, as well as altering intersegmental coordination. At the cortical level, CPG input modulates the activity of pyramidal cells, most likely so that their descending signals will occur at appropriate times in the cycle.

The goal is for the organism to produce adaptive behavior. This can only occur if all levels of the nervous system are aware of and responsive to activity at other levels. The cortex is privy to information from all sensory modalities, several of which do not reach the spinal pattern generator. The cortex is responsible for integrating all of this rich sensory input and generating desired responses. However, cortical neurons, while integrating all of this sensory information, must be aware of the state and activity of the limbs and trunk musculature if cortical activity is to move the organism properly through the environment. The spinal pattern generator is the first line of proprioceptive sensory integration, but the spinal cord and the other levels of the nervous system mutually interact to meet the needs of the organism.

Road Maps: Motor Pattern Generators; Neuroethology and Evolution

Related Reading: Chains of Oscillators in Motor and Sensory Systems; Command Neurons and Command Systems; Gait Transitions; Locomotion, Invertebrate; Locomotion, Vertebrate; Spinal Cord of Lamprey: Generation of Locomotor Patterns

References

- Buschges, A., Ramirez, J.-M., and Pearson, K., 1991, Reorganization of sensory regulation of locust flight after partial deafferentation, *J. Neurobiol.*, 23:31–43.
- Cohen, A. H., 1992, The role of heterarchical control in the evolution of the central pattern generator for locomotion, *Brain Behav. Evol.*, 40:112–124. ♦
- Grillner, S., Wallén, P., and Brodin, L., 1991, Neuronal network generating locomotor behavior in lamprey: Circuitry, transmitters, membrane properties and simulation, *Annu. Rev. Neurosci.*, 14:169–199.
- Guan, L., Kiemel, T., and Cohen, A. H., 2001, Impact of movement and movement-related feedback on the lamprey central pattern generator for locomotion, *J. Exp. Biol.*, 204:2361–2370.
- Hiebert, G., Gorassini, M., Jiang, W., Prochazka, A., and Pearson, K., 1994, Corrective responses to loss of ground support during walking: II. Comparison of intact and chronic spinal cats, *J. Neurophysiol.*, 71:611–622.
- Jankowska, E., and Lundberg, A., 1981, Interneurons in the spinal cord, *TINS*, 4:230–233.

- Kiemel, T., and Cohen, A., 1998, Estimation of coupling strength in regenerated lamprey spinal cords based on a stochastic phase model, *J. Comput. Neurosci.*, 5:267–284.
- Kiemel, T., and Cohen, A., 2001, Bending the lamprey spinal cord causes a slowly-decaying increase in the frequency of fictive swimming, *Brain Res.*, 900:57–64.
- McNaughton, B., Mizumori, C., Barnes, C., Leonard, B., Marquis, M., and Green, E., 1994, Cortical representation of motion during unrestrained spatial navigation in the rat, *Cerebral Cortex*, 4:27–39.
- Nussbaum, M., El Manira, A., Gossard, J.-P., and Rossignol, S., 1996, Presynaptic mechanisms during rhythmic activity in vertebrates and invertebrates, in *Neurons, Networks, and Motor Behavior* (P. S. G. Stein, S. Grillner, A. I. Selverston, and D. G. Stuart, Eds.), Cambridge, MA: MIT Press, pp. 237–251. ♦
- Prochazka, A., Gillard, D., and Bennett, D. J., 1997, Implications of positive feedback in the control of movement, *J. Neurophysiol.*, 77:3237–3251.
- Rossignol, S., Lund, J. P., and Drew, T., 1988, The role of sensory inputs in regulating patterns of rhythmic movements in higher vertebrates: A comparison between locomotion, respiration and mastication, in *Neural Control of Rhythmic Movement in Vertebrates* (A. H. Cohen, S. Rossignol, and S. Grillner, Eds.), New York: Wiley, pp. 201–284. ♦
- Whelan, P. J., and Pearson, K. G., 1997, Comparison of the effects of stimulating extensor group I afferents on cycle period during walking in conscious and decerebrate cats, *Exp. Brain Res.*, 117:444–452.

Sensorimotor Learning

Daniel M. Wolpert and J. Randall Flanagan

Introduction

Skilled motor behavior is neither the result of rapid sensorimotor processes nor of fast or powerful effector mechanisms. Rather, the secret lies in the way tasks are organized and controlled by the nervous system. What sets humans apart from many robots that are far stronger and faster is our ability to select motor commands—both predictive and reactive—that are tailored to the task at hand and the physical properties of the environment. At the heart of our ability to deal with different tasks and contexts is sensorimotor learning.

Whereas some simple species show no motor learning, the need for motor learning arises in species in which the organism's environment, body, or task change. Specifically, when such changes are unpredictable, they cannot be prespecified in a control system, and therefore flexibility in the control process is required. Skills such as running on complex terrain or manipulating novel tools place a premium on motor learning. Similarly, as body size and proportions change with development, significant changes in the controller are required. Finally, learning is the only mechanism fast enough to allow us to master new tasks that are specified by social conventions, such as writing or dancing.

This article focuses on one perspective of motor learning, that is using internal models to learn transformations between sensory and motor variables. Both experimental and computational approaches to the study of internal model learning are reviewed.

Internal Models

The study of motor control is fundamentally concerned with the relationship between sensory signals and motor commands. Mapping between motor commands and sensory signals is bi-directional, and to specify the direction under consideration, a definition is adopted in which *forward* indicates the causal direction from motor commands into their sensory consequences, and *inverse* indicates the opposite direction; for example, transforming a desired sensory consequence into the motor commands that would achieve it.

The transformations from motor commands to their sensory consequences and vice versa are determined by the physics of the environment, musculoskeletal system, neural conduction and processing, and the sensory receptors. However, these physical transformations may also be represented internally within the central nervous system, and the phrase *internal model* is used to distinguish the actual transformation and its representation in the nervous system. Thus, the internal forward dynamic model is a model

within the brain that can predict how our arm will move—and the sensations that will arise—given a specific motor command.

Forward internal models capture the causal relationship between actions and their outcomes. Based on the efference copy produced in parallel with motor commands, the forward model predicts the sensory consequences of the ensuing movement. The central nervous system can use this prediction in several ways (Miall and Wolpert, 1996; ACTION MONITORING AND FORWARD CONTROL OF MOVEMENTS). One use of a forward model is to provide a fast internal loop that helps stabilize feedback control systems. Feedback control in biological systems is subject to potential difficulties with stability, because the sensory feedback through the periphery is delayed by a significant amount. Such delays can result in instability when trying to make rapid movements under feedback control. In predictive control, a forward model is used to provide internal feedback of the predicted outcome of an action. This internal feedback can be used before sensory feedback is available, thereby preventing instability.

Inverse models also play an important role in motor control. A particularly clear example of an inverse dynamic model arises in the vestibulo-ocular reflex (VOR). The VOR couples the movement of the eyes to the motion of the head, thereby allowing an organism to keep its gaze fixed in space (VESTIBULO-OCULAR REFLEX). This is achieved by causing the motion of the eyes to be equal and opposite to the motion of the head. In effect, the VOR control system must compute the motor command that is predicted to yield a particular eye velocity. This computation is an internal inverse model of the physical relationship between muscle contraction and eye motion.

We need to learn an inverse model in order to accurately estimate the motor commands required to achieve a desired sensory response. The feedforward control this allows is essential for most natural movements in which feedback is available too late to guide movement. There have been many control systems proposed in the literature that use direct control; that is, control architectures that do not explicitly use internal models. However, any good controller can be thought of as implicitly implementing an inverse model of the system being controlled. In other words, knowledge about the physical behavior of the system being controlled is employed by the controller.

Internal Model Learning

Skilled motor behavior requires both inverse and forward internal models. These models capture information about the properties of the sensorimotor system. These properties are not static but change

throughout life, both on a short time scale, as a result of interactions with the environment, and on a longer time scale, as a result of growth. Internal models must therefore be adaptable to changes in the properties of the sensorimotor system. A major aspect of motor learning can be viewed as the acquisition of forward and inverse internal models appropriate for different tasks and environments (Wolpert and Ghahramani, 2000; Wolpert, Ghahramani, and Flanagan, 2001).

Forward Model Learning

In supervised learning the target output can be provided by an external teacher, for example during imitation learning. However, the target output can also be specified internally based on sensory signals and higher-level goals. Such *self-supervised* learning is involved in the acquisition of a forward model that tries to predict the sensory consequence of an outgoing motor command (Figure 1, top). Here, the desired output of the model is readily available; it is the actual sensory consequence. The environment, therefore, readily provides an appropriate training signal to learn predictors of sensory feedback. The difference between the predicted and actual sensory feedback can be used as an error signal to update a predictive model. The neural mechanisms that lead to such predictive learning in the cerebellum-like structure of electric fishes have recently been partially elucidated (Bell, 2001; ELECTROLOCATION).

Inverse Model Learning: Supervised

An inverse model must learn to convert desired consequences into motor commands. One possible way to do this is with direct inverse modeling in which the inverse model observes the motor system during a “motor babbling” stage in which random motor commands are applied. The inverse model then observes the consequences of the motor commands and learns to associate these consequences (as its input) with the motor command that caused them (as its output). After learning, the inverse model’s input is the desired sensory consequence and its output should be the appropriate motor commands. This direct inverse modeling approach is well behaved for linear systems. For nonlinear systems, however, a difficulty arises that is related to the general “degrees-of-freedom problem” in motor control. In many motor systems different motor commands can lead to the same consequence. Therefore, during learning the inverse model will try to associate one outcome with many different motor commands, each of which will lead to this outcome. The inverse model will therefore learn to associate this outcome with the average of all motor commands that can cause it. However, in general, the average of all the motor commands, each of which individually lead to a common outcome, will not lead to the same outcome and therefore, in practice, direct inverse modeling fails.

Two learning mechanisms have been developed to deal with the problem associated with direct inverse model learning. These rely on transforming the outcome error back into a motor error that can be used to train the inverse model. For example, when we throw a dart, the error we receive is coded in visual coordinates. This sensory error must be converted into motor command errors suitable to update the inverse model. The two principal methods proposed in the motor control literature for solving this problem are “feedback error learning” and “distal supervised learning.”

Kawato, Furukawa, and Suzuki (1987) developed the feedback error learning approach to inverse model learning, which avoids the problem of direct inverse modeling (Figure 1, bottom). Feedback error learning makes use of a feedback controller to guide the learning of the inverse (feedforward) model. The total motor command acting on the motor system is the sum of the feedback control signal and the output from the inverse model. The feedback controller transforms the trajectory error, in sensory coordinates, into

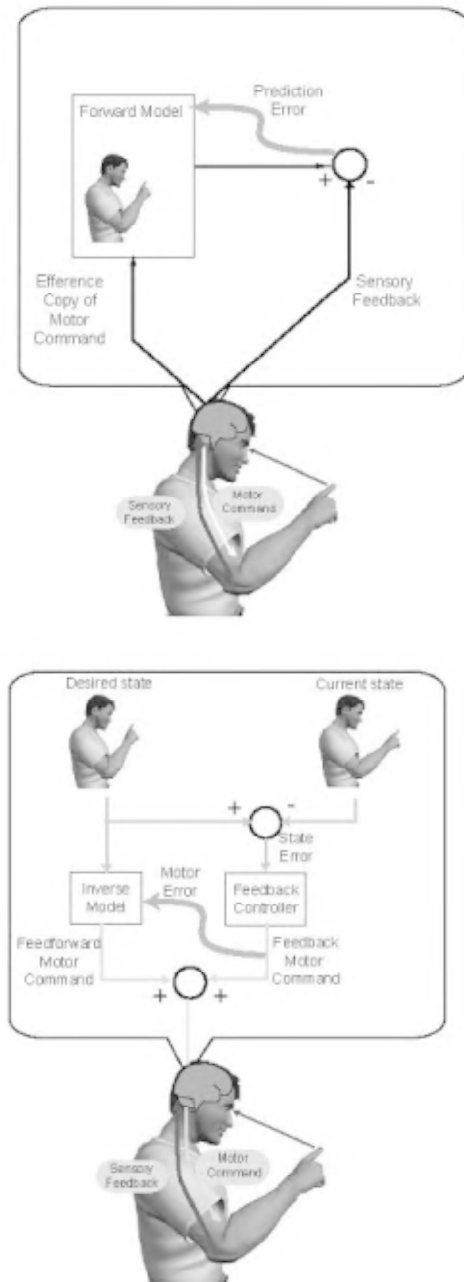


Figure 1. Schematics of forward model and inverse model learning. Top panel shows the forward model using a copy of the descending motor command, known as efference copy, to predict the sensory consequences of action. This prediction can be compared to the actual sensory feedback to generate a prediction error, which can be used (thick arrow) to update the forward model. The bottom panel shows a schematic of feedback-error learning. The aim is to learn an inverse model that can generate motor commands given a series of desired states. A hard-wired and low gain feedback controller is used to correct for errors between desired and estimated states. This generates a feedback motor command that is added to the feedforward motor command generated by the inverse model. If the feedback motor command goes to zero, then the state error will, in general, also be zero. Therefore the feedback motor command is a measure of the error of the inverse model and is used as the error signal to train it.

a feedback motor command, and this forms the error signal used to train the inverse model. This training signal therefore represents the sensory error converted into motor command coordinates. If the feedback motor command can be driven, through learning, to zero, then the inverse model is providing the appropriate motor command (because the outcome error must be zero) and the inverse model has been learned. Even for nonlinear systems, the feedback-error-learning method is generally successful.

In distal supervised learning, a forward model is used to convert outcome errors into errors in the motor command (Jordan and Rumelhart, 1996). The forward model must itself be learned from observations of the motor commands and their consequences on the motor system. The distal supervised learning approach is therefore composed of two interacting processes, one process in which the forward model is learned and another process in which the forward model is used in the learning of the inverse model. For the learning of the inverse model, the inverse model and the forward model are joined together and are treated as a single composite learning system. During this training process, the parameters in the forward model are held fixed and the errors in outcome propagated through the network to update the parameters of the inverse model only. In this way, the forward model is used to convert the outcome errors into motor errors.

Inverse Model Learning: Reinforcement

In *reinforcement learning* (REINFORCEMENT LEARNING IN MOTOR CONTROL), for each input to and output from the learning system, the environment provides feedback in the form of either reward or punishment. The overall performance measure that the system tries to maximize is the sum of total future rewards that may be weighted to favor immediate gain over longer-term gain. The concept of an overall punishment signal, or cost, from reinforcement learning has been very influential in motor control (OPTIMIZATION PRINCIPLES IN MOTOR CONTROL). Because of kinematic redundancy, almost any task can, in principle, be achieved in infinitely many ways. Consider, for example, the number of ways in which you could press an elevator button. Given all these possibilities, it is surprising that almost every study of the way the motor system solves a given task shows highly stereotyped movement patterns, both between repetitions of a task and between individuals on the same task. Such stereotypy is predicted when we consider tasks within the optimal control framework, in which a dynamic system (e.g., the arm) must be controlled so as to minimize a cost function (e.g., error reaching to a target). Mathematically, optimal control theory and reinforcement learning theory are equivalent. The difference is in emphasis: the former usually focuses on continuous state systems with known dynamics and known cost function, while the latter focuses on discrete state systems with unknown dynamics and cost functions that must be learned through experience. An important idea in motor learning has been to try to reverse-engineer the cost function the CNS uses, i.e., what is being optimized, from observed movement patterns and perturbation studies. For example, it has been proposed that there is noise in the motor command and that the amount of noise scales with the magnitude of the motor command. In the presence of such noise the same sequence of intended motor commands, if repeated many times, will lead to a probability distribution over movements. Aspects of this distribution, such as the spread of positions or velocities of the hand at the end of the movement, can be controlled by modifying the sequence of motor commands. In a simple aiming movement, the cost is the final error, as measured by the variance about the target. Assuming the presence of signal-dependent noise, a model that minimizes this cost accurately predicts the trajectories of both saccadic eye movements and arm movements

Different neural structures may be particularly adapted for different computational forms of learning (Doya, 2000). For example, the dopaminergic systems in the basal ganglia have been tied to signals that one would expect in reinforcement learning, such as expected reward, and dysfunctions of these systems are related to movement disorders, addiction, and other problems that could be related to reinforcement signals. Similarly, signals in the cerebellum have been linked to supervised errors. It has been shown that climbing fibres, which may act as a training signal to the cerebellum, can be used to train an inverse dynamic model of the eye (Kawato, 1999).

Novel Dynamic Learning

Recent work on motor learning has focused on the representation of the inverse dynamic model. When subjects make point-to-point movements in which the dynamics of their arm are altered, for example by using a robot or rotating room to generate a force field acting on the arm, they initially exhibit trajectories that deviate from their normal paths and velocity profiles (for reviews see Mussa-Ivaldi, 1999; Lackner and DiZio, 2000). However, over time, subjects adapt and move naturally in the presence of the force field. This can be interpreted as adaptation of the inverse model or the incorporation of an auxiliary control system with a new internal model to counteract the novel forces experienced during movement. Several theoretical questions have been addressed using this motor learning paradigm. The first explored the representation of the controller and in particular whether it was best represented in joint or Cartesian space. This was investigated by examining the generalization of motor learning at locations in the workspace remote from where subjects had adapted to the force field (Shadmehr and Mussa-Ivaldi, 1994). By assessing in which coordinate system the transfer occurred, evidence was provided for joint-based control. Another important advance was made in a study designed to answer whether the order in which states (positions and velocities) were visited was important for learning or whether having learned a force field for a set of states subjects would be able to make natural movements when visiting the states in a novel order. The findings showed that the order was unimportant and argued strongly against a rote learning of individual trajectories. In addition it has been shown that state-dependent fields are learned more efficiently than temporally changing fields, and that during learning both forward and inverse models are simultaneously adapted with the forward model leading.

Modular Learning

Recently, research has begun to shift away from examining learning of a single internal model to considering how we are able to learn a variety of tasks. Many situations that we encounter are derived from a combination of previously experienced situations, such as novel conjunctions of manipulated objects and environments. Internal models can be regarded conceptually as motor primitives, which are the building blocks used to construct intricate motor behaviors with an enormous vocabulary. By modulating the contribution to the final motor command of the outputs of a set of internal models, an enormous repertoire of behavior can be generated. One architecture that is capable of learning to act in multiple situations is the MOSAIC model (for review see Wolpert, Miall, and Kawato, 1998). In this architecture a set of forward models (predictors) are used as a set of hypothesis testers to assess which predictor best models the current task. This information is then used to weigh the outputs of a set of corresponding inverse models (controllers). This system can simultaneously learn multiple predictors and controllers as well as how to select the controller appropriate for a given task.

Recent studies have shown that after learning two different contexts, the CNS can appropriately mix the outputs both within the visuomotor domain and across the visuomotor and dynamic domains. Our understanding of the mechanisms of motor learning has gained from examining how learning one task can interfere with learning others. When trying to learn two opposing dynamic or visuomotor rearrangements interference occurs when they are presented in quick succession, but not when they are separated by several hours (Brashers-Krug, Shadmehr, and Bizzi, 1996). This suggests that motor learning undergoes a period of consolidation during which time the motor memory is susceptible to being disrupted. However, if the sensorimotor context is different, then opposite internal models can be simultaneously maintained in motor working memory and subsequently consolidated. For example, subjects can learn and consolidate two opposing force fields if the configuration of the wrist is different for the two fields. In contrast, arbitrary changes in context, such as color cues, are not sufficient for learning of two opposing fields. This suggests that the internal model captures a mapping between motor commands and sensory consequences that is determined by the force field but does not represent the force field per se. Recent studies have suggested that subjects are able to independently learn visuomotor and dynamic transformations presented in close temporal proximity. However, it appears that such independence is only observed when the two transformations depends on different kinematic variables (e.g., a position-dependent visuomotor rotation and a velocity-dependent force field).

Recent evidence indicates that the cerebellum plays a central role in the long-term storage of internal models (for a review see Wolpert, Miall, and Kawato, 1998). In addition it has been suggested that the spinal cord may store a small set of MOTOR PRIMITIVES (q.v.) or basis functions (Bizzi et al., 2000). The idea is to simplify control by combining a small number of primitives, for example patterns of muscle activations (synergies), in different proportions rather than individually controlling each muscle (Mussa-Ivaldi, 1999).

Road Map: Mammalian Motor Control

Background: Motor Control, Biological and Theoretical

Related Reading: Action Monitoring and Forward Control of Movements; Cerebellum and Motor Control; Motor Primitives; Robot Arm Control; Robot Learning

References

- Bell, C. C., 2001, Memory-based expectations in electrosensory systems, *Curr. Opin. Neurobiol.*, 11(4):481–487. ♦
- Bizzi, E., Tresch, M. C., Saltiel, P., and d'Avella, A., 2000, New perspectives on spinal motor systems, *Nat. Rev. Neurosci.*, 1(2):101–108. ♦
- Brashers-Krug, T., Shadmehr, R., and Bizzi, E., 1996, Consolidation in human motor memory, *Nature*, 18:252–255.
- Doya, K., 2000, Complementary roles of basal ganglia and cerebellum in learning and motor control, *Curr. Opin. Neurobiol.*, 10(6):732–739. ♦
- Jordan, M. I., and Rumelhart, D. E., 1996, Forward models: Supervised learning with a distal teacher, *Cognit. Sci.*, 16:307–354
- Kawato, M., 1999, Internal models for motor control and trajectory planning, *Curr. Opin. Neurobiol.*, 9(6):718–727. ♦
- Kawato, M., Furuwaka, K., and Suzuki, R., 1987, A hierarchical neural network model for the control and learning of voluntary movements, *Biol. Cybernetics*, 56:1–17
- Lackner, J. R., and DiZio, P. A., 2000, Aspects of body self-calibration, *Trends. Cogn. Sci.*, 4(7):279–288. ♦
- Miall, R. C., and Wolpert, D. M., 1996, Forward models for physiological motor control, *Neural Networks*, 9(8):1265–1279. ♦
- Mussa-Ivaldi, F. A., 1999, Modular features of motor control and learning, *Curr. Opin. Neurobiol.*, 9(6):713–717. ♦
- Shadmehr, R., and Mussa-Ivaldi, F. A., 1994, Adaptive representation of dynamics during learning of a motor task, *J. Neurosci.*, 14:3208–3224.
- Wolpert, D. M., Ghahramani, Z., and Flanagan, J. R., 2001, Perspectives and problems in motor learning, *Trends Cogn. Sci.*, 5(11):487–494. ♦
- Wolpert, D. M., and Ghahramani, Z., 2000, Computational principles of movement neuroscience, *Nature Neurosci.*, 3:1212–1217. ♦
- Wolpert, D. M., Miall, R. C., and Kawato, M., 1998, Internal models in the cerebellum, *Trends. Cogn. Sci.*, 2:338–347. ♦

Sensory Coding and Information Transmission

John Hertz and Stefano Panzeri

Introduction

The brain is an information-processing machine. Although this statement is commonplace, for a long time most neurobiologists understood the word “information” in it only in the informal, qualitative sense. However, in recent years, quantitative studies based on Shannon’s information theory (Shannon, 1948; Cover and Thomas, 1991) have markedly sharpened our understanding of neuronal coding.

Only a few years after Shannon’s invention of information theory, MacKay and McCulloch (1952) attempted to estimate the information transmission capacity of single spiking neurons. Assuming that it is limited only by the refractory time (1 ms) and the discriminability of successive spikes, one easily obtains an upper bound on the transmission rate of the order of 1,000 bits/s. A more restrictive bound is obtained by computing the actual entropy of spike trains. This is less than the above limit, because neurons spike less than half the time and different 1 ms intervals are not independent. Still, numbers on the order of 500 bits/s are found.

Now, if neurons are intrinsically very noisy devices, reliable transmission will require a high degree of redundancy, and the

actual rate at which they convey information will be correspondingly lower. It has been commonplace to suppose that this is the case, but neural firing may appear noisy to us only because we do not understand it. To achieve even the beginning of an understanding of how the brain works, it is necessary to measure the rate at which neurons actually carry information.

Of course, we do not know how much of the measured information is actually used by downstream neurons. However, measurements of information transmission can usefully identify features of the neural code and thereby help us understand how the brain computes.

An alternative approach to the problem of neural coding is to try to do decoding: optimally estimating the stimulus, given the neuronal response (see, e.g., POPULATION CODES and the review by Borst and Theunissen, 1999).

In the 1970s, Eckhorn and Pöpel (1974, 1975) laid the foundation for much subsequent work on measuring transmitted information by calculating the rate at which neurons in the cat lateral geniculate nucleus carried information about a random train of visual flashes, independent of a priori assumptions about how it was

coded. They found rates from about 10 bits/s to as high as 60 bits/s, depending on how fast the stimulus was flashed.

In the 1980s and 1990s, several groups extended this approach. Among them were William Bialek, Rob de Ruyter, and their collaborators (de Ruyter van Steveninck and Bialek, 1988; Bialek and Rieke, 1992), who studied information transmission by single neurons involved in vision in flies, hearing in frogs, and mechanoreception in crickets; and Barry Richmond and co-workers (Optican and Richmond, 1987; Heller et al., 1995), who made similar analyses for spatial pattern vision in monkeys. By now, the quantitative characterization of neurons in terms of information transmission rates has become a standard tool.

Here we review two recent approaches to measuring transmitted information. The first, employed by Bialek and his collaborators, is based on direct estimation of the spike train entropies in terms of which transmitted information is defined. The second, developed by Panzeri and Schultz, is based on an expansion to second order in the length of the spike trains.

Entropy and Information

Whatever system we are interested in, the formal characterization of the problem is the same. The animal is presented a stimulus s from some set S , and the response of a neuron (or several neurons) is measured. For spiking neurons, the response can be represented completely generally in the following way. Time is divided into intervals (typically 1 ms) small enough that there is never more than one spike per interval. The response can then be described by a binary vector with one component per interval, equal to a 1 or 0 according to whether or not the neuron fired a spike in that interval. We denote this vector simply by r and the whole set of responses by R .

To get a grasp of what “transmitted information” means, consider an experiment in which many spike trains are recorded for many stimuli. There is generally intrinsic variability in the system or the measurement process, so we formulate our description of the problem in terms of distributions of stimuli and responses. The fact that the responses depend to some extent on the stimuli means that the stimulus-conditional response probabilities $P(r|s)$ are not in general equal to the unconditional or stimulus-averaged probabilities $P(r) = \sum_s P(r|s)P(s)$. We expect that the variability of the responses evoked by a given stimulus will generally be less than that of the entire response set. The transmitted information is a simple measure of this variability difference.

A general way of characterizing variability is as an entropy. Thus, the total or unconditional response entropy $H(R) = -\sum_r P(r) \log_2 P(r)$ describes the variability of the entire response ensemble. Analogously, the response entropy conditional on a given stimulus s is $H(R|s) = -\sum_r P(r|s) \log_2 P(r|s)$. Its average over s , denoted $H(R|S)$ —the average response variability for a fixed stimulus—is termed the *noise entropy*. The transmitted information $I(R; S)$, also called the mutual information between stimuli and responses, is simply the difference between the total and noise entropies: $I(R; S) = H(R) - H(R|S)$. Although the corresponding difference for a single s , $H(R) - H(R|s)$, is not necessarily positive (a single stimulus might evoke highly variable responses), $I(R; S)$ can be proved to be non-negative. It is zero only when the stimulus-response relationship is completely random, and it takes its maximum possible value $H(R)$ only when the response is deterministic (for a given stimulus, every response is identical).

Therefore, the calculation of the transmitted information requires simply the accurate estimation of the conditional probabilities $P(r|s)$ and, from them, the total and averaged conditional entropies, as described above. If one has sufficient data, one can just estimate response probabilities as the fraction of all responses with a particular value of r . This can be difficult because the full response space

has such a high dimensionality: 2^T bins for spike trains T time units long.

The difficulty can be circumvented by employing specific models for the encoding, i.e., for the relevant conditional probabilities. Almost all the work mentioned above was done this way. This reduces the dimensionality of the response space, but at a cost: particular features of the results might be artifacts of the model used. Recent investigations have concentrated instead on model-independent methods, and here we review two sets of studies along these lines.

Direct Calculations

In the approach taken by Bialek’s group (Strong et al., 1998), one tries to estimate the probabilities $P(\{t_i\})$ and $P(\{t_i\}|s)$ directly from histograms of the data. (We use the notation $\{t_i\}$ for the response r to emphasize that we mean the entire set of spike times, not just a spike count or some other low-dimensional measure.) These estimated probabilities are then used to calculate the total and noise entropies $H(R)$ and $H(R|S)$ and thereby the transmitted information, as described above.

In the experimental paradigm used, the fly views the moving pattern for very long periods, and we are interested in the information transmission rate. To estimate this, one considers time windows of length T and tries to extrapolate to the limit of very long windows. For small T , it is easy to make reliable entropy estimates directly (consider a 1 ms window in which only two responses are possible, spike or no spike), but these estimates ignore possible correlations between spikes at different times. For very large T , on the other hand, the exponentially growing dimensionality of the space of responses prevents one from estimating their probabilities reliably from histograms based on the data.

The solution is to start with small T , then go to larger and larger windows, trying to identify the large- T trend while the estimates are still reliable. The estimated entropy rates (entropies divided by T) at lower values of T fit a nice straight line as a function of $1/T$, so it is simple to extrapolate to $1/T = 0$ to estimate the asymptotic rates.

The difference between the resulting total and noise entropy rates (Figure 1) gave an estimate of 78 ± 5 bits/s (1.8 bits/spike) for a time-discretization window $\Delta t = 3$ ms. Reassuringly, this number is consistent with those found by other methods (de Ruyter van Steveninck and Bialek, 1988; Bialek and Rieke, 1992).

The noise entropy was just about half of the total entropy; that is, half of the total observed variability in the neuronal firing is signal and half is noise.

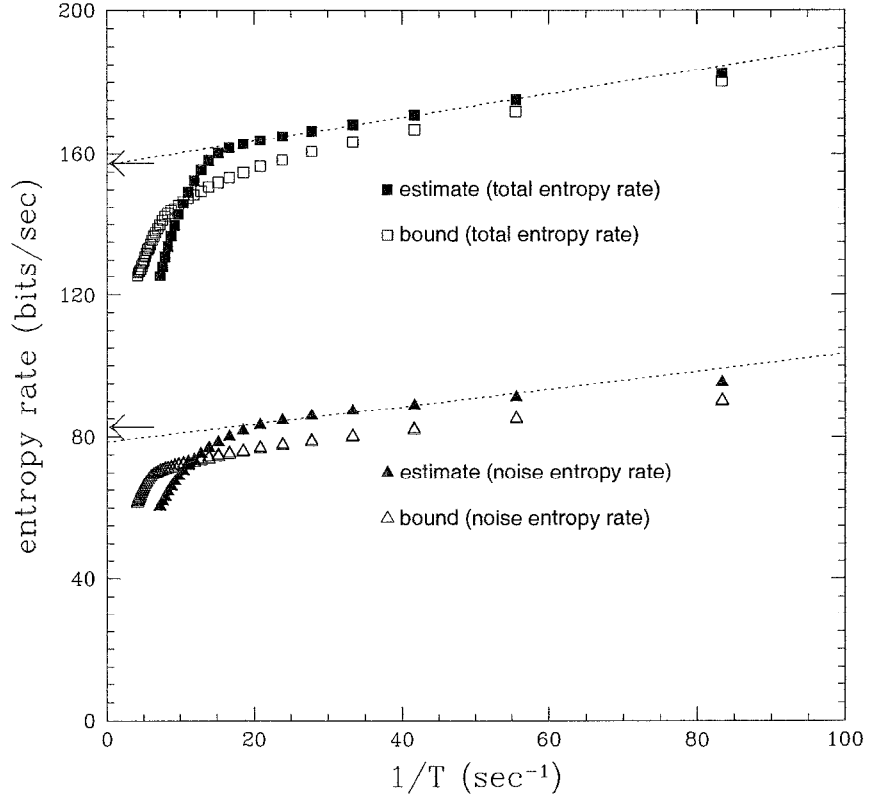
Varying Δt , one can explore how much information is encoded in spike timing at different degrees of precision. The transmission rate was found to vary roughly logarithmically with Δt , with a maximum value of 90 bits/s found for a resolution of 0.7 ms.

T-Expansion

Panzeri and Schultz (2001) and Schultz and Panzeri (2001) observed that it was possible to make a formal expansion of both the total and noise entropies in powers of T , the length of the time interval over which the response is measured. The computational advantage of using this expansion is that, working, say, to second order in T , one only need estimate the quantities $P(t^a|s)$ (the probability, given stimulus s , of a spike from unit a at time t^a , conditional on stimulus s) and $P(t_1^a, t_2^b|S)$ (the joint probability of a spike from unit a at time t_1^a and one from unit b at time t_2^b , also conditional on s), not the full conditional response probabilities $P(\{t_i^a\}|s)$. Thus, fewer data are required to make a robust estimation.

Beyond second order in T , the expansion becomes very cumbersome, but the second-order expansion proves quite accurate for

Figure 1. Procedure for estimating total and noise entropy rates. Entropies are estimated for finite-time window width T , plotted functions of $1/T$, and extrapolated to $T = \infty$. Here the bin width $\Delta t = 3$ ms. The upper sets of points are for the total entropy and the lower sets are for the noise entropy. Also shown (open symbols) are the Ma bounds. The dashed lines show the extrapolations $T \rightarrow \infty$. (From Strong et al., 1998. Reprinted with permission.)



experimentally relevant response times (20–100 ms). Furthermore, the first- and second-order terms can be written in a way that isolates the potential effects of spike timing and of correlation in both signal and noise, permitting direct insight into the nature of the neuronal code.

We now examine these terms. For generality, we consider the case of multiple neurons, labeled by an index a running from 1 to C . The response r is the full set of spike times t_i^a of all the neurons. The expansion takes the form

$$I(\{t_i^a\}; S) = TI_t(\{t_i^a\}; S) + \frac{T^2}{2} I_n(\{t_i^a\}; S) + O(T^3) \quad (1)$$

The first-order term has a very simple form:

$$TI_t(\{t_i^a\}; S) = \sum_{a=1}^C \int dt^a \left\langle \bar{r}_a(t^a|s) \log_2 \frac{\bar{r}_a(t^a|s)}{\bar{r}_a(t^a)} \right\rangle_s \quad (2)$$

Here $\bar{r}_a(t^a|s)$ is the average firing rate at time t^a , conditional on stimulus s , and $\bar{r}_a(t^a) = \langle \bar{r}_a(t^a|s) \rangle_s$ is the rate averaged over all stimuli. Since Equation 2 is a linear sum over each time t and cell a , it gives the information gained about the stimulus (i.e., the entropy reduction) from knowing the spike train, if each response time bin and cell were to convey independent information. Deviations from independent information transmission (i.e., synergy or redundancy effects) are expressed by the second-order term, considered next. Correlations between cells or between spikes for a given cell do not enter at first order in T .

At order T^2 , correlations between pairs of spikes affect the result. We can describe these correlations by some auxiliary quantities $\gamma_{ab}(t_1^a, t_2^b|s)$ and $v_{ab}(t_1^a, t_2^b)$, defined as follows. Consider, for stimulus s , the joint probability rate of a spike at time t_1^a by neuron a and one by neuron b at t_2^b , which we write in the form

$$P(t_1^a, t_2^b; s) = \bar{r}_a(t_1^a|s) \bar{r}_b(t_2^b|s) [1 + \gamma(t_1^a, t_2^b|s)] \quad (3)$$

If different spikes were independent, the joint probability rate would just be the product of the separate single-spike rates, i.e., $\gamma(t_1^a, t_2^b|s) = 0$. The quantity $v_{ab}(t_1^a, t_2^b)$ measures correlations in the rates across stimuli in an analogous way:

$$v_{ab}(t_1^a, t_2^b) = \frac{\langle \bar{r}_a(t_1^a) \bar{r}_b(t_2^b) \rangle_s}{\bar{r}_a(t_1^a) \bar{r}_b(t_2^b)} - 1 \quad (4)$$

In terms of these quantities, the second-order transmitted information can be written

$$\begin{aligned} \frac{T^2}{2} I_n(\{t_i^a\}; S) = & \frac{1}{2 \ln 2} \sum_{ab} \int dt_1^a dt_2^b \bar{r}_a(t_1^a|s) \bar{r}_b(t_2^b|s) \left\{ v_{ab}(t_1^a, t_2^b) \right. \\ & + [1 + v_{ab}(t_1^a, t_2^b)] \ln \frac{1}{1 + v_{ab}(t_1^a, t_2^b)} \Big\} \\ & + \frac{1}{2} \sum_{ab} \int dt_1^a dt_2^b \langle \bar{r}_a(t_1^a|s) \bar{r}_b(t_2^b|s) \gamma(t_1^a, t_2^b|s) \rangle_s \\ & \times \log_2 \frac{1}{1 + v_{ab}(t_1^a, t_2^b)} + \frac{1}{2} \sum_{ab} \\ & \times \int dt_1^a dt_2^b \langle \bar{r}_a(t_1^a|s) \bar{r}_b(t_2^b|s) [1 + \gamma(t_1^a, t_2^b|s)] \\ & \times \log_2 \left[\frac{\langle \bar{r}_a(t_1^a|s') \bar{r}_b(t_2^b|s') \rangle_s [1 + \gamma(t_1^a, t_2^b|s)]}{\langle \bar{r}_a(t_1^a|s') \bar{r}_b(t_2^b|s') \rangle_s [1 + \gamma(t_1^a, t_2^b|s')] } \right] \Bigg\rangle_s \end{aligned} \quad (5)$$

We consider the three terms on the right-hand side of Equation 5 separately. The first one is the only one that would be there if there were no correlations between spikes. It is always negative. Since the first-order term (Equation 2) grows linearly with time but the total information is bounded, the rate of information accumu-

lation has to slow down with time, and this term is the first place where we see that effect. The second term reflects the effect of spike correlations, through $\gamma(t_1^a, t_2^b|s)$, but these correlations are averaged over stimuli (weighted in proportion to the rates $\bar{r}_a(t_1^a|s)$ and $\bar{r}_b(t_2^b|s)$ they evoke). Thus, this term describes the effects of *stimulus-independent* firing correlations. The final term, which cannot be negative, is the part of the information due to stimulus-dependent correlations.

This treatment was for the complete neuronal response, i.e., r is the set of firing times t_i^r for all the neurons. Often, one is interested in how much information is carried by the spike count alone, independent of the firing times. This information can be expanded in the same way, and the result is analogous to Equations 2 and 5. The differences are, first, there are no time integrals; second, the conditional and unconditional spike counts replace the full response rates $\bar{r}_a(t^a|s)$ and $\bar{r}_a(t^a)$; and third, there is no longer any time dependence in the correlation coefficients $\gamma_{ab}(s)$ and v_{ab} .

To see what one can learn from this kind of analysis, we review its application to data from rat barrel cortex (Panzeri et al., 2001). In the experiment, single whiskers were stimulated and the responses of 106 neurons were collected and analyzed.

The first significant finding was that a substantial fraction of the total information carried by these neurons about which whisker was being stimulated was encoded temporally. That is, there was considerable information carried by the full response (the complete sets of spike times) that was not carried by the spike counts alone. The size of this effect varied a lot from neuron to neuron, but, on average, including spike timing added about 50% to the information contained in the spike count.

The nature of the temporal code employed by these neurons can be explored further by examining the contributions from the separate terms in Equations 2 and 5. Figure 2 shows the principal findings. Most of the information (83%) could be accounted for without taking any spike correlations into account. More specifically, different stimuli evoke responses with different PSTHs $\bar{r}_a(t^a|s)$, and these differences account for most of the discriminability of the stimuli. For example, $\bar{r}_a(t^a|s_1)$ might rise a lot faster than $\bar{r}_a(t^a|s_2)$. In this case an early spike would be a good indication that it was evoked by s_1 rather than s_2 . Indeed, in the present case, the first-

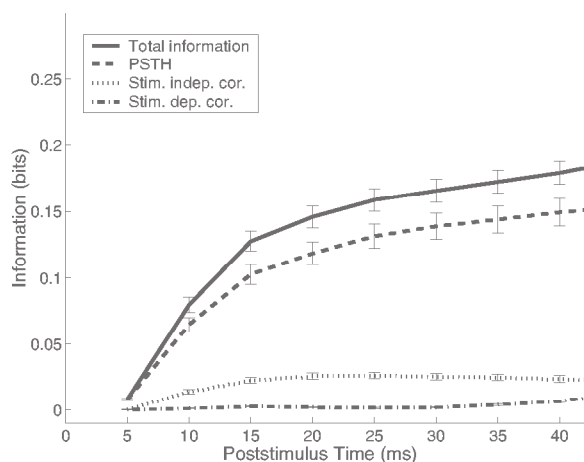


Figure 2. Autocorrelations carry little information about stimulus position in rat somatosensory cortex: Contributions to the transmitted information from PSTHs (dashed line; Equation 2 and the first term on the right-hand side of Equation 5), stimulus-independent correlations (dotted line; second term on the right-hand side of Equation 5), and stimulus-dependent correlations (dashed-dotted line; third term on the right-hand side of Equation 5). Bars denote SEM. (Redrawn from Panzeri et al., 1998.)

spike time alone conveyed (on average) as much information as could be gained from all spikes, ignoring correlations. Almost all of the remaining 17% of the information (13.5%) was coded in stimulus-independent correlations (i.e., the second term on the right-hand side of Equation 5), with only 3.5% in stimulus-dependent correlations (the final term).

Discussion

These examples illustrate how, given sufficient data, neural information transmission can be measured systematically and accurately, and how one can begin to sort out details of the neural code employed by the brain in specific cases. The computational technology is by now rather well developed, and we can expect that in the future it will be applied to more and more data, in different animals, in different brain regions, and in the context of different brain functions. One can then hope that, integrating this knowledge with the insight gained from exploring the implications of particular coding strategies (see SPARSE CODING IN THE PRIMATE CORTEX and FEATURE ANALYSIS), a coherent general picture of the encoding and computational strategies employed in neural processing will begin to emerge. An extensive and readable treatment of these and related issues can be found in the book by Rieke et al. (1997).

Road Map: Neural Coding

Related Reading: Adaptive Spike Coding; Optimal Sensory Encoding; Rate Coding and Signal Processing

References

- Bialek, W., and Rieke, F., 1992, Reliability and information transmission in spiking neurons, *Trends Neurosci.*, 15:428–434.
- Borst, A., and Theunissen, F. E., 1999, Information theory and neural coding, *Nature Neurosci.*, 2:947–957. ♦
- Cover, T. M., and Thomas, J. A., 1991, *Elements of Information Theory*, New York: Wiley. ♦
- de Ruyter van Steveninck, R. R., and Bialek, W., 1988, Real-time performance of a movement-sensitive neuron in the blowfly visual system: Coding and information transfer in short spike sequences, *Proc. R. Soc. Lond. B*, 212:1–34.
- Eckhorn, R., and Pöpel, B., 1974, Rigorous and extended application of information theory to the afferent visual system of the cat: I. Basic concepts, *Kybernetik*, 16:191–200.
- Eckhorn, R., and Pöpel, B., 1975, Rigorous and extended application of information theory to the afferent visual system of the cat: II. Experimental results, *Biol. Cybern.*, 17:7–17.
- Heller, J., Hertz, J. A., Kjær, T. W., and Richmond, B. J., 1995, Information flow and temporal coding in primate pattern vision, *J. Computat. Neurosci.*, 2:175–193.
- MacKay, D. M., and McCulloch, W. S., 1952, The limiting information capacity of a neuronal link, *Bull. Math. Biophys.*, 14:127–135.
- Optican, L. M., and Richmond, B. J., 1987, Temporal encoding of two-dimensional patterns by single units in primate inferior temporal cortex: III. Information-theoretic analysis, *J. Neurophysiol.*, 57:162–178.
- Panzeri, S., Petersen, R., Schultz, S. R., Lebedev, M., and Diamond, M., 2001, The role of spike timing in the coding of stimulus location in rat somatosensory cortex, *Neuron*, 29:769–777.
- Panzeri, S., and Schultz, S. R., 2001, A unified approach to the study of temporal, correlational and rate coding, *Neural Comp.*, 13:1311–1349.
- Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W., 1997, *Spikes: Exploring the Neural Code*, Cambridge, MA: MIT Press. ♦
- Schultz, S. R., and Panzeri, S., 2001, Temporal correlation and neural spike train entropy, *Phys. Rev. Lett.*, 86:5823–5826.
- Shannon, C. E., 1948, A mathematical theory of communication, *Bell Syst. Tech. J.*, 27:379–423, 623–653. Available: <http://cm.bell-labs.com/cm/ms/what/shannonday/paper.htm>.
- Strong, S. P., Koberle, R., de Ruyter van Steveninck, R. R., and Bialek, W., 1998, Entropy and information in neural spike trains, *Phys. Rev. Lett.*, 80:197–200.

Sequence Learning

Peter Ford Dominey

Introduction

The forward linear progression of time imposes a fundamental sequential structure on all behavior, and thus the capacity to store and manipulate sequential information is of central importance for adaptive systems. Sequential structure is not unidimensional, however, and the potentially dissociable aspects of sequence processing are likely to be associated with distinct neural systems. From this perspective, the current article will characterize behavioral sequences in terms of their serial, temporal, and abstract structure, and aspects of the associated neural processing systems.

Serial structure or order is defined by the relation between an element or set of elements, and its successor. This dimension can be characterized in terms of length and complexity. *Length* is the number of elements in the sequence. *Complexity* refers to the maximum number of elements that must be remembered in order to know the correct successor. Temporal structure is defined in terms of the durations of elements (and the possible pauses that separate them), and intuitively corresponds to the familiar notion of rhythm. Thus, two sequences may have identical serial structure and different temporal structure, or the opposite. Abstract structure is defined in terms of generative rules that describe relations between repeating elements within a sequence. Thus, the two sequences A-B-C-B-A-C and D-E-F-E-D-F have different serial structure, but are both generated from the same abstract structure 123-213, and are thus said to be isomorphic. While perhaps not exhaustive, these three dimensions at least partially span the space of possible behavioral sequence structure. This article focuses on how these different dimensions of sequence structure can be encoded in neural systems based on behavioral studies in different patient and control groups, and related simulation studies. Related issues in temporal sequence learning are found in the articles TEMPORAL SEQUENCES: LEARNING AND GLOBAL ANALYSIS; TEMPORAL PATTERN PROCESSING; and RECURRENT NETWORKS: LEARNING ALGORITHMS (q.v.).

Learning Serial and Temporal Structure

Aspects of Recurrent Networks

A fundamental property of a sequence processing system is that the state of the system should contain information about the current sequence element, but should also be influenced by the succession of previous elements. This is the case for recurrent networks in which context units receive inputs from units encoding internal state and feed back into these state units. These recurrent connections between the state and context units allow information from previous time steps to influence the current network activity, thus providing a representation of sequence context. Learning that requires modification of recurrent connections, however, introduces significant technical challenges. Specifically, after an input is presented, multiple network cycles can occur before an output is generated and the error is evaluated and corrected. A given connection weight in the recurrent network has contributed to the error, but in a different way on each successive cycle of information passing through this weight. The technical problem thus arises in unraveling this weight's contribution to the error over these successive time steps in order to implement the error-reducing learning. A number of methods have been developed to deal with the complexity of applying learning algorithms to recurrent connections (Pearlmutter, 1995, and RECURRENT NETWORKS: LEARNING ALGORITHMS). In general, many of these methods of resolving the

problem of learning in recurrent networks over multiple time steps are biologically implausible, because they are not consistent with forward running time, and/or because they have excessive computational and memory storage requirements.

The method used by Elman (1990) in the simple recurrent network (SRN) is to cut off the temporal history at one or two time steps, yielding a simplified learning method but still quite efficient sequence learning capability. The SRN has been demonstrated to be sensitive to serial structure in a number of different domains including representing regularities in the serial structure of language (Elman, 1990), modeling finite-state automata (Cleeremans, Servan-Schreiber, and McClelland, 1989), speech segmentation (Christiansen, Allen, and Seidenberg, 1998), and simulating human performance in sensorimotor sequence learning (Cleeremans, 1993) in serial reaction time (SRT) tasks.

Sequence Learning in the Serial Reaction Time Task (SRT)

The serial reaction time task (SRT) developed by Nissen and Bullemer (1987) has been quite extensively used in the study of human sequence learning, and is thus an interesting object for simulation studies. In the SRT task, reaction times (RTs) for visual stimuli that are presented in a repeating sequence are reduced with respect to RTs for stimuli in randomly presented series (see Figure 1B). This RT difference is the measure of sequence learning. One of the principal manipulations in the SRT paradigm has been the performance of a concurrent or "dual" task that typically impairs the sensorimotor learning (Nissen and Bullemer, 1987). Curran and Keele (1993) proposed the existence of dissociable attentional and nonattentional forms of sequence learning, and that the dual task disrupts attentional sequence learning, while leaving the less efficient nonattentional learning intact. Cleeremans (1993) suggested alternatively that the dual task impairs the processing of successive sequence elements, and he thus introduced random noise to the activity of a subset of units in a simple recurrent network (SRN) during SRT learning, resulting in perturbed performance quite similar to that observed in human subjects in dual task conditions (Cleeremans, 1993). This, however, leaves open the question of how the dual task conditions introduced processing noise to produce such a perturbation in sequence learning.

Stadler (1995) suggested that the dual task condition disrupts sequence learning by preventing consistent temporal organization of the sequence due to the delays introduced during the response-stimulus interval (RSI) by the dual task processing. He thus demonstrated experimentally that the introduction of random delays during the RSI perturbs sequence learning in a manner quite similar to that of the dual task condition, while introducing a purely attentional load with no temporal disorganization does not. This suggested that dual task results could be explained without resorting to dissociable learning mechanisms. Testing this temporal disorganization hypothesis directly in a simulation study would require the presentation of successive elements with realistic RSI delays.

A Temporal Recurrent Network

Temporal structure in sequences can be encoded by systems that employ mechanisms including distributions of delay lines (TEMPORAL SEQUENCES: LEARNING AND GLOBAL ANALYSIS) between units, time-dependant short-term memory elements (TEMPORAL PATTERN PROCESSING), and the temporal dynamics of recurrent networks, though this is not the case for the standard implemen-

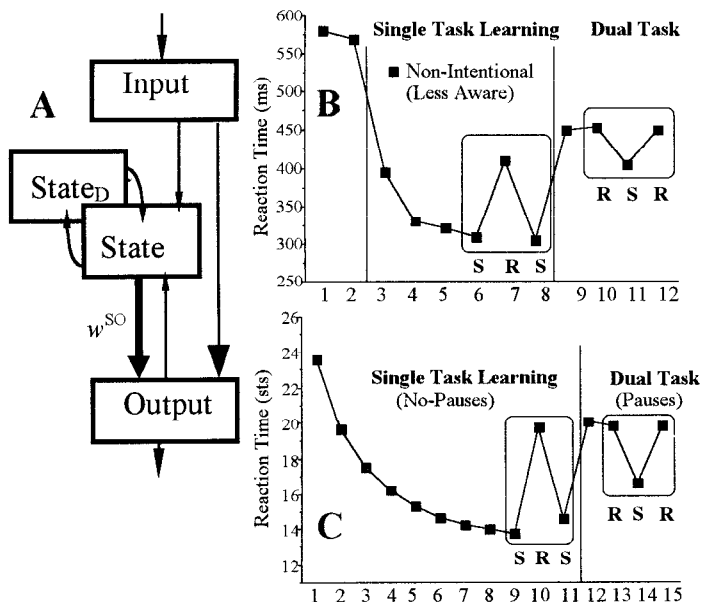


Figure 1. A, TRN architecture. Each structure is a 5×5 matrix of leaky integrator neurons with sigmoid firing thresholds. Reinforcement learning in w^{SO} synapses binds activation patterns in State to output activations in Output. B, SRT learning. Mean RTs for successive blocks of 120 sequence elements. RTs for sequentially presented stimuli in blocks 3–6 are reduced, and then increase in random block 7, revealing learning. Learning expression reduced in dual task as revealed by reduced difference in random (block 11) vs. sequence (blocks 10 and 12) RTs. (Data from Experiment 1 of Curran and Keele, 1993.) C, Simulation of SRT learning by the TRN. RTs are reduced as w^{SO} synapses associate sequence-specific State activity with corresponding predictable sequence elements, yielding stronger activation and faster rise to threshold for sequence (S) vs. random (R). Dual task conditions achieved by introduction of random pauses as suggested by Stadler (1995), simulating human data in B. (Data from Dominey, 1998.)

tations of the SRN where the next element is expected on the next network cycle as discussed previously. Dominey, Arbib, and Joseph (reviewed in Dominey et al., 1998) introduced an alternative sequence learning model that takes full advantage of the temporal dynamics of a recurrent network, while eliminating the complexity of learning in recurrent connections (see Figure 1A). Recurrent connections in the State network of leaky integrator neurons—considered to represent cortico-cortical connections in prefrontal cortex (PFC)—are preset with random values that vary between -0.55 and $+0.45$, and do not change during learning. Associative learning takes place in modifiable connections between the recurrent state layer and the output layer, corresponding to dopamine-related plasticity in corticostriatal synapses. Neurons in the simulated PFC (State in Figure 1) demonstrated receptive fields sensitive both to the spatial location of sequence elements, and their serial order, as observed in electrophysiological recordings in PFC of the non-human primate (Dominey, Arbib, Joseph in Dominey et al., 1998).

With respect to the SRT task (Figure 1B), as a sequence is successively repeated and learned, reaction times for predictable elements in the sequence become progressively reduced due to increased State-Output connection strengths that yield more rapid activation of the leaky integrator neurons in Output for elements in the repeating sequence. Thus the model naturally displays reduced RTs for predictable sequence elements (Figure 1C).

Processing of Serial and Temporal Structure

Given its fixed recurrent connections, the TRN exploits the temporal coding capabilities of a recurrent system while avoiding the complexity of recurrent learning, and is thus well adapted to examining the effects of serial and temporal structure in SRT learning. During SRT learning, we introduced random delays during the RSI interval between the model response and the next sequence elements, to simulate the effects of dual task learning as in Stadler's (1995) experiment. This temporally distorted input produced a degraded representation of the sequence in the recurrent network, resulting in weaker activation of the learned State-Output associations, and a corresponding increase in reaction times. This argues for the temporal structure disruption hypothesis of Stadler (1995)

and explains human sensitivity to perturbations in temporal structure of such sequences (Dominey, 1998). In contrast, if temporal structure is introduced in a systematic (rather than random) way that is coherent with the serial structure then these regularities can be represented in a systematic way in the recurrent network and can be learned (Dominey in Dominey and Ramus, 2000).

Indeed, serial and temporal structure are often correlated in behavioral sequences, a condition that likely plays an important role in behavioral sequence learning. In particular, in the earliest phases of language acquisition it is likely that sensitivity to correlations in serial and temporal structure provides the foundation for the construction of more complex linguistic representations. Christiansen et al. (1998) thus demonstrated how the SRN can exploit multiple cues, each of which alone is insufficient, in learning to segment words in a continuous stream. Because of the temporal processing constraints of the SRN, temporal structure is coded symbolically in these simulations, rather than in real time.

In this context the TRN was confronted with human performance in which newborns discriminate between languages in different rhythm classes based on their temporal structure (Dominey and Ramus, 2000). Simulating these results with the TRN, sentences from five different languages were recoded as consonant-verb (C-V) sequences, where the only defining information was the rhythmic structure. Like children, the TRN was able to discriminate between languages from different rhythm classes (e.g., English and Spanish) but not between languages from the same rhythm class (e.g., Dutch and English).

Learning Abstract Structure

Although it has been demonstrated that the serial and temporal structure of sequences can be processed by a common architecture, it is not clear that this generalization extends to abstract structure. In an elegant presentation of abstract structure processing in the infant, Marcus et al. (1999) demonstrated that after only 2 minutes' total exposure to 16 sound sequences such as "le-di-di, ji-we-we..." 7-month old infants can extract the common abstract structure (in this case ABB), and can transfer this knowledge in order to recognize new sequences as isomorphic (matching) or not with the learned pattern.

Marcus et al. concluded that infants can learn rules such as ABB, in which A and B represent variables that can be instantiated with new values during generalization and transfer to new sequences, and that such rule representations cannot be realized by statistical methods or standard sequencing models including the SRN. In a series of comments in *Science* and *Trends in Cognitive Science*, opponents argued that if training and transfer domains shared overlapping representations, then statistical learning methods and the SRN would work. Marcus et al. countered that transfer can still be observed even when there is no overlap in the representation, and that systems that require this overlap will fail in these cases.

Addressing this transfer issue, Dienes, Altmann, and Gao (1999) modified the SRN by adding additional units and connections for representing the transfer domain, and for learning the mapping from training to transfer domains. This model has been able to explain a significant set of results from the artificial grammar learning domain, in which this issue of transfer is a central question. Thus the “overlap” problem is solved by learning, but in a manner that avoids representation of the common rule. One behavioral result of this mapping strategy is that transfer is not immediate, but requires significant exposure to the new material for the mapping to be learned.

It remains likely, however, that the “rule” vs. “instance” distinction is behaviorally and neurophysiologically valid (reviewed in Dominey et al., 1998). In this context, we observed in human adults that the serial and abstract structure of sequences, such as ABCBAC and DEFEDF, can be learned independently (Dominey et al., 1998). In implicit (naïve) conditions, subjects learned serial structure, but were unable to exploit knowledge of the underlying abstract structure 123-213 when exposed to a new “isomorphic” sequence based on this abstract structure. Indeed, only when subjects were explicitly aware of a possible underlying abstract structure were they able to learn both the serial and abstract structure and rapidly transfer the abstract knowledge to new isomorphic sequences. When exposed to this task, the TRN learned only the serial structure, independent of the abstract structure. This is because the network lacks a representation of the internal relations that characterize abstract structure. To address this representational deficit, we modified the TRN to contain a set of short-term memory (STM) components, that store the 7 ± 2 previous sequence elements, and a recognition function applied to this STM to detect repetition relations, with the recurrent network operating on these relations. In this context, ABCBAC is represented as “u,u,u, n-2, n-4, n-3” where “u” corresponds to unique or nonrepeating, and “n-2” corresponds to repetition of the element 2 positions behind (Dominey et al., 1998). The thus modified abstract recurrent network (ARN) operates on these abstract sequence representations rather than their serial structure. This is consistent with the idea that, to the extent that the training and transfer domain do not overlap, a recurrent network is not enough, and additional hardware is required. For Dienes et al. (1999) the additional hardware was used to construct the mapping between training and transfer domains. In Dominey et al. (1998) it allowed representation of the internal or “abstract” structure of the rule that describes the general mapping itself.

Neuropsychological Evidence for the Serial/Abstract Dissociation

If the hypothesis that serial and abstract structure are treated by dissociable neurophysiological processes is correct, then we should be able to isolate patient groups that display this dissociation. In this context, Parkinson’s disease is characterized by motor disorders of akinesia, tremor, and rigidity that result from a massive destruction of midbrain dopamine-producing neurons. This deterioration has significant impact on the functional organization of

the frontostriatal system that is the neuroanatomical correlate of the TRN sequence learning model. Thus, numerous studies indicate that the implicit learning of the serial structure of sensorimotor sequences is impaired in these patients. Interestingly, we demonstrated that these patients retained a significant capability to acquire and transfer knowledge of abstract sequential structure to new isomorphic sequences in a serial reaction time task, suggesting that while implicit sequence learning relies on an intact frontostriatal system, explicit learning of abstract structure does not (or less so). In contrast, it is known that schizophrenic patients tend to perform well on automated, implicit processing, with more difficulty on explicit attentional processing, functionally associated with a hypofrontality. Accordingly, we observed that schizophrenic patients displayed an intact capability to learn serial structure, but failed to learn abstract structure. These results from simulation and neuropsychological studies support the hypothesis that processing of serial and abstract structure rely on dissociable neural mechanisms. These and related studies are reviewed in Dominey et al. (1998).

Relation to Language Processing

Although language clearly requires task specific capacities, it is likely that more generalized capabilities to process serial, temporal, and abstract sequential structure could also come into play. This applies both for learning to segment speech into words (Christiansen et al., 1998), and in the use of abstract generative rules for structural transformations.

Syntactic comprehension is the process of determining “who did what to whom” from a purely syntactic analysis. Considering the sentences “Sally¹ was introduced to Bill² by John³” vs. “John³ introduced Sally¹ to Bill²,” we see that ordering of the agent (John), object (Sally), and recipient (Bill) vary in these active and passive sentences. In a simplified manner, if we consider syntactic comprehension as a recovery of the canonical (active) structure, then the passive to active transformation can be characterized by the abstract rule 123-312. In this context we can predict that if syntactic comprehension shares a common neurophysiological basis with abstract structure manipulation, then performance deficits in the two tasks should be correlated.

In order to test this prediction, we compared performance in syntactic comprehension and in abstract sequential structure processing in a population of aphasic patients with syntactic comprehension deficits. Linear regression revealed that performance in the two tasks was highly correlated, suggesting a common underlying neurophysiological basis. This observation was likewise replicated in a population of schizophrenic patients that displayed correlated deficits in their processing of syntactic and abstract structure (Lelakov et al., 2000).

While such correlation results argue for shared neurophysiological processing, they do not rule out explanations based on a more global cognitive deficit. In order to examine this issue more directly one can exploit brain imagery techniques during the performance of these abstract and syntactic tasks. In this context, we note that the on-line identification of syntactic structures (e.g., active, passive, relative, etc.) is largely guided by closed class function words (e.g., to, by, the) that thus play a crucial role in allowing a parser to apply syntactic knowledge. ERP studies have demonstrated that grammatical function words evoke a left anterior negativity (LAN) between 400–600 ms (reviewed in Hoen and Dominey, 2000). In a related study, Hoen and Dominey (2000) examined brain potentials associated with the processing of special “function” symbols that trigger one of two possible abstract rules that will apply in a cognitive sequencing task. The objective was to determine if these nonlinguistic function symbols would be processed in the same neural networks as grammatical function words in natural lan-

guage. In order to examine this process in a nonlinguistic sequencing context, two abstract structure conditions were studied (1) ABCXBCA and (2) ABCYDEF. In condition (1) the function symbol “X” indicates that the second triplet will be a systematically transformed version of the first triplet. In condition (2) the function symbol “Y” indicates that the second triplet will have no relation to the first. When brain responses to the function symbols X and Y were compared, a relative left anterior negativity was observed for X vs. Y, with a temporal and topographic scalp localization quite similar to the LAN observed for function word processing (Hoen and Dominey, 2000).

These patient and ERP studies suggest a framework in which grammatical markers, either in the form of distinct function words, or as morphological markers, would indicate the appropriate syntactic frames (abstract structure), while these syntactic frames would be applied to open class words for syntactic comprehension. Interestingly, this framework maps quite naturally onto the dual-process TRN/ARN model for the treatment of serial/temporal and abstract structure. Function words (or their morphological equivalents) are processed in the recurrent network of the TRN, thus defining the syntactic context. Open class nouns are stored in the STM components of the abstract network ARN, and based on the syntactic context, these STM components (and thus the contained nouns) are then linked with their corresponding thematic role and retrieved appropriately. In this configuration, the model is able to learn the mapping between syntactic structure and the corresponding conceptual or thematic structure in order to successfully complete a standard syntactic comprehension task. Although this represents an interesting extension of sequential cognition toward language processing, it is clearly a preliminary first step that leaves much work ahead.

Discussion

At the outset of this article, it was suggested that behavioral sequences can be considered in terms of their serial, temporal, and abstract structure. A behavioral sequence learning framework was thus established in which serial and temporal structure are represented in a temporal recurrent network (TRN) with recurrent connections implemented in corticocortical connections of the frontal cortex. State activity in the recurrent network is then functionally associated with behavioral responses via modifiable corticostriatal projections. A dissociated abstract recurrent network (ARN) that is required for manipulating abstract structural relations is potentially implemented in a distributed network that includes the perisylvian cortex in and around Broca’s area. The proposal that transfer of sequence knowledge to a new domain requires additional, dissociable, processes remains an open debate. Though the approaches

are somewhat different, both Dienes et al. (1999) and Dominey et al. (1998) argued that recurrent networks alone could not address this transfer, which requires additional representational capabilities. Indeed, it is likely that both approaches of between-domain mapping and abstract rule representation are neurophysiological realities.

Road Map: Cognitive Neuroscience

Related Reading: Prefrontal Cortex in Temporal Organization of Action; Speech Processing: Psycholinguistics; Temporal Pattern Processing

References

- Christiansen, M. H., Allen, J., Seidenberg, M. S., 1998, Learning to segment speech using multiple cues: A connectionist model, *Lang. Cognit. Proc.*, 13:221–268.
- Cleeremans, A., Servan-Schreiber, D., and McClelland, J. L., 1989, Finite state automata and simple recurrent networks, *Neural Comp.*, 1:372–381.
- Cleeremans, A., 1993, *Mechanisms of Implicit Learning: Connectionist Models of Sequence Processing*, Cambridge, MA: MIT Press. ♦
- Curran, T., and Keele, S. W., 1993, Attentional and nonattentional forms of sequence learning, *J. Exp. Psych.: Learning, Mem. Cog.*, 19(1):189–202.
- Dienes, Z., Altmann, G., and Gao, S.-J., 1999, Mapping across domains without feedback: A neural network model of transfer of implicit knowledge, *Cognit. Sci.*, 23:53–82.
- Dominey, P. F., Lelekov, T., Ventre-Dominey, J., Jeannerod, M., 1998, Dissociable processes for learning the surface and abstract structure of sensorimotor sequences, *J. Cognit. Neurosci.*, 10(6):734–751. ♦
- Dominey, P. F., 1998, Influences of temporal organization on transfer in sequence learning: Comments on Stadler (1995) and Curran and Keele (1993), *J. Exp. Psychol.: Learning, Mem. Cog.*, 24(1):234–248.
- Dominey, P. F., and Ramus, F., 2000, Neural network processing of natural language: I. Sensitivity to serial, temporal and abstract structure of language in the infant, *Lang. Cognit. Proc.*, 15(1):87–127.
- Elman, J. L., 1990, Finding structure in time, *Cognit. Sci.*, 14:179–211. ♦
- Hoen, M., and Dominey, P. F., 2000, ERP analysis of cognitive sequencing: A left anterior negativity related to structural transformation processing, *Neuroreport*, 28;11(14):3187–3191.
- Lelekov, T., Franck, N., Dominey, P. F., and Georgieff, N., 2000, Cognitive sequence processing and syntactic comprehension in schizophrenia, *Neuroreport*, 14;11(10):2145–2149.
- Marcus, G. F., Vijayan, S., Bandi Rao, S., and Vishton, P. M., 1999, Rule learning by seven-month-old infants, *Science*, 283(5398):77–80.
- Nissen, M. J., and Bullemer, P., 1987, Attentional requirement of learning: Evidence from performance measures, *Cognit. Psychol.*, 19:1–32. ♦
- Pearlmutter, B. A., 1995, Gradient calculation for dynamic recurrent neural networks: A survey, *IEEE Transactions on Neural Networks*, 6(5):1212–1228. ♦
- Stadler, M. A., 1995, The role of attention in implicit learning, *J. Exp. Psychol.: Learning, Mem. Cog.*, 21:674–685.

Short-Term Memory

Emmanuel Guigon, Etienne Koechlin, and Yves Burnod

Introduction

It is generally agreed that two temporally distinct neural processes contribute to the acquisition and expression of brain functions. Transient variations in membrane potential (neuronal activity), on a time scale of milliseconds, reflect the flow of information from neuron to neuron and define the function of neuronal networks. These variations can result in long-lasting (and possibly permanent)

alterations in neuronal operations, for instance through activity-dependent changes in synaptic transmission.

There is now strong evidence for a complementary process acting over an intermediate time scale. A wide variety of temporal patterns of activity are actively generated by neurons and local circuits of neurons; an example is the transformation of transient inputs into long-lasting sustained or oscillatory activity. Experimental studies in invertebrates have demonstrated that such tem-

poral patterns produce motor programs and are generated both by the molecular properties of each neuron and by the connectivity of the local network (Harris-Warrick and Marder, 1991; Marder et al., 1996). In vertebrates, long-lasting activities are neural correlates of transient memory processes. These patterns of activity allow past events to be represented and behavioral reactions to future, predictable events to be prepared (see PREFRONTAL CORTEX IN TEMPORAL ORGANIZATION OF ACTION). That these patterns also result from both the intrinsic properties of single neurons and the synaptic interactions between neurons is now well recognized (Llinás, 1988; Marder et al., 1996; Durstewitz, Seamans, and Sejnowski, 2000).

In this article, we discuss cellular and neural network mechanisms that could be involved in the formation of short-term memory (STM) traces in the vertebrate brain. We address three issues: (1) What are the different types of STM traces? (2) How do intrinsic and synaptic mechanisms contribute to the formation of STM traces? (3) How do STM traces translate into long-term memory representations of temporal sequences? We note that we are concerned here neither with exact definitions and properties of psychological concepts nor with detailed biophysical or biochemical mechanisms involved in the characterization of short-term memory processes, but only with computational mechanisms underlying these processes. We also note that these mechanisms may well underlie a wide variety of seemingly different biological processes (e.g., emergence of orientation selectivity in visual cortex, dynamics of head-direction cells in the limbic system, directional tuning in motor cortex) and so may be relevant to understanding brain functions.

Types of Neural Short-Term Memory Traces

As shown in Figure 1, two broad types of STM traces exist. In the first type (Figure 1A), transient inputs are transformed into long-lasting activity patterns. The output traces could represent a membrane potential, a discharge frequency, or any biophysical or biochemical variable (e.g., intracellular calcium concentration). Ideally, the level of maintained activity would be proportional to the intensity of the input stimulus (intensity memory; Figure 1A₁), or tuned about a preferred (spatial) stimulus (spatial memory or

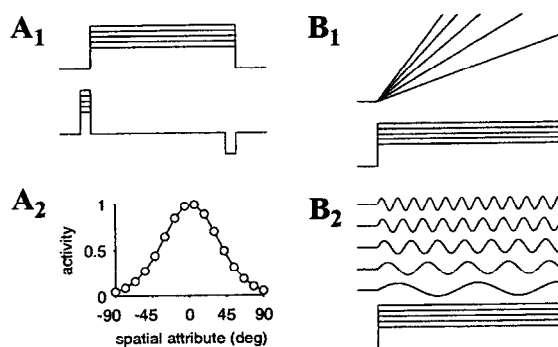


Figure 1. Types of STM traces. A₁, Intensity memory. Intensity of a transient input (lower trace) is translated into a long-lasting activity (upper trace) of equal (or proportional) amplitude. Five activity traces are shown, corresponding to five input intensities (in the same order). Horizontal time scale and vertical intensity scale are not specified (same as for B₁ and B₂). A₂, Spatial memory. The memorized value of a spatial attribute (here 0°) is represented by a tuned activity distribution in a population of neurons selective to this attribute. B₁, Activity ramp. A constant input (lower trace) is translated into a time-varying linearly increasing activity (upper trace). The slope of output activity is proportional to input intensity. B₂, A constant input is translated into oscillations (the output traces are shown separately for clarity). Oscillatory frequency is proportional to input intensity.

memory field; Figure 1A₂). Spatial and intensity memory mechanisms are relevant to working memory, or the ability to hold relevant information in memory for future utilization in the guidance of behavior. Neural correlates of working memory are found mainly in the anterior regions of the cerebral cortex (e.g., prefrontal cortex) as stimulus-selective sustained neuronal discharges (see PREFRONTAL CORTEX IN TEMPORAL ORGANIZATION OF ACTION). Complete references on the issue of working memory models can be found in Durstewitz et al. (2000).

In the second type of STM trace (Figure 1B), constant inputs are transformed into time-varying outputs, such as ramps with different slopes (Figure 1B₁) or oscillations at different frequencies (Figure 1B₂) or of different amplitudes (not shown). Again, characteristics of the output patterns (slope, frequency, amplitude) should be related to the intensity of the input. Activity ramps are found as correlates of preparatory and anticipatory processes in sensorimotor and cognitive behaviors. They are ubiquitous in cortical parietal and frontal region, and could participate in muscular recruitment, preparation for response, and decision-making processes (Hanes and Schall, 1996). Little attention has been paid to the formation of ramps. Oscillatory activities are mentioned here because they are typical STM traces to which a central role in many behavioral processes has been attributed (see SLEEP OSCILLATIONS). Cellular and network mechanisms of oscillations are dealt with in the literature (see OSCILLATORY AND BURSTING PROPERTIES OF NEURONS) and are not addressed here.

Many other STM traces could be built by combining these two types. Furthermore, in physiological recordings, additional time-varying components would be found in inputs and outputs corresponding to different types of variability in neural processing.

Cellular and Network Mechanisms of STM Traces

Sustained Activity

The basic mechanism for the formation of maintained activities is a neuron with a recurrent excitatory connection

$$\tau \frac{dI}{dt} = -I + wf(I) + I_{in} \quad (1)$$

where I is a variable that could represent the total synaptic current, $f(I) = 1/[1 + \exp(s(0.5 - I))]$, the firing rate of the neuron, w is the weight of the recurrent connection, and I_{in} is the input current. The bifurcation diagram of this equation was plotted with I_{in} as a parameter (Figure 2A). For $I_{in} < I_1$ and $I_{in} > I_2$, the equation has a single, stable, fixed point. Elsewhere there are three fixed points: two are close to the minimum and maximum firing rate, respectively, and are stable (lower and upper branches in Figure 2A). The third one is unstable (dashed middle branch). Transient inputs elicit transition between the stable states (Figure 2B).

This model is illustrative of a general class of neural networks in which persistent states arise from reverberating activity through recurrent excitatory loops (see COMPUTING WITH ATTRACTORS). The main drawback of these models is that the level of maintained activities is close to the neural saturation level, and the resting level is silent (Figure 2A), in contradistinction to physiological observations. More realistic persistent activities are found when excitation and inhibition are represented by different neuronal populations (Durstewitz et al., 2000). Furthermore, low- and high-frequency persistent states can coexist when inhibition slightly dominates excitation.

The models just discussed describe neural processing in terms of firing rate or synaptic current, whereas a physiologically realistic representation is defined by equations governing membrane potential (i.e., Hodgkin-Huxley equations and equations for synaptic inputs; see SYNAPTIC INTERACTIONS). Thus, an open question is

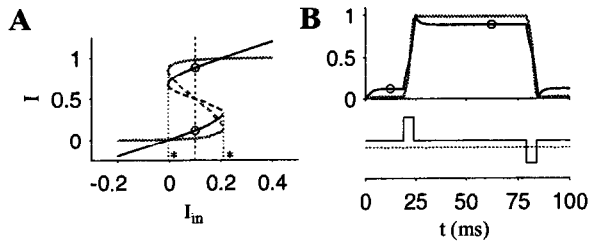


Figure 2. A, Bifurcation diagram of Equation 1. The plot depicts the value of equilibrium state synaptic currents (black curve) and corresponding firing rates (gray curve). States belonging to plain (dashed) lines are stable (unstable). Vertical dotted lines delimit a region with three equilibrium states and define input currents I_1 and I_2 (asterisks). B, Activity profile (same conventions as in A) for $I_{in} = 0.1$ (vertical dashed line in A) and transient excitatory and inhibitory inputs (lower trace). Steady states are indicated by \circ . Parameters were $\tau = 2$, $s = 10$, $w = 0.8$.

whether similar properties would be found in a biophysically realistic model of a spiking neuron. Simulations show that the ability of a neural network to maintain robust delay activity at physiological rates (e.g., 15–20 Hz) depends on the nature of synaptic transmission. In fact, the largest component of the synaptic transmission, mediated by AMPA receptors, has a fast decay, which leads to persistent discharges at frequencies above 50 Hz (Wang, 1999). The contribution of slow synaptic transmission through NMDA receptors could help bypass this effect and reduce the frequency of persistent activity to the required level (Wang, 1999). However, it is unknown whether the density of NMDA receptors is large enough to play such a role.

We discussed how maintained activity can result from recurrent interactions within neuronal populations. Alternatively, maintained activity could correspond to the depolarized state of an intrinsically bistable neuron (Marder et al., 1996). Intrinsic bistability is characterized by the existence of two or more stable states (e.g., a hyperpolarized state and a more depolarized state in which the neuron discharges), with the transition from one state to the other effected by transient synaptic events (Marder et al., 1996). Numerous examples of bistability have been described in the literature, both in invertebrates (e.g., *Aplysia*, crab stomatogastric ganglion) and in vertebrates (spinal cord, cerebellum, thalamus). The cellular bases of bistability generally involve a low-threshold persistent inward conductance, i.e., a depolarizing conductance that activates in the subthreshold range and does not inactivate (see ION CHANNELS: KEYS TO NEURONAL SPECIALIZATION). If the neuron is endowed with a spiking mechanism, the depolarized state corresponds to the discharge of action potentials; otherwise it is a plateau potential.

The models described so far maintained persistent activities in an all-or-none fashion, at a level prescribed by their structure. These observations support the concept of a subset of strongly and uniformly connected neurons representing a discrete attractor (see CORTICAL HEBBIAN MODULES). In the following paragraphs we show that adequate choice of recurrent synaptic interactions allows memory encoding of continuously valued variables (intensity or spatial memory; see Figure 1A).

Intensity memory (Figure 1A₁) can be addressed in the framework of linear recurrent neural networks, i.e.,

$$\tau \frac{dI}{dt} = -I + WI + I_{in} \quad (2)$$

where I is the N -dimensional vector of output activities and W is a symmetric synaptic matrix. This equation, which is a multidimensional linear generalization of Equation 1, can be solved ex-

plicitly for I ,

$$I(t) = \sum_{i=1}^N a_i(t) e_i$$

where $\{e_i\}$ are the eigenvectors of W . Persistent activity appears in the case where one eigenvalue (index k) of W is equal to 1 and all the other eigenvalues are smaller than 1. The solution becomes

$$I(t) \approx \frac{e_k}{\tau} \int_0^t I_{in}(t') \cdot e_k dt' \quad (3)$$

This equation shows that the network can hold a faithful memory of the amplitude of a transient input. However, this property is lost when the synaptic matrix is even slightly perturbed. This would also be the case in a nonlinear version of this model.

This principle was used in the framework of conductance-based spiking models by Seung et al. (2000) to explore brainstem networks involved in the control of eye position. In this model, brainstem neurons are integrators that convert transient signals driving changes in eye position into a persistent memory of eye position (Figure 3A).

The linear recurrent network described by Equation 2 can also generate *spatial memory* profiles. This occurs when (1) each neuron i is identified by a periodic parameter θ_i (e.g., a preferred direction in $[0; 2\pi]$), (2) the $\{\theta_i\}$ are uniformly distributed in $[0; 2\pi]$, and (3) the synaptic weight between neurons i and j is $W_{ij} = \cos(\theta_i - \theta_j)$. In this case, W has only two non-zero eigenvalues equal to 1, and acts as a filter that suppresses all harmonics ≥ 2 in the input signal. Thus, the network generates and maintains a cosine distribution of activity from any nonuniform transient input. The constraint on the eigenvalues of W can be relieved in a nonlinear version of Equation 2:

$$\tau \frac{dI}{dt} = -I + W[I]_+ + I_{in} \quad (4)$$

where $[]_+ ([u]_+ = u \text{ if } u \geq 0, \text{ otherwise } [u]_+ = 0)$ translates current into firing rate. In this case, a spatially selective activity profile can persist in the presence of a constant background (Figure 3B). This behavior appears in the case of strong, spatially modulated excitatory connections and corresponds to the existence of a continuous line of stable states (Hansel and Sompolinsky, 1998).

A realistic implementation of spatial memory was described by Compte et al. (2000). Their network involved (1) excitatory and inhibitory neurons modeled as leaky integrate and fire units (see

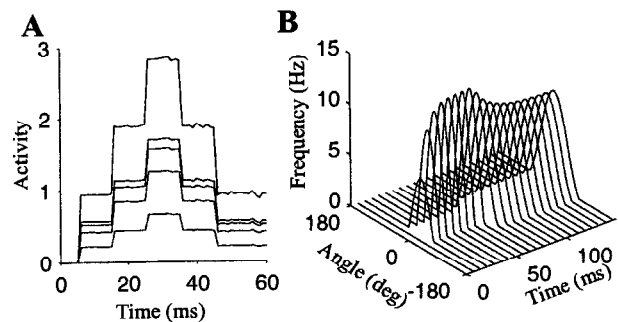


Figure 3. A, Intensity memory. Simulation of Equation 3 for five neurons. The eigenvector e_k was $[0 \ 0.25 \ 0.5 \ 0.75 \ 1]$. Transient inputs (duration 2 ms) were delivered at times 5, 15, 25 (excitatory), and 35, 45 (inhibitory). B, Spatial memory. Simulation of Equation 4. Matrix W was made of local Gaussian excitation and global inhibition. A tuned activity profile (width 40°) was presented at time 10 and replaced at time 50 by an untuned profile of the same amplitude.

SINGLE-CELL MODELS), (2) spatially structured connections between the excitatory neurons, and (3) slow (NMDA) synaptic transmission (Wang, 1999). The model displayed both low spontaneous activity and robust stimulus-evoked selective persistent discharges at physiological frequencies ($\sim 20\text{--}30$ Hz). In the preceding model (Equation 4), spatial pattern formation resulted from a continuous bifurcation. On the other hand, there is genuine network bistability in the present model, which authorizes transition between resting and activated states by transient excitatory inputs. Compte et al. (2000) observed a drift in time of persistent activity patterns in the presence of noise, which resulted in a degraded memory of encoded stimuli. In fact, in all models, the spatial patterns are only marginally stable (Hansel and Sompolinsky, 1998).

An attractive hypothesis would be that both synaptic and intrinsic properties contribute to the formation of persistent activity. Lisman, Fellous, and Wang (1998) proposed that NMDA receptor-mediated bistability could participate in the maintenance of selective working memory activity. Camperi and Wang (1998) used conditional bistability in a continuous attractor network (Equation 4) and showed that it can contribute to the stability of maintained activities against perturbations, although it was not involved in their maintenance per se.

On the whole, these mechanisms provide reasonable clues to how sustained activities can be maintained in neuronal populations. However, several questions remain open: (1) All of the models have built-in instability and require finely tuned synaptic weights to work appropriately. Is this instability a characteristic feature of sustained discharges in the nervous system? How could this instability be removed? Which mechanisms allow the development and maintenance of exact synaptic structures? (2) The models have many features in common and apply to the emergence of orientation selectivity in visual cortex, dynamics of head-direction cells in the limbic system, directional tuning in motor cortex, and persistent activities in prefrontal cortex. Are there definite differences in the neural substrate of these functions? For instance, is the putative role of slow synaptic transmission identified by some models a characteristic feature of prefrontal cortical circuits?

Activity Ramp

When a constant current is injected in a neuron, a time-varying pattern of activity is observed that depends on passive properties of the neuron (membrane time constant) and active membrane characteristics (voltage-gated ionic conductances). The former effect is illustrated by the voltage response of a passive membrane

$$\frac{dV}{dt} = -\frac{V}{\tau} + I$$

for different values of τ and different I . The time to reach a threshold V_θ is given by

$$T_\theta = -\tau \ln \left(1 - \frac{V_\theta}{\tau I} \right)$$

This relation is strongly nonlinear and shows that neither I nor τ can efficiently be used to specify a duration. At best it could be used for durations below 50 ms.

The same is approximately true in a Hodgkin-Huxley model (see AXONAL MODELING) because the sodium and potassium conductances of the action potential are weakly activated in the subthreshold range. The time to reach a given frequency is not yet an appropriate timing mechanism, because steady-state discharge settles within a few time constants.

At the single neuron level, robust STM properties arise from the presence of a slowly inactivating potassium (Ks) conductance (Marder et al., 1996; Delord et al., 2000). The functioning principle

of the Ks conductance is the following. It creates a slowly decaying hyperpolarizing current whose initial level can be specified by prior conditioning of the neuron. For instance, prior hyperpolarization sets a large persistent outward current that slows down the rate of membrane potential changes in the subthreshold range during a subsequent depolarization. Figure 4A shows that the latency-to-the-first-spike can be up to 10 s in the presence of a Ks conductance with an inactivation time constant of 2 s, and the relationship between the injected current and the latency is close to linear for a latency of up to approximately 7 s.

The Ks conductance also influences the suprathreshold behavior of the neuron. The discharge frequency gradually increases toward its steady-state level as the Ks current decays (Figure 4B). Both the initial and final frequency increase with the level of injected current, which results in a modest change in the slope of the time-frequency curve with the injected current (Figure 4C). Recruitment at variable rates, as described in Figure 1B, can be approached by combining the effect of Ks conductance and synaptic interactions in a population of uniformly connected neurons (Figure 4B) (Delord et al., 2000). In this case, because of recurrent excitation, a smaller amount of injected current is required to obtain a given steady-state frequency. Thus the initial frequencies are lower and vary in a smaller range. Accordingly, the slope is more strongly modulated by the injected current than in the absence of synaptic interactions (Figure 4C). The strength of this modulation is directly controlled by the strength of synaptic weights (Figure 4D). Thus, adaptive recruitment at variable rates is made possible by the simultaneous action of synaptic and intrinsic properties in a neural network. Interestingly, slowly inactivating potassium conductances are found in neurons of most regions of the central nervous system with a time constant ranging from hundreds of milliseconds to several tens of seconds (Llinás, 1988).

Could a similar property be obtained by purely synaptic effects? In fact, the linear recurrent network described by Equation 2 has the required property (Figure 3A). The formation of persistent activities begins by a linear ramp with a slope proportional to the amplitude of the input current (Equation 3). However, as mentioned

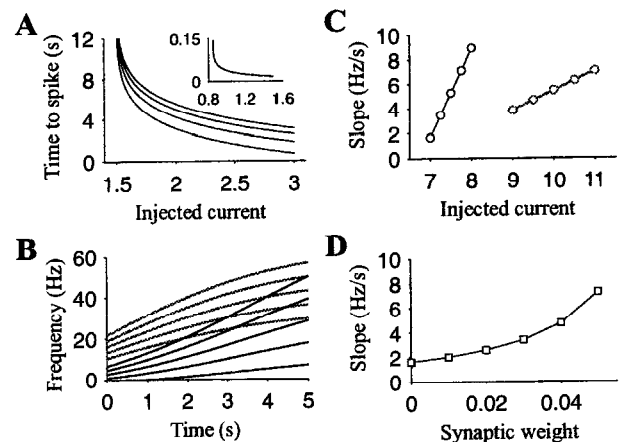


Figure 4. A, Time to the first spike as a function of the injected current in the presence of a Ks conductance in a Hodgkin-Huxley model. The curves correspond to different initial level of availability of the Ks conductance (maximal for the upper curve). The result in the absence of Ks conductance is shown at upper right. B, Time course of frequency increase for different levels of injected current in a recurrent (dark lines) and nonrecurrent (gray lines) network. C, Slope of the frequency increase as a function of the injected current (linear regression on the results of B). D, Slope of the frequency increase as a function of the synaptic weight in the recurrent network. Details on the methods can be found in Delord et al. (2000).

earlier, exactly tuned synaptic weights are necessary to the proper functioning of the network. It is unclear whether the nervous system can reach the required degree of accuracy in the adjustment of synaptic weights (Seung et al., 2000).

From Short-Term to Long-Term Memory Traces

Synaptic plasticity is a central mechanism in models of learning and memory (see HEBBIAN SYNAPTIC PLASTICITY). The most popular approach involves shaping functions of neural networks by activity-dependent modification of synapses based on Hebbian learning rules. Accordingly, information stored in long-term memory reflects correlation (i.e., temporal contiguity) between transient neuronal activities on a time scale of 0–100 ms. Sutton and Barto (1981) recognized that learning rules based on temporal contiguity are inappropriate to represent temporal dependencies in, for example, the framework of classical conditioning. They proposed that STM traces (synapse-specific and output traces) are involved in the acquisition of new conditioned behaviors. This principle has led to the development of the temporal difference algorithm, which provides a powerful way to learn predictions of future events (see REINFORCEMENT LEARNING). In this framework, STM traces are translated into a long-term representation of the temporal structure of behavior. A related approach applied to the prefrontal cortex is described in Guigon et al. (1995).

Discussion

STM traces become essential as soon as the time scale of behavior extends beyond the duration of phasic signaling (e.g., ~0–500 ms)—in other words, in almost any behavioral situation. We discussed cellular and network mechanisms involved in the formation of STM traces. We described simplified network models that provide mathematical conditions for the formation and stability of STM traces, and more detailed realistic models with which to assess the biophysical basis of these phenomena.

Despite impressive results, our understanding of these mechanisms is far from complete. First, each neuron is endowed with a wealth of intrinsic properties (Llinás, 1988), very few of which have been considered in models. The respective contribution of synaptic and intrinsic factors is unknown. Second, persistent discharges and ramps constitute a small subset of the rich dynamic repertoire of neural populations observed in vivo, and it is unclear how they can be combined to form more complex memory traces. Third, the great majority of models fail to be robust facing noise and inexact tuning of parameters (e.g., synaptic weights). A future

challenge is to relate these pieces of memory to cognitive functions that are very demanding of temporary storage and manipulation of information, such as planning, reasoning, and language use.

Road Map: Dynamic Systems

Background: I.3 Dynamics and Adaptation in Neural Networks

Related Reading: Amplification, Attenuation, and Integration

References

- Camperi, M., and Wang, X.-J., 1998, A model of visuospatial working memory in prefrontal cortex: Recurrent network and cellular bistability, *J. Comput. Neurosci.*, 5:383–405.
- Compte, A., Brunel, N., Goldman-Rakic, P., and Wang, X.-J., 2000, Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model, *Cereb. Cortex*, 10:910–923.
- Delord, B., Baraduc, P., Costalat, R., Burnod, Y., and Guigon, E., 2000, A model study of cellular short-term memory produced by slowly inactivating potassium conductances, *J. Comput. Neurosci.*, 8:251–273.
- Durstewitz, D., Seamans, J., and Sejnowski, T., 2000, Neurocomputational models of working memory, *Nature Neurosci. Suppl.*, 3:1184–1191.
- Guigon, E., Dorizzi, B., Burnod, Y., and Schultz, W., 1995, Neural correlates of learning in the prefrontal cortex of the monkey: A predictive model, *Cereb. Cortex*, 5:135–147.
- Hanes, D., and Schall, J., 1996, Neural control of voluntary movement initiation, *Science*, 274:427–430.
- Hansel, D., and Sompolinsky, H., 1998, Modeling feature selectivity in local cortical circuits, in *Methods in Neuronal Modeling: From Ions to Networks*, 2nd ed. (C. Koch and I. Segev, Eds.), Cambridge, MA: MIT Press, pp. 499–567.
- Harris-Warrick, R., and Marder, E., 1991, Modulation of neural networks for behavior, *Annu. Rev. Neurosci.*, 14:39–57.
- Lisman, J., Fellous, J.-M., and Wang, X.-J., 1998, A role for NMDA-receptor channels in working memory, *Nature Neurosci.*, 1:273–275.
- Llinás, R., 1988, The intrinsic electrophysiological properties of mammalian neurons: Insights into central nervous system function, *Science*, 242:1654–1663.
- Marder, E., Abbott, L., Turrigiano, G., Liu, Z., and Golowasch, J., 1996, Memory from the dynamics of intrinsic membrane currents, *Proc. Natl. Acad. Sci. USA*, 93:13481–13486.
- Seung, H., Lee, D., Reis, B., and Tank, D., 2000, Stability of the memory of eye position in a recurrent network of conductance-based model neurons, *Neuron*, 26:259–271.
- Sutton, R., and Barto, A., 1981, Toward a modern theory of adaptive networks: Expectation and prediction, *Psychol. Rev.*, 88:135–170.
- Wang, X.-J., 1999, Synaptic basis of cortical persistent activity: The importance of NMDA receptors to working memory, *J. Neurosci.*, 19:9587–9603.

Silicon Neurons

Rodney Douglas and Christof Rasche

Introduction

Silicon neurons are analog electronic circuits fabricated in complementary metal-oxide-silicon (CMOS) using very large-scale integration (VLSI) methods. CMOS is a medium for manufacturing transistors whose conductivity can be altered by an applied electric field. This technology is commonly used to construct the digital circuits found in general-purpose computers, but the silicon neurons discussed in this article are not a kind of digital computer. Instead, the same CMOS VLSI technology is used to construct

analog circuits whose physics is analogous to the physics of membrane conductivity. This analogy permits the circuits to emulate the electrophysiological behavior of biological neurons in real time, while the high component density offered by VLSI technology provides a means of fabricating large networks of silicon neurons.

Neuronal systems are difficult to model because they are composed of large numbers of nonlinear elements and have a wide range of time constants. Consequently, their mathematical behavior can rarely be solved analytically. The usual approach is to simulate

these problems on a general-purpose digital computer (Koch and Segev, 1998). But for any given computer, the speed of these simulations is limited by the shortest time constant in the problem. Furthermore, the simulation time slows dramatically as the number and coupling of elements increase. By contrast, silicon neurons operate in real time, and the speed of the network is independent of the number of neurons or their coupling. Thus, networks of silicon neurons are especially suited to the investigation of questions that arise from the real-time interaction of the system with its environment. Nevertheless, the design of special-purpose hardware is a significant investment, particularly if it is analog hardware, since analog VLSI (aVLSI) design is still very much an art form.

Analog VLSI has a controversial role in the study of neural computation. This controversy arises out of a debate over the role of precision in computation. Digital computation is guaranteed precise to the number of bits used in the computation. However, the most compact analog circuits have low precision as a result of uncertain calibration between transistors. Some proponents of neuromorphic aVLSI design claim that these circuits provide a natural route for exploring the principles of biological computing, which must also make do with low precision (Mead, 1989). Unlike the ideal components used in conventional computers, real neurons are not homogeneous. Even within a morphological class such as the pyramidal cells of the cerebral cortex, they show a wide range of behavior. They are poorly insulated conductors; they have a variety of nonlinear conductance elements and large amounts of stray capacitance; they are sensitive to environmental changes; and a significant fraction of them malfunction or stop during the operational life of the system. Nevertheless, the smallest vertebrate brain is vastly more competent at interaction with the real world than are our most elaborate supercomputers. From the level of conductances, through synapses to neurons and networks, the nervous system elements obtain precision, speed, and computational power using imperfect elements. Understanding the architectures and adaptive processes that allow the nervous system to extract precise information from a noisy and ambiguous environment with uncalibrated components is a central problem of computational neuroscience (see SYNAPSE TRANSMISSION). Analog VLSI circuits have similar intrinsic variability, and so synthesis of silicon neurons is a method of exploring the principles of computations that must use unreliable components. The philosophy of neuromorphic engineering is that the medium of computation is an intrinsic part of the computation itself.

Mapping Neurons into CMOS

The strategy for mapping neurons into CMOS varies between research groups, and this article will not review the full range of options that are being explored. Instead, we will focus on a few examples that are representative of the various types. The feature that distinguishes a silicon neuron from an ANN neuron is that it includes a large number of time constants. The silicon neuron's dynamical complexity implies that the input-output relationship of silicon neurons cannot be encapsulated by a single sigmoidal function; instead, a given synaptic input will have a different effect on the output, depending on where and when it is applied.

Conceptually, the neuron can be divided into four parts: the dendrite, which receives inputs; the soma, which translates the inputs to an output; the axon, which distributes the output; and the synapses, which transmit the output to the target neurons. At a more physical level, traditional modelers of biological neurons have divided the continuous neuronal membrane of the dendrites and soma into a series of compartments to facilitate numerical computation (Koch and Segev, 1998) (see DENDRITIC PROCESSING). Each compartment is considered to be isopotential and spatially uniform in

its properties. The connectivity of the compartments mirrors the spatial morphology of the modeled cell.

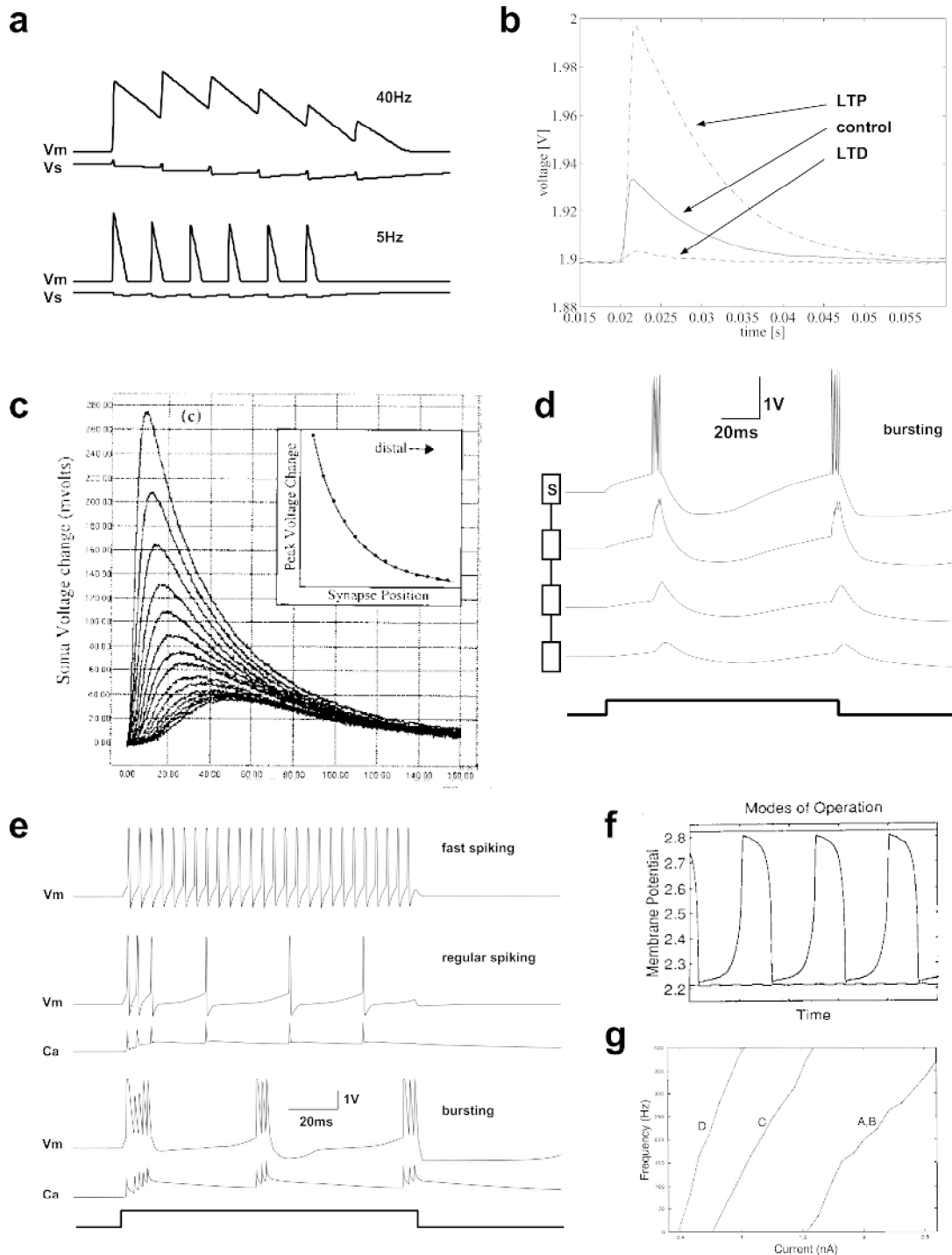
Elias has constructed neuromorphic VLSI neurons with 112 passive compartments that model the leakiness of the cellular membrane and the axial resistance of the intracellular medium using space-efficient switched capacitors to implement resistances (Elias and Northmore, 1999). Each compartment provides passive temporal filtering of the inputs. More recently, we have incorporated circuits that model voltage-dependent conductances into the dendritic compartments (Figure 1D). The resolution of the segmentation is a compromise between the questions that must be addressed by the model, the resources required by each compartment, and error tolerance. For example, neurons with between 5 and 30 compartments are a common compromise for digital simulations of cortical and hippocampal circuits (Koch and Segev, 1998).

The simplest electronic somatic model is a single passive compartment that produces a digital spike event when it is charged to a threshold voltage. This integrate-and-fire neuron (see SINGLE-CELL MODELS) has been used for various purposes ranging from sensors to general network simulations. For example, Elias uses an integrate-and-fire model for his dendrites (Elias and Northmore, 1999). Boahen extended a silicon integrate-and-fire neuron to include a spike-frequency adaptation mechanism. This version is used in a neuromorphic retina.

The oscillatory character of neurons is compactly expressed in mathematical models such as the FitzHugh-Nagumo or Morris-Lecar model (see OSCILLATORY AND BURSTING PROPERTIES OF NEURONS). Both models have been implemented (Linares-Barranco et al., 1991; Patel and DeWeerth, 1997) (Figure 1F). The latter is used to model the intersegmental coordination of the lamprey.

In addition to the passive properties of the lipid membrane, the biological neuronal membrane contains active ionic channels. In the silicon neurons of Douglas and Mahowald (Mahowald and Douglas, 1991), the compartments are populated by modular subcircuits, each of which emulates a particular ionic conductance (see ION CHANNELS: KEYS TO NEURONAL SPECIALIZATION). The dynamics of these circuits is qualitatively similar to the Hodgkin-Huxley mechanism without implementing their specific equations. The various modules that have been designed and tested so far emulate, for example, the sodium and potassium spike currents, persistent sodium current, various calcium currents, calcium-dependent potassium current, potassium A current, nonspecific leak current, and an exogenous (electrode) current source. The prototypical circuits are modified in various ways to emulate the particular properties of a desired ion conductance. For example, some conductances are sensitive to calcium concentration rather than membrane voltage and require a separate voltage variable representing free calcium concentration. Synaptic conductances (see below) are sensitive to ligand concentrations, and these circuits require a voltage variable representing neurotransmitter concentration. This array of ionic conductances, with their different time constants, gives rise to state-dependent dynamics within the compartments. These circuits can be composed to approximate the electrophysiological behavior of various neurons, for example pyramidal cells (Figure 1E). The somatic conductances have been extended to permit the neuron to adapt its discharge sensitivity curve to the statistics of its input current (Shin and Koch, 1999) (Figure 1G; see also ACTIVITY-DEPENDENT REGULATION OF NEURONAL CONDUCTANCES).

Even more detailed models have been fabricated by Dupeyron and colleagues. They used Bi-CMOS technology to implement the detailed differential equations of the Hodgkin-Huxley formalism. Their goal is to build a silicon nerve cell that can interact directly and exactly with real neurons investigated during experiments (Dupeyron et al., 1996)



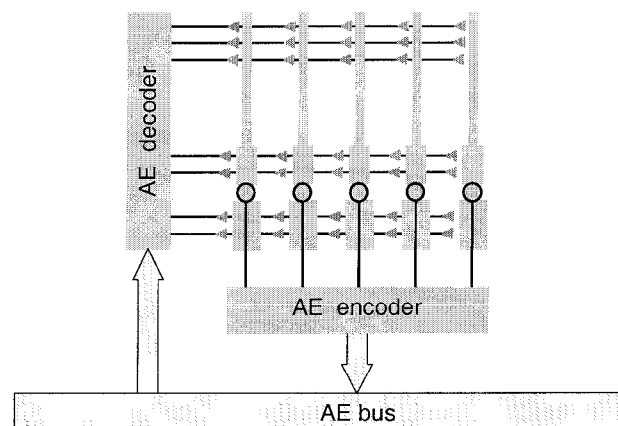
the advantage of continuous communication between neurons. It is appropriate when the neuron output is encoded as a continuous analog variable. An alternative approach is to use the high speed of electrical signals in metal wires to multiplex slow neuronal signals.

Action potential representations of neuronal output are compatible with multiplexing because the output of the neuron is active only during the action potential. Furthermore, the action potential is a digital amplitude signal that can be robustly transmitted between chips. Digital amplitude signals are robust to noise and interchip variability and have been used to advantage in VLSI neural networks (Murray and Tarassenko, 1994). Event-based digital data-encoding methods, such as the address-event representation (AER) (Mahowald, 1994), virtual wires (Elias and Northmore, 1999), and that used by Vittoz et al. (Mortara, Vittoz, and Venier, 1995), broadcast action potential events occurring in neurons onto a common data bus. Many silicon neurons can share the same bus because switching times in CMOS and on the bus are much faster than the switching times of neurons. Events generated by silicon neurons can be broadcast and removed from a data bus at frequen-

cies of more than a megahertz. Therefore, more than 1,000 address-events could be transmitted in the time it takes one neuron to complete a single action potential. Multiplexing strategies are most effective if, as in their biological counterparts, only a small fraction of the silicon neurons embedded in a network are active at any time.

Event-based digital encoding methods facilitate network reconfigurability. These digital multiplexing schemes work by placing on the common communications bus the identity (a digital address) of the neuron generating an action potential. In some implementations, the bus broadcasts this source address to all synapses, which decode the addresses (Figure 2A). In this way those synapses that should be "connected" to the source neuron detect that it has generated an action potential, and they initiate a synaptic input on the dendrite to which they are attached. In other implementations (Deiss, Douglas, and Whatley, 1999), the so-called address-event is translated from a postsynaptic bus to a presynaptic bus through a programmable lookup table that maps the addresses of source neurons to (lists of) destination synapse addresses. The topology of the network is defined by the mapping of source neurons to

a.



b.

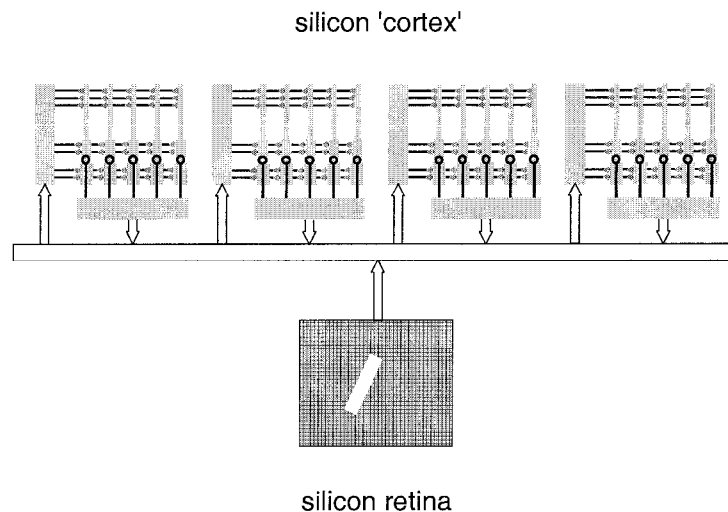


Figure 2. Connections between silicon neurons based on address-event representation. This figure shows an implementation in which source addresses are broadcast to all synapses, and the synapses decode the addresses. *A*, Multineuron chip attached to an address-event (AE) bus. Action potentials generated by neurons (gray broomsticks) are detected by an on-chip AE encoder that broadcasts the binary addresses of the source neurons on the AE bus. AE decoders activate the destination synapses "connected" to the source neurons. *B*, A silicon retina sends address-events to a number of multineuron chips communicating over a common AE bus in a manner similar to that implemented in the system described in Deiss, Douglas, and Whatley (1999).

destination synapses as defined by the content of the lookup table.

Synaptic circuits have been developed that approximate AMPA and NMDA (see NMDA RECEPTORS: SYNAPTIC, CELLULAR, AND NETWORK MODELS) excitatory synapses; and also the potassium-mediated and chloride-mediated (shunting) inhibitory synapses (Douglas and Mahowald, 1995). Synaptic response often depends on the history of its own presynaptic input, on a wide range of time scales. We have developed circuits that emulate the short-time-scale synaptic depression (Figure 1A). On longer scales, a spike-based learning synapse has been implemented that modifies its synaptic weight according to the correlation of pre- and postsynaptic spikes (Häfliger and Mahowald, 1999) (Figure 1B).

Working with Analog Silicon

Custom CMOS circuits are created by an iterative process of design, fabrication, and experiment. In the design phase, the correspondence between elements of the analog circuit and those of the neural system are established, and the variable parameters identified. Computer simulations and mathematical analyses of the proposed circuit subunits are useful at this stage. The electronic circuit design is then transposed into a layout design that expresses the circuit as a sandwich of layers in the silicon chip. The layout is drawn with specialized computer-aided design (CAD) software on a workstation or personal computer. The final layout instructions are used by the silicon foundry to fabricate the chip. The MOSIS service at the University of Southern California's Information Sciences Institute (<http://www.mosis.org/>) accepts layout by electronic mail and returns a fabricated chip in about 10 weeks. Through the MOSIS service, fabrication costs range from approximately \$600 for four pieces of a small 2.2×2.2 mm $2.0\text{-}\mu\text{m}$ feature size chip suitable for prototyping a few neurons to \$15,000 for 20 pieces of a large 9.4×9.7 mm $1.2\text{-}\mu\text{m}$ feature size chip suitable for fabricating a retina or a network of silicon neurons. The European Union's Europractice initiative offers a similar IC prototyping service (<http://www.imec.be/europractice>).

It is important to consider the range of desired behaviors when designing the circuits. Within certain limits, the dynamics of the model neurons can be varied parametrically when the neuron is in use. Often, the behavior of analog circuits can be controlled by the voltages applied to the gates of their various transistors. In the Douglas and Mahowald compartmental model neurons, these parameters determine, for example, the temporal dynamics of activation and inactivation, the voltages at which they occur, and the maximum conductance that can be obtained when fully activated. The effect of changing these parameters is immediate. Thus, the electrophysiological "personality" of the silicon neuron can be switched rapidly, for example, from a regular adapting to a bursting pyramidal cell, as in Figure 1E. Of course, only the parameters that were incorporated into the design at the time of fabrication are available for reconfiguring the performance of the neuron. If additional properties are required, another silicon neuron with different morphology or different types of channels can be fabricated using variations of the basic circuit modules.

To produce compact circuits, it may be necessary to make approximations in the design of the circuit modules. For example, the analog conductances may saturate, and hence these neurons would not perform as real neurons if they were clamped at voltages very far away from the resting potential. These errors are insignificant when the cell is operating in the physiological range because the linear regions of the circuits are arranged in such a way that the deviation from true neuronal behavior is minimized within that range.

The circuits constituting the neuron must be arranged spatially on the surface of the chips. One possibility is to distribute the dendritic compartments of individual neurons across multiple chips

(Van der Spiegel et al., 1994). This approach is useful because the number of compartments, and hence the number of synaptic inputs to a particular neuron, is not limited by the chip boundaries. However, this division requires a method of transmitting accurately between multiple chips the analog voltages and currents at the compartment boundaries. Even if such analog values are accurately conveyed between chips, the values may be misinterpreted because of the interchip variability, the so-called mismatches, inherent in the technology. The alternative approach of Elias is to fabricate entire silicon neurons, and networks of neurons on the same chip. The single-chip solution is appropriate for networks of simple neurons that have only local connectivity mediated by graded synapses, such as the outer layers of the retina (Mahowald, 1994). It is also appropriate for networks of spiking neurons distributed across multiple chips, where only robust, action potential events need be transmitted between chips. However, the number of neurons that can be fabricated on a single chip is limited and depends on the complexity of the neuron. A large development chip has an area of roughly 100 mm^2 . Using $1.2\text{-}\mu\text{m}$ fabrication technology, a retinal chip can accommodate a roughly 100×100 array of simplified photoreceptors together with horizontal cells, bipolar cells, amacrine cells, and retinal ganglion cells (Boahen, 1999). For more comprehensive neurons and more general connectivity, a reasonable target would be a linear array on the order of 100 neurons, each having up to about 10 dendritic compartments. The number of synapses (on the order of 10 to 100) that can be incorporated depends on the size of the synaptic circuits, which in turn depends on their degree of biological realism. There are ways to improve the number of synaptic inputs. One possibility is to map multiple presynaptic inputs into a single postsynaptic circuit. This strategy can raise the effective number of inputs by an order of magnitude. An additional consideration is that as the technology used to implement aVLSI designs continues to evolve toward smaller feature sizes, further improvements in scale can be expected. Nevertheless, an important focus of work must be the development of more compact synapses. Learning synapses that autonomously modify their stored connection strengths in a Hebbian way are another important element of current work (see, e.g., Häfliger and Mahowald, 1999).

Once the chip has been fabricated, its performance is explored using experimental methods similar to those used in a real neurophysiological preparation, except that many more variables can be observed. For example, the response of the analog chip to stimulation is measured in real time with an oscilloscope. Except for the variable parameters included in the design, the circuits cannot be altered after fabrication, and so errors in the specification of the neuron cannot be corrected as easily as in software simulations. Also, care must be taken to plan experiments before the chip is fabricated so that instrumentation circuitry can be included to observe the state of interesting analog variables. The designs of circuits evolve with understanding gained by experiment.

Discussion

A number of groups are currently investigating the properties of analog silicon neurons in networks. At the time of this writing, the scale of these networks ranges from tens of neurons to several thousands of neurons. The next 5 years should see increases in the number of neurons implemented in a single system, owing both to an improvement in the basic fabrication technology and to improved implementations. The optimal degree of biological realism is an open question and is likely to be task dependent. Furthermore, the cost of a computational element ultimately depends on the device physics of the computational primitives, so that more effective methods for performing a computation may eventually become available. The most promising path for the development of these networks is interfacing them to sensors and effectors that can in-

teract dynamically with the real world. Analog VLSI is not the only way to build such systems, but it does have some striking advantages. Analog emulation is inherently parallel, the circuits are extremely compact by comparison with a digital circuit performing an equivalent computation, and the power consumption is often a few orders of magnitude less than their digital equivalents. These properties lend themselves to the construction of small, autonomous neuromorphic systems that can interact directly with the world and so provide a platform for studying animal behaviors by emulation.

In Memoriam. We dedicate this article to our late colleague and co-author, Misha Mahowald.

Road Map: Implementation and Analysis

Background: Single-Cell Models

Related Reading: Analog VLSI Implementations of Neural Networks; Biophysical Mechanisms in Neuronal Modeling; Digital VLSI for Neural Networks; Neuromorphic VLSI Circuits and Systems

References

- Boahen, K. A., 1999, Retinomorph chips that see quadruple images, in *Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-inspired Systems (MicroNeuro '99)*, Los Alamitos, CA: IEEE Computer Society Press, pp. 12–20.
- Deiss, S. R., Douglas, R. J., and Whatley, A. M., 1999, A pulse-coded communications infrastructure for neuromorphic systems, in *Pulsed Neural Networks* (W. Maass and C. M. Bishop, Eds.), Cambridge, MA: MIT Press, chap. 6, pp. 157–178.
- Douglas, R., and Mahowald, M., 1995, A construction set for silicon neurons, in *An Introduction to Neural and Electronic Networks*, 2nd ed. (S. F. Zornetzer, J. L. Davis, C. Lau, and T. McKenna, Eds.), San Diego, CA: Academic Press, chap. 14, pp. 277–296. ♦
- Dupeyron, D., Le Masson, S., Deval, Y., Le Masson, G., and Dom, J.-P., 1996, A BiCMOS implementation of the Hodgkin-Huxley formalism, in *Proceedings of the Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, Los Alamitos, CA: IEEE Computer Society Press, pp. 311–316.
- Elias, J. G., and Northmore, D. P. M., 1999, Building silicon nervous systems with dendritic tree neuromorphs, in *Pulsed Neural Networks* (W. Maass and C. M. Bishop, Eds.), Cambridge, MA: MIT Press, chap. 5, pp. 135–156.
- Häfliger, P., and Mahowald, M., 1999, Spike based normalizing Hebbian learning in an analog VLSI artificial neuron, *Analog Integrated Circuits and Signal Processing* 18, Special issue: *Learning on Silicon*, 2/3:133–139.
- Koch, C., and Segev, I., Eds., 1998, *Methods in Neuronal Modelling: From Ions to Networks*, 2nd ed., Cambridge, MA: MIT Press.
- Linares-Barranco, B., Sanchez-Sinencio, E., Rodriguez-Vazquez, A., and Huertas, J. L., 1991, A CMOS implementation of FitzHugh-Nagumo neuron model, *IEEE J. Solid-State Circuits*, 26:956–965.
- Mahowald, M., 1994, *An Analog VLSI System for Stereoscopic Vision*, Boston, MA: Kluwer Academic. ♦
- Mahowald, M. A., and Douglas, R. J., 1991, A silicon neuron, *Nature*, 354:515–518.
- Mead, C., 1989, *Analog VLSI and Neural Systems*, Reading, MA: Addison-Wesley. ♦
- Mortara, A., Vittoz, E. A., and Venier, P., 1995, A communication scheme for analog VLSI perceptive systems, *IEEE J. Solid-State Circuits*, 30:660–669.
- Murray, A., and Tarassenko, L., 1994, *Analogue Neural VLSI*, London: Chapman and Hall.
- Patel, G. N., and DeWeerth, S. P., 1997, An analogue VLSI Morris-Lecar neuron, *Electron. Lett. IEEE*, 33:997–998.
- Shin, J., and Koch, C., 1999, Dynamic range and sensitivity adaptation in a silicon spiking neuron, *IEEE Trans. Neural Netw.*, 10:1232–1238.
- Van der Spiegel, J., Donham, C., Etienne-Cummings, R., Fernando, S., Mueller, P., and Blackman, D., 1994, Large scale analog neural computer with programmable architecture and programmable time constants for temporal pattern analysis, in *Proceedings of the International Conference on Neural Networks*, vol. III, Orlando, FL: IEEE, pp. 1830–1835.

Simulated Annealing and Boltzmann Machines

Emile H. L. Aarts and Jan H. M. Korst

Introduction

Simulated annealing was introduced in the 1980s by Kirkpatrick, Gelatt, and Vecchi (1983) and Černý (1985) as a local search approach to handle hard combinatorial optimization problems. The approach is based on randomized techniques that are quite similar to the Monte Carlo methods used in statistical physics. The origin of the method lies in the physical annealing process, which is used to find low-energy states of solids. Since its introduction, simulated annealing has been extensively used in a remarkably broad range of applications, including computer engineering, molecular physics, biology, chemistry, and cognitive engineering. Meanwhile, the approach has established a strong position as a successful optimization tool. In this article, we concentrate on what is known as *basic simulated annealing*, thus discarding the wealth of generalized approaches that have been developed. (The interested reader is referred to Aarts and Korst, 1989, 2001.)

Boltzmann machines were introduced by Hinton and Sejnowski (1983) as a class of artificial neural networks that can be viewed as an extension of discrete Hopfield networks (Hopfield, 1982) in two ways. First, they replace the greedy local search dynamics of Hopfield networks with a randomized local search dynamics. Second, they replace the relatively simple Hebbian learning rule with

a more powerful stochastic learning algorithm. Boltzmann machines are randomized neural networks that implement computational features similar to those used in simulated annealing. The resulting neural networks can perform optimization, classification, and learning tasks. Boltzmann machines apply massive parallelism and adaptive adjustment of neural states and interneural connection weights through the self-organization of stochastic computing elements. This class of neural networks is interesting for three reasons. First, it offers a generalized approach to the three basic connectionist issues, i.e., search, representation, and learning (Hinton, 1989). Second, it is supported by a mathematical formalism that facilitates analysis of the network's dynamics and learning properties. Third, it is relatively easy to implement in hardware. Again, we restrict ourselves in this article to a presentation of the basic properties. For a more elaborate treatment we refer to Zwietering and Aarts (1991).

There is a close relation between simulated annealing and Boltzmann machines. Boltzmann machines can be viewed as a massively parallel implementation of simulated annealing, thus offering a means to speed up considerably the slow convergence of sequential simulated annealing implementations. Moreover, simulated annealing is the built-in self-organization technique of a Boltzmann machine, thus providing this class of neural networks with a powerful stochastic learning and retrieval mechanism.

Simulated Annealing

The use of simulated annealing presupposes the definition of a combinatorial optimization problem and a neighborhood. A *combinatorial optimization problem* is a set of problem instances where each *instance* is a pair (\mathcal{S}, f) with \mathcal{S} the set of feasible solutions and $f: \mathcal{S} \rightarrow \mathbb{Z}$ a cost function that assigns a cost value to each solution. The problem is to find a *globally optimal solution*, i.e., an $i^* \in \mathcal{S}$ such that $f(i^*) \leq f(i)$, for all $i \in \mathcal{S}$. Furthermore, $f^* = f(i^*)$ denotes the optimal cost value, and $\mathcal{S}^* = \{i \in \mathcal{S} \mid f(i) = f^*\}$ denotes the set of optimal solutions. A *neighborhood function* is a mapping $\mathcal{N}: \mathcal{S} \rightarrow 2^{\mathcal{S}}$, which defines for each solution $i \in \mathcal{S}$ a set $\mathcal{N}(i) \subseteq \mathcal{S}$ of solutions that are in some sense close to i . The set $\mathcal{N}(i)$ is called the *neighborhood* of solution i , and each $j \in \mathcal{N}(i)$ is called a *neighbor* of i . We shall assume that $i \in \mathcal{N}(i)$ for all $i \in \mathcal{S}$. A solution $\hat{i} \in \mathcal{S}$ is *locally optimal (minimal)* with respect to \mathcal{N} if

$$f(\hat{i}) \leq f(j) \quad \text{for all } j \in \mathcal{N}(\hat{i})$$

The set of locally optimal solutions is denoted by \mathcal{S}^* .

Roughly speaking, simulated annealing starts with an initial solution in \mathcal{S} and then continually tries to find better solutions by searching neighborhoods and applying a stochastic acceptance criterion. This is laid out schematically in Figure 1. The procedure INITIALIZE selects a start solution from \mathcal{S} , GENERATE selects a solution from the neighborhood of the current solution, and STOP evaluates a stop criterion that determines termination of the algorithm.

Simulated annealing continually selects a neighbor of a current solution and compares the difference in cost between these solutions to a threshold. If the cost difference is within the threshold, the neighbor replaces the current solution. Otherwise, the search continues with the current solution. The sequence $(t_k \mid k = 0, 1, 2, \dots)$ denotes the thresholds where t_k is used at iteration k of the algorithm and is given by a random variable with expected value $\mathbb{E}(t_k) = c_k \in \mathbb{R}^+$, $k = 0, 1, 2, \dots$. The thresholds t_k follow a probability distribution function F_{c_k} over \mathbb{R}^+ . Simulated annealing uses randomized thresholds with values between zero and infinity, and the probability of a threshold t_k being at most $y \in \mathbb{R}^+$ is given by $\mathbb{P}_{c_k}\{t_k \leq y\} = F_{c_k}(y)$. This implies that each neighboring solution can be chosen with a finite probability to replace the current solution.

The basic simulated annealing version of Kirkpatrick et al. (1983) and Černý (1985) takes for F_{c_k} the negative exponential distribution with parameter $1/c_k$. This choice is identical to the following *acceptance criterion*. For any two solutions $i, j \in \mathcal{S}$, the probability of accepting j from i at the k th iteration is given by

$$\mathbb{P}_{c_k}\{\text{accept } j\} = \begin{cases} 1 & \text{if } f(j) \leq f(i) \\ \exp\left(\frac{f(i) - f(j)}{c_k}\right) & \text{if } f(j) > f(i) \end{cases} \quad (1)$$

The parameter c_k is used in the simulated annealing algorithm as a *control parameter*, and it plays an important role in the conver-

```

procedure SIMULATED_ANNEALING;
begin
  INITIALIZE ( $i_{\text{start}}$ );
   $i := i_{\text{start}}$ ;
   $k := 0$ ;
  repeat
    GENERATE ( $j$  from  $\mathcal{N}(i)$ );
    if  $f(j) - f(i) < t_k$  then  $i := j$ ;
     $k := k + 1$ ;
  until STOP;
end

```

Figure 1. Pseudocode of the basic simulated annealing algorithm.

gence of the algorithm. A characteristic feature of simulated annealing is that, besides accepting improvements in cost, it also accepts, to a limited extent, deteriorations in cost. Initially, at large values of c , large deteriorations are accepted; as c decreases, only smaller deteriorations are accepted, and finally, as the value of c approaches 0, no deteriorations at all are accepted. Arbitrarily large deteriorations are accepted with positive probability; for these deteriorations, however, the acceptance probability is small.

The Relation with Physics

The origin of simulated annealing and the choice of the acceptance criterion can be found in the physical annealing process. In *condensed matter physics*, *annealing* is a thermal process for obtaining low-energy states of a solid in a *heat bath*. It consists of the following two steps: first, the temperature of the heat bath is increased to a high value, at which the solid melts; second, the temperature is carefully decreased until the particles of the melted solid arrange themselves in the ground state of the solid. In the liquid phase, all particles of the solid arrange themselves randomly. In the ground state, the particles are arranged in a highly structured lattice, and the energy of the system is minimal.

The physical annealing process can be modeled successfully by computer simulation methods based on *Monte Carlo techniques* such as first proposed by Metropolis et al. (1953), who gave a simple algorithm for simulating the evolution of a solid in a heat bath to *thermal equilibrium*. The analogy between the annealing of a physical many-particle system and the application of simulated annealing to a combinatorial optimization problem is obvious: solutions in a combinatorial optimization problem are equivalent to states of the physical system; the cost of a solution is equivalent to the energy of a state, and transitions to neighbors are equivalent to state changes. The control parameter plays the role of the temperature.

Asymptotic Convergence

Simulated annealing can be viewed as a sampling process whose outcomes are neighboring solutions. This class of processes can be mathematically modeled using finite Markov chains. Let \mathcal{O} denote a set of possible outcomes of a sampling process. A *Markov chain* is a sequence of *trials* satisfying the *Markov property*, which states that the probability of the outcome of a given trial depends only on the outcome of the previous trial. In the case of simulated annealing, a trial corresponds to a transition, and the set of outcomes is given by the finite set of solutions.

The convergence of simulated annealing follows from the *stationarity* property of Markov chains and can be formulated as follows. Let (\mathcal{S}, f) be an instance of a combinatorial optimization problem and \mathcal{N} a neighborhood function that is defined in such a way that any two solutions in \mathcal{S} can be reached from each other in a finite number of steps. Then the Markov chain associated with basic simulated annealing has a stationary distribution $q(c)$, whose components denote the probability of finding a solution $i \in \mathcal{S}$ after a large number of trials at control parameter value c . The components are given by

$$q_i(c) = \frac{|\mathcal{N}(i)| \exp(-f(i)/c)}{\sum_{j \in \mathcal{S}} |\mathcal{N}(j)| \exp(-f(j)/c)} \quad \text{for all } i \in \mathcal{S} \quad (2)$$

Furthermore,

$$q_i^* \stackrel{\text{def}}{=} \lim_{c \downarrow 0} q_i(c) = \begin{cases} \frac{1}{|\mathcal{S}^*|} & \text{if } i \in \mathcal{S}^* \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where \mathcal{S}^* denotes the set of optimal solutions.

Cooling Schedules

The theoretical results just presented imply convergence to optimality after an infinite number of trials. A finite-time implementation of the algorithm, which, as a consequence of the above, can no longer guarantee finding an optimal solution, may result in a much faster execution of the algorithm without giving in too much on the solution quality. A *finite-time* implementation of simulated annealing is obtained by generating a sequence of homogeneous Markov chains of finite length at descending values of the control parameter (Aarts and Korst, 1989). For this, a set of parameters must be specified that governs the convergence of the algorithm. This set of parameters is referred to as a *cooling schedule*. It specifies

- an *initial value* of the control parameter,
- a *decrement function* for lowering the value of the control parameter,
- a *final value* of the control parameter (implicitly), specified by a *stop criterion*, and
- a *finite length* of each homogeneous Markov chain.

Typical cooling schedules in simulated annealing start at sufficiently large initial values of the control parameter, thus allowing acceptance of virtually all proposed transitions. Next, the decrement function and the Markov chain lengths are chosen such that at the end of each individual Markov chain, the probability distribution of the solutions is close to the stationary distribution, which is referred to as *quasi-equilibrium*. Since at large values of c , the probability distribution of the solutions equals the stationary distribution by definition (cf. Equation 2), one may expect that the cooling schedule enables the probability distribution to “closely follow” the stationary distributions, so that, as $c \downarrow 0$, the probability distribution is close to q^* , the uniform distribution on the set of optimal solutions given by Equation 3. It is intuitively clear that large decrements in c require longer Markov chains in order to restore quasi-equilibrium at the next value of the control parameter. Thus, there is a trade-off between large decrements of the control parameter and small Markov chain lengths. Usually, one chooses small decrements in c to avoid extremely long chains, but alternatively, one could use large values for the Markov chain length in order to be able to make large decrements in c .

The search for adequate cooling schedules has been the subject of many studies over the past years. The interested reader is referred to Aarts and Korst (2001).

Issues from Practice

During its 20 years of existence, simulated annealing has been applied to a large variety of problems, ranging from practical, real-life problems to theoretical test problems. VLSI design, atomic and molecular physics, and picture processing are the three problem areas in which simulated annealing is probably most frequently applied. The set of theoretical test problems includes almost all of the well-known problems in discrete mathematics and operations research, such as coding, graph coloring, graph partitioning, and sequencing and scheduling problems (see Aarts and Lenstra, 1997). General overviews of applications of simulated annealing are given by Aarts and Korst (1989) and Collins, Eglese, and Golden (1988).

Broadly speaking, after 20 years of practical experience, it is widely accepted that simulated annealing can find good solutions for a wide variety of problems, but often at the cost of substantial running times. As a result, the true merits of the algorithm become obvious in industrial problem settings, where running times are of

little or no concern. As an example, we mention design problems, since in those cases one is primarily interested in finding high-quality solutions, whereas design time often plays only a minor role. A well-known successful simulated annealing area in this respect is VLSI design.

Boltzmann Machines

A Boltzmann machine is a neural network consisting of a set \mathcal{U} of two-state neurons that are interconnected by a set \mathcal{C} of weighted symmetric connections. A neuron $u \in \mathcal{U}$ can be in one of two states: either it is firing, corresponding to state 1, or it is not firing, corresponding to state -0 . A *configuration* k is a $|\mathcal{U}|$ -dimensional vector that describes the global state of the neural network. The state of an individual neuron u in configuration k is given by $k(u)$. The set of all configurations is denoted by \mathcal{R} . A connection $\{u, v\} \in \mathcal{C}$ is *activated* in a given configuration k if both u and v have state 1, i.e., if $k(u) \cdot k(v) = 1$. The set of connections may contain self-connections whose role is similar to that of thresholds in classical neural networks, i.e., $\{\{u, u\} \mid u \in \mathcal{U}\} \subset \mathcal{C}$. A weight $w_{u,v} \in \mathbb{R}$ is associated with the connection between neurons u and v . By definition, $w_{u,v} = w_{v,u}$ for each pair $u, v \in \mathcal{U}$. The weight is a quantitative measure of the *desirability* that $\{u, v\}$ be activated. If $w_{u,v} > 0$, it is desirable that $\{u, v\}$ be activated; if $w_{u,v} < 0$, it is undesirable. Connections with a positive (negative) weight are called *excitatory* (*inhibitory*). The *energy function* $E : \mathcal{R} \rightarrow \mathbb{R}$ assigns to each configuration k a real number, called the *energy*, which equals the negated sum of the weights of the activated connections, i.e.,

$$E(k) = - \sum_{\{u,v\} \in \mathcal{C}} w_{u,v} k(u) k(v) \quad (4)$$

Generally speaking, the energy is low if many excitatory connections are activated and is high if many inhibitory connections are activated. The energy is a global measure indicating to what extent the neurons in the network have reached a consensus about their individual states, subject to the individual weights. Since the weights impose local constraints, these networks are also often called *constraint satisfaction networks* (Hinton and Sejnowski, 1983). The basic idea of implementing local constraints as connection weights in networks dates back to the work of Moussouris (1974).

Network Dynamics

Self-organization in a Boltzmann machine is achieved by allowing neurons to change their states from 0 to 1 or the reverse. Let the network be in configuration k . Then a *state change* of neuron u results in a configuration l , with $l(u) = 1 - k(u)$ and $l(v) = k(v)$ for each $v \neq u$. Furthermore, let \mathcal{C}_u denote the set of connections incident with neuron u , excluding $\{u, u\}$. Then the *difference in energy* $\Delta E_k(u) = E(l) - E(k)$ induced by a state change of neuron u in configuration k is given by

$$\Delta E_k(u) = (2k(u) - 1) \left(w_{u,u} + \sum_{\{u,v\} \in \mathcal{C}_u} w_{u,v} k(v) \right) \quad (5)$$

The effect on the energy, resulting from a state change of neuron u , is completely determined by the states of its adjacent neurons and the corresponding connection weights. Consequently, each neuron can locally evaluate its state change, since no global calculations are required.

In a Boltzmann machine, the response of an individual state change of neuron u to its adjacent neurons in a configuration k is a stochastic function that is given by

$$\mathbb{P}_c\{\text{accept a state change of neuron } u \mid k\} = \frac{1}{1 + \exp(-\Delta E_k(u)/c)} \quad (6)$$

where $\Delta E_k(u)$ is given by Equation 5 and $c \in \mathbb{R}^+$ again denotes the control parameter.

State changes in a Boltzmann machine are governed by simulated annealing. For the sake of presentation, we distinguish between two models, *sequential Boltzmann machines* and *parallel Boltzmann machines*.

Sequential Boltzmann machines. In a sequential Boltzmann machine, neurons may change their states only one at a time. The resulting iterative procedure can be described as a sequence of Markov chains where each chain consists of a sequence of trials and the outcome of a given trial depends probabilistically only on the outcome of the previous trial. A *trial* consists of two steps: given a configuration k , first a neighboring configuration k_u is generated, determined by a neuron $u \in \mathcal{U}$ that proposes a state change. Second, whether or not k_u is accepted is evaluated. If it is accepted, the outcome of the trial is k_u ; otherwise it is k . If the probability of accepting a state change is given by the response function of Equation 6, we obtain the following result, which is analogous to the results of Equations 2 and 3. Given a sequential Boltzmann machine with a response function given by Equation 6, then the following two statements hold:

1. The probability $q(c)$ of obtaining a configuration k after a sufficiently large number of trials carried out at a fixed value of c is given by

$$q_k(c) = \frac{\exp(C_k/c)}{\sum_{i \in \mathcal{K}} \exp(C_i/c)} \quad (7)$$

2. For $c \downarrow 0$, Equation 7 reduces to a uniform distribution over the set of configurations with minimum energy.

Parallel Boltzmann machines. To model parallelism in a Boltzmann machine, we distinguish between synchronous and asynchronous parallelism. In *synchronous parallelism*, sets of state changes are scheduled in successive trials, where each trial consists of a number of individual state changes. During each trial, a neuron is allowed to propose a state change exactly once. Synchronous parallelism requires a global clocking scheme to control the synchronization. For an extensive treatment of synchronously parallel Boltzmann machines and related work, see Little and Shaw (1978), Peretto (1984), and Zwietering and Aarts (1991). Here, we briefly summarize the most important results. The main result is a conjecture, which states that under certain mild conditions a stationary distribution different from Equation 7 is attained, which converges as c approaches 0 to a distribution over the set of configurations for which the so-called *extended consensus* is maximal, where the extended consensus is a modified energy function. A proof of this conjecture is an open problem. However, for two special cases, correctness can be proved. These are the cases of *limited parallelism*, where neurons may change their states simultaneously only if they are not adjacent, and *full parallelism*, where in each trial all neurons may change their states simultaneously.

In *asynchronous parallelism*, state changes are evaluated concurrently and independently. Units generate state changes and accept or reject them on the basis of information that is not necessarily up to date, since the states of adjacent neurons may have changed in the meantime. Asynchronous parallelism does not require a global clocking scheme, which is of advantage in hardware implementations. However, this type of parallelism cannot be mod-

eled by Markov chains but requires a completely different approach, and so far little progress has been made in this direction. A brief discussion on the subject can be found in Aarts and Korst (1989).

Combinatorial Optimization and Classification

The ability of a Boltzmann machine to obtain low-energy configurations can be used to handle combinatorial optimization (CO) and classification problems.

Combinatorial optimization. A Boltzmann machine can be used to solve CO problems by defining a correspondence between the configurations of the Boltzmann machine and the solutions of the CO problem in such a way that the cost function of the CO problem is transformed into the energy function associated with the Boltzmann machine. In general, this can be done by formulating the CO problem as a 0–1 integer programming problem. The values of the 0–1 variables correspond to the states of the neurons. The cost function and the constraints that go with the CO problem are implemented by choosing the appropriate connections and their weights. In this way, minimizing the energy in the Boltzmann machine is equivalent to solving the corresponding CO problem. More specifically, it is often possible to construct a Boltzmann machine such that the following properties hold:

1. Each locally minimal configuration of the Boltzmann machine corresponds to a feasible solution.
2. The lower the energy of the corresponding configuration, the better the cost of the corresponding feasible solution.

These properties imply that a feasible solution can be obtained and that configurations with near-minimal values of the energy function correspond to near-optimal solutions of the CO problem. This feature enables Boltzmann machines to be used for approximation purposes, as we have demonstrated for several well-known problems, including the traveling salesman problem, graph coloring, and independent set (Aarts and Korst, 1989).

Classification. A classification problem can be formalized as a pair $(\mathcal{O}, \mathcal{S})$, where \mathcal{O} denotes a set of *objects* and \mathcal{S} a collection of disjoint subsets $\mathcal{S}_1, \dots, \mathcal{S}_l$ that partition \mathcal{O} . The problem is to determine for a given object $o \in \mathcal{O}$ the subset $\mathcal{S}_j \subset \mathcal{S}$ to which it belongs. In practice, the set of objects is usually very large, and providing an explicit description of each subset is impracticable. The subsets are therefore often implicitly described by specifying a number of “typical” examples for each subset. To use a Boltzmann machine for solving classification problems, the set of neurons is subdivided into three disjoint subsets, \mathcal{U}_i , \mathcal{U}_h , and \mathcal{U}_o , denoting the sets of *input*, *hidden*, and *output neurons*, respectively. The states of the input neurons are fixed by some *input pattern*. This pattern is a coded representation of an object $o \in \mathcal{O}$ that is to be classified. The remaining neurons then adjust their states to minimize the energy, subject to the fixed states of the input neurons. After minimization of the energy, the states of the output neurons represent the subset \mathcal{S}_j to which o is thought to belong. In this way, a Boltzmann machine can implement a given input-output function.

To implement a given input-output function on the input and output neurons, hidden neurons are usually required in addition. The minimum number of hidden neurons required to solve a given classification problem strongly depends on the intrinsic complexity of the input-output function that is to be implemented (Aarts and Korst, 1989). For classification problems, choosing appropriate connection weights is often difficult, since different items may give rise to conflicting connection weights. However, a Boltzmann ma-

chine can often acquire a given input-output function by learning from examples, as shown in the next section.

Learning

Learning in a Boltzmann machine takes place by examples that fix the states of the *environmental neurons*, i.e., the input and output neurons. The hidden neurons are used to construct an internal representation that captures the regularities of the examples fixing the states of the environmental neurons. The learning algorithm we discuss starts by setting all connection weights equal to zero. Next, a sequence of learning cycles is completed, each consisting of two phases. In the first phase, or the *fixed phase*, examples of a given input-output function successively fix the states of the environmental neurons, and for each example the Boltzmann machine is equilibrated using the current set of connection weights. In the second phase, the *free-running phase*, all neurons are free to adjust their state, and again the Boltzmann machine is equilibrated. In both phases, we assume the Boltzmann machine to use limited parallelism. At the end of each learning cycle the connection weights are modified using statistical information obtained from the two phases. This process is continued until the average change, over a number of learning cycles, of the connection weights approaches zero. If the learning is successfully completed, then the Boltzmann machine is able to complete a partial example, i.e., a situation where only a subset of the environmental neurons is fixed, by minimizing the energy. In this way a Boltzmann machine not only can reproduce given examples but also is often capable of classifying correctly objects that in some sense resemble the objects that are used during learning.

Extension of the network structure. The set of environmental neurons is denoted by $\mathcal{U}_{io} = \mathcal{U}_i \cup \mathcal{U}_o$. An *environmental configuration* l is determined by the states of the neurons $u \in \mathcal{U}_{io}$. The state of an environmental neuron u in an environmental configuration l is denoted by $q_l(u)$. \mathcal{Q} denotes the set of all environmental configurations. With each environmental configuration l , a subspace \mathcal{Q}_l can be associated given by

$$\mathcal{Q}_l = \{k \in \mathcal{R} \mid \forall u \in \mathcal{U}_{io} : r_k(u) = q_l(u)\}$$

which consists of all configurations for which the states of the environmental neurons are given by l .

The *learning set* \mathcal{T} consists of the environmental configurations that can be used to fix the environmental neurons during learning. Clearly, $\mathcal{T} \subseteq \mathcal{Q}$.

Toward a learning algorithm. Here we discuss the basic elements of the Boltzmann machine learning algorithm proposed by Ackley, Hinton, and Sejnowski (1985). The objective of the learning algorithm is to modify the connection weights such that a Boltzmann machine in a free-running phase tends, with a large probability, to be in those environmental configurations that belong to the learning set. To this end we introduce two probability distributions, d and d' , defined over the set of environmental configurations. d_l is the probability of obtaining an environmental configuration l in a fixed phase. d'_l is the probability of obtaining an environmental configuration l in a free-running phase. The probability distribution d is determined by the environmental configurations in the learning set and by the frequency at which they are used to fix the environmental neurons. d_l is large if l belongs to the learning set, and l is frequently used to fix the environmental neurons. The probability distribution d' depends on the connection weights and, as is pointed out below, d' can be defined by using the stationary distribution of Equation 7. The objective of the learning algorithm can be formulated as follows: *modify the connection weights of the Boltzmann machine such that d' is close to d* . If $d \approx d'$, then the Boltz-

mann machine can determine for a given input the corresponding output. More generally, if a subset of the environmental neurons is fixed, then the Boltzmann machine can determine the most probable corresponding environmental configuration.

An information-theoretic measure of the distance between the two probability distributions d and d' is the divergence G (Kullback, 1959), given by

$$G = \sum_{l \in \mathcal{Q}} d_l \ln \frac{d_l}{d'_l} \quad (8)$$

It can be shown that $G = 0$ if and only if $d_l = d'_l$ for all $l \in \mathcal{Q}$, and $G > 0$ otherwise. For further properties of G , we refer to Aarts and Korst (1989). The objective of the learning algorithm can be rephrased as: *minimize G by modifying the connection weights*.

Before we describe how G is minimized, we re-address the network dynamics. Using Equation 7, we know that after equilibration in a free-running phase, the Boltzmann machine tends to be in low-energy configurations. If equilibrium is achieved, d'_l is given by

$$d'_l(c) = \sum_{k \in \mathcal{Q}_l} q_k(c)$$

where $q_k(c)$ are the components of the stationary distribution given by Equation 7. The partial derivative of G with respect to $w_{u,v}$ can be written as

$$\frac{\partial G}{\partial w_{u,v}} = \frac{p'_{u,v} - p_{u,v}}{c} \quad (9)$$

where $p_{u,v}$ and $p'_{u,v}$ denote the probabilities of connection $\{u, v\}$ being activated at equilibrium in the fixed and the free-running phases, respectively (see, e.g., Aarts and Korst, 1989). To minimize G , it suffices to collect statistics on $p_{u,v}$ and $p'_{u,v}$ and to iteratively change the connection weight proportionally to the difference between the probabilities, i.e.,

$$w_{u,v} := w_{u,v} + \eta(p'_{u,v} - p_{u,v}) \quad (10)$$

where $\eta \in \mathbb{R}^+$ is called the *learning parameter*.

Informally speaking, the learning algorithm consists of the following steps:

1. Set all connection weights to zero.
2. Complete a number of learning cycles, until $d \approx d'$, where each learning cycle consists of the following steps:
 - a. Fixed phase: use a number of examples to fix the environmental neurons, equilibrate for each example, and collect statistics on $p_{u,v}$.
 - b. Free-running phase: equilibrate and collect statistics on $p'_{u,v}$.
 - c. Modify connection weights.

By definition, environmental configurations l not included in the learning set \mathcal{T} have a corresponding probability $d_l = 0$. Consequently, G is not properly defined for these environmental configurations unless $d'_l = 0$, which can only be realized by infinitely large connection weights, owing to the stochastic nature of the Boltzmann machine optimization. To avoid this problem, a *noise ratio* is introduced that allows the environmental neurons in the fixed phase to change their states with a certain probability. Furthermore, the use of noise suppresses the unlimited growth of connection weights. This was found to be essential for obtaining good results.

The average case performance and time complexity of the learning algorithm depend on the following issues:

1. The parameters of the learning algorithm, viz., the cooling schedule, the learning parameter, noise ratio, and the number of examples per cycle.

2. The ratio between the environmental and the hidden neurons.
3. The way the neurons are connected.
4. The intrinsic difficulty of a given classification problem.

Many studies in the literature have considered the influence of these issues on the performance of the learning algorithm. Most of these studies were based on experimental evaluations. The overall conclusion is that learning in a Boltzmann machine is rather slow. This is essentially due to the time needed to equilibrate and collect statistics. Several approaches that might speed up the learning algorithm have been proposed, including hardware acceleration, multivalued neurons, and higher-order choices for the energy function. For further reading refer to Aarts and Korst (1989).

Road Map: Learning in Artificial Networks

Background: Computing with Attractors

Related Reading: Statistical Mechanics of Neural Networks

References

- Aarts, E. H. L., and Korst, J. H. M., 1989, *Simulated Annealing and Boltzmann Machines*, New York: Wiley. ♦
- Aarts, E. H. L., and Korst, J. H. M., 2001, Selected topics in simulated annealing, in *Essays and Surveys in Metaheuristics* (C. Ribeiro and P. Hansen, Eds.), Boston: Kluwer, pp. 1–37.
- Aarts, E. H. L., and Lenstra, J. K., Eds., 1997, *Local Search in Combinatorial Optimization*, New York: Wiley.
- Ackley, D. H., Hinton, G. E., and Sejnowski, T. J., 1985, A learning algorithm for Boltzmann machines, *Cognit. Sci.*, 9:147–169. ♦
- Černý, V., 1985, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *J. Optimiz. Theory Appl.*, 45:41–51.
- Collins, N. E., Eglese, R. W., and Golden, B. L., 1988, Simulated annealing: An annotated bibliography, *Am. J. Math. Manag. Sci.*, 8:209–307.
- Hinton, G. E., 1989, Connectionist learning procedures, *Artif. Intell.*, 40:185–234.
- Hinton, G. E., and Sejnowski, T. J., 1983, Optimal perceptual inference, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 448–453.
- Hopfield, J. J., 1982, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, 79:2554–2558. ♦
- Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P., 1983, Optimization by simulated annealing, *Science*, 220:671–680.
- Kullback, S., 1959, *Information Theory and Statistics*, New York: Wiley. ♦
- Little, W. A., and Shaw, G. L., 1978, Analytic study of the memory storage capability of a neural network, *Math. Biosci.*, 39:281–290.
- Metropolis, M., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., 1953, Equation of state calculations by fast computing machines, *J. Chem. Phys.*, 21:1087–1092. ♦
- Moussouris, J., 1974, Gibbs and Markov random systems with constraints, *J. Statis. Phys.*, 10:11–33.
- Peretto, P., 1984, Collective properties of neural networks: A statistical physics approach, *Biol. Cybern.*, 50:51–62.
- Zwietering, P. J., and Aarts, E. H. L., 1991, Parallel Boltzmann machines: A mathematical model, *J. Parallel Distrib. Comput.*, 13:65–75. ♦

Single-Cell Models

Christof Koch, Chun-Hui Mo, and William Softky

Most of the roughly ten billion neurons in the human cerebral cortex are tiny, membrane-bound bags of saltwater, shaped like trees (including roots). Each is surrounded by more salt water and by other neurons, many of which it is connected to. The vast majority of them communicate by means of brief, all-or-none pulses (called spikes, or action potentials), each lasting a bit less than a millisecond.

Researchers often want to distill the complex shape and behavior of a real neuron into a simpler model, either to guide a neurobiological experiment or to construct a functional network. But choosing which neuronal properties to keep and which to ignore is heavily influenced by how one interprets the pulses. In particular, it is common to reduce the train of action potentials issued by a neuron with an equivalent point process, but beyond that much is in dispute.

Most theories assume that information is carried in the average rate of pulses over a time much longer than a typical pulse-width, so that the occurrence times of particular pulses simply represent jitter in an averaged analog signal. A neural model in such a theory might be a mathematical function that produces a real-valued output from its many real-valued inputs; that function could be linear or nonlinear, static or adaptive, and might be instantiated in analog silicon circuits or in digital software.

However, a few theories assume that each single neural pulse carries reliable, precisely timed information. A neural model in such a theory fires only on the exact coincidence of several input pulses, and quickly “forgets” when it last fired, so that it is always ready to fire on another coincidence. Whether real neurons operate in this regime or in the slower average-rate regime awaits further neurobiological experiment. Both types of codes and single-neuron

models have special features and advantages; understanding the models touches issues of bandwidth, nonlinearity, and the fundamental precision and function of single nerve cells. Of course, it is well possible that the nervous system uses both, depending on place, time, and circumstance.

Formal Models

It is easiest to understand and analyze the models that are the least like real neurons. Virtually all such models share two features in common. First, each model neuron combines many inputs, both “excitatory” and “inhibitory,” into a single output. And each neuron has at least one internal state variable (conceptually corresponding to the cell’s average membrane potential), which increases monotonically with the total amount of excitatory input and decreases with inhibitory input. Thus, the neuron is constrained to “adding up” (in a rough sense) its positive and negative inputs, and *cannot* independently assign an arbitrary value to each of its possible input combinations.

McCulloch-Pitts Model

The McCulloch-Pitts model neuron (see Koch, 1999, for details), more than fifty years old, has many progeny present in digital circuits, in the form of logic gates. The explicit assumptions of this model are that each binary “pulse” represents a logical statement (i.e., *true* or *false*), and that each neuron performs an exact, noise-free, synchronous computation on its input pulses.

If any one of the model neuron’s inhibitory inputs is active, the output is shut off, or inactive. Otherwise, all the active excitatory

inputs x_i are multiplied by their *synaptic weights* w_i and then added up. However, only if this activity level exceeds a preset *threshold* θ is the output active:

$$Y = \begin{cases} 1 & \text{if } \sum_i w_i x_i > \theta \text{ and no inhibition} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

McCulloch and Pitts showed that a large enough number of such units, with weights and connections set properly and operating synchronously, could in principle perform any possible computation (Arbib, 1987).

An even simpler model is the *linear neuron*, which simply adds up its inputs and delivers the sum as output, with no thresholds or other nonlinearities. The neuron's real-valued output is the sum of its inputs x_i , weighted by real-valued coefficients w_i :

$$Y = \sum_i w_i x_i \quad (2)$$

Networks of linear neurons can be treated analytically, using well-established matrix methods. Unlike the spikes and rates from real neurons, outputs from the linear model can become negative or arbitrarily large.

Perceptron Model

Rosenblatt's Perceptron model (see Koch, 1999, for details) is formally similar to the McCulloch-Pitts model, having synchronous inputs and producing outputs between zero and one. But the perceptron creates a real-valued (not binary) output, representing the *average firing rate* of the cell. As with the linear model, the internal variable V of a perceptron is the weighted sum of its inputs:

$$V = \sum_i w_i x_i \quad (3)$$

A *threshold* or *bias* θ is subtracted from V and is then passed through a continuous and monotonically increasing function g :

$$Y = g(V - \theta) \quad (4)$$

The nonlinear function g is *sigmoidal*: it asymptotes zero as $V \ll \theta$ and saturates at 1 for $V \gg \theta$ (Figure 1C). This function mimics the empirical relationship between the cell's input current and its firing rate in several ways: the output is non-negative, it is very small below θ , it monotonically increases with input, and the firing rate has an upper bound.

Hopfield Neurons

In Hopfield's binary model (see Koch, 1999, for details), the output of neuron i is the step function of V_i and the threshold θ ,

$$Y_i = \begin{cases} 0 & \text{if } V_i < \theta \\ 1 & \text{if } V_i > \theta \end{cases} \quad (5)$$

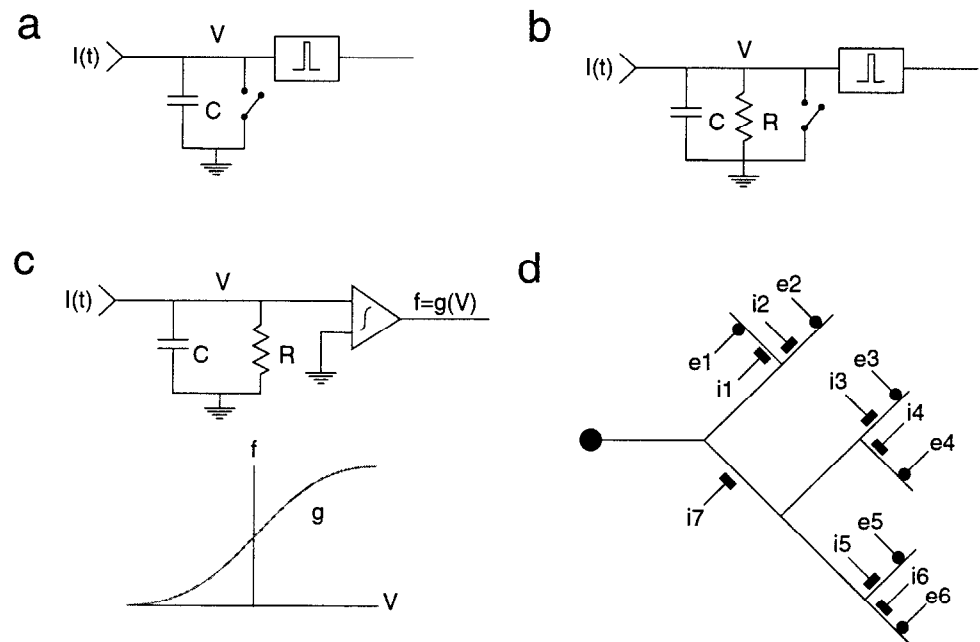
Unlike the McCulloch-Pitts or the Perceptron model, each Hopfield neuron updates its state at a *random* time, independently of any other neurons.

Both Hopfield's binary and continuous-valued models (Figure 1C) are similar to perceptrons in isolation, but can act as associative memories in highly interconnected networks.

Polynomial Neurons

The appeal of models like those discussed previously is that only a very simple function of the inputs—a weighted sum—is neces-

Figure 1. Simple neuronal models expressed in the commonly used electrical circuit idiom. In (A–C), the entire neuron is reduced to a single spatial compartment (point-model). The summed synaptic input is described by a net current $I(t)$. A, An integrate-and-fire unit. If the voltage V exceeds a fixed threshold, a unit pulse is generated, and all charge on the capacitance is removed by resetting V to zero. The output of this and the leaky integrate-and-fire model (B, in which charge leaks away with a time constant given by the product of the capacitance C and resistance R) is a series of asynchronous spikes (C). In a rate neuron, these discrete pulses are replaced by a continuous output rate whose amplitude is proportional to the inverse of the average interspike interval. The monotonically increasing relationship between V and the output rate $f = g(v)$ can be thought of as the discharge function of a population of spiking cells. D, Nonlinear, saturating interactions can be mediated in a passive dendritic tree by synapses that increase the postsynaptic conductance. The interaction between excitation (circles) and inhibition of the shunting type (elongated boxes) is of the AND-NOT type and is specific in space and in time. For instance, the inhibitory synapse i_7 vetoes excitation e_3 or e_6 but has only a negligible effect on e_1 . (Modified from Koch and Segev, 2000).



sary for them to work. However, such a sum cannot distinguish among the individual contributions to it. For the neuron to respond strongly to correlations among particular input pairs or groups, one must include multiplicative terms, and then sum over the products. Such a *sigma-pi* (Σ = sum, Π = product) neuron computes its internal state as the sum of contributions from a set of monomials

$$V = a_1 + b_1x_1 + b_2x_2 + c_1x_1^2 + c_2x_1x_2 + \cdots \quad (6)$$

This state variable can then be passed through the usual nonlinear function g . It is clear that such “neurons” are computationally richer than linear or threshold units—just one can implement parity, exclusive-or, or *lookup table* functions. Furthermore, such models also better represent the operations of real neurons containing highly branched dendrites with voltage-dependent membrane conductances (Mel, 1994).

Biophysical Models

Although many crucial properties of real neurons remain unknown, biophysical models incorporate some known properties of neural tissue. Like real, spiking neurons, these models produce spikes rather than continuous-valued outputs.

Integrate-and-Fire Models

This family of single-cell models, about a century old, divides the membrane behavior conceptually into two distinct and discontinuous regimes: a prolonged period of linear “integration” (adding up of inputs), and a sudden “firing” (Figure 1A and B). The “integrate-and-fire” model relaxes the requirement that a single set of continuous differential equations describe the cell’s two very different regimes. The cell voltage starts from zero, rising or lowering according to the synaptic input. When the voltage reaches a certain threshold θ , the cell instantly fires an output pulse and resets the voltage. After a refractory period—a brief “dead time” during which the cell cannot fire at all—the unit is ready to fire again.

The simplest type of integrator model is a leak-free capacitance (Figure 1A). With steady, DC input current, it acts like a relaxation oscillator or a current-to-frequency converter, producing regular output pulses at a rate depending on the input current.

If the input instead arrives in brief excitatory pulses (e.g., spikes from other cells), so that N pulses are necessary to reach threshold, then this model acts like a divide-by- N counter, firing on every N th input pulse. Its output rate depends on the *average* of the overall firing rate of the inputs. For small, random input pulses, the output firing will be fairly regular as the input randomness is averaged out. The fact that such a neuron can smooth out input noise is one of its great advantages.

The addition of a leak resistance in parallel to the capacitance makes a *leaky integrate-and-fire neuron*, which will only fire if the excitatory input is strong enough to overcome the leak (Figure 1B). The time constant $\tau = RC$ divides the model’s operation into two qualitatively distinct regimes: temporal integration and fluctuation-detection. When τ is larger than the mean time between output spikes, the leak is insignificant and the model temporally integrates its input (Figure 2). When τ is much smaller than the average output interval, then production of a spike depends not on the *average* input but on input *fluctuations*: only a rare fluctuation will bring the voltage above threshold. Here, the output represents a precisely timed threshold-crossing computation with a binary output—in this regime the neuron is neither linear nor analog.

Integrate-and-fire models can account for remarkable many aspects of the behavior of real neurons (Koch, 1999).

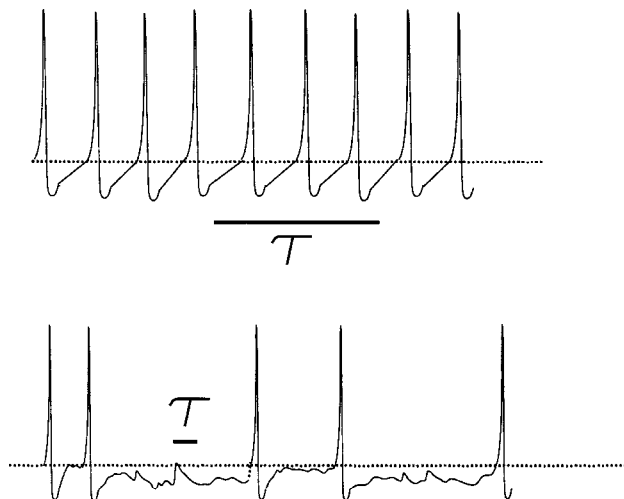


Figure 2. A schematic of the two distinct operating regimes of an integrate-and-fire model. The top trace shows how current input ramps up the neuron’s internal “voltage” to produce regular spikes. The input current determines the *average slope* of the voltage between spikes. This can only occur if the interval between spikes is less than the neuron’s time constant, so that the spiking frequency reliably reflects the average input current. A strikingly different situation is shown in the lower trace, where a relatively shorter time constant causes the neuron to *forget* when its last spike occurred. Now the input current determines the *average voltage* between spikes, which is nearly constant and below spiking threshold. Here a spike is only generated by a brief fluctuation so that each spike can be interpreted as the distinct binary output of a fast, multiplicative computation, reporting the precise coincidence of several contributing inputs.

The Hodgkin-Huxley Model of Squid Axon

Biophysically much more accurate single-cell models that can account for the very complex, nonstationary behavior of real neurons are lumped under the catch-all name of Hodgkin-Huxley models, since they use a mathematical formalism little different from that set out fifty years ago in the ground-breaking work of Hodgkin and Huxley (1952) on the electrodynamics of the squid axon. The dynamics are modeled by numerous coupled, nonlinear differential equations that describe the behavior of continuous currents that depend in a nonlinear manner on the membrane potential.

In its quiescent state, the inside of a typical neuron has a negative voltage (relative to the extracellular fluid). The cell membrane acts like a capacitor. The electrical charge carriers (various species of ions such as Na^+ , K^+ , Cl^- , and Ca^{2+}) pass through special pores or *ionic channels* embedded in the membrane (Hille, 1992). Although each individual channel is either open or closed (in a partially stochastic manner), the current through many channels in parallel is well approximated by continuous, deterministic equations, much as the laws of electrical current describe averages over many electrons. That is, the continuous, macroscopic, and deterministic membrane currents derive from binary, microscopic, and stochastic ionic channels.

In the simplest case, the channels’ collective behavior resembles an ohmic (or passive) resistor across the membrane. The combination of the resistance and the capacitance creates a membrane time-constant τ , which is typically between 5 and 50 ms.

Other ion channels are nonlinear: their conductance *depends on* voltage. For instance, an action potential occurs when the membrane potential becomes depolarized enough that voltage-controlled sodium channels open, initiating the fast positive-feedback event of a spike. One spike lasts between one-half and

one millisecond, and is followed by a few milliseconds of *refractory period* during which it is difficult or impossible to fire another spike.

This process was described by Hodgkin and Huxley in 1952 in one of the most successful of all models in neurobiology: a four-dimensional set of coupled, nonlinear, partial differential equations. Those equations describe the initiation and propagation of action potentials in axons well enough that they are often treated as “gospel truth,” although they are technically imperfect phenomenological fits rather than expressions derived from first principles.

Simplified versions—the so-called *FitzHugh-Nagumo* and van der Pol oscillator equations (see Koch, 1999, for details)—yield qualitatively the same kind of subthreshold behavior and limit-cycle oscillations as the original Hodgkin-Huxley equations, but their reduced parameters are more difficult to be interpreted biophysically.

Although the Hodgkin-Huxley methodology is powerful, it suffers from the drawback that it requires detailed knowledge of a myriad of parameters. It is frequently difficult to properly constrain all of these degrees of freedom.

Modified Single-Point Models

More realistic neural models must account for Nature’s rich array of nonlinear currents and additional internal variables, only some of which are understood.

Internal Variables. Most of the simple models outlined previously have only a single internal variable: the membrane potential. An additional variable, such as the concentration of free, intracellular calcium, can give a wider array of functions. Calcium concentration roughly represents a running average of the cell’s recent activity. This temporal averaging—along with the ability of calcium ions to remain trapped near synapses—makes it a candidate for modulating synaptic strength. Calcium also participates in “adaptation,” a hysteresis effect in which the calcium accumulated from past spikes makes it more difficult to fire new ones.

Additional Ionic Currents. Most neurons typically contain a dozen or more *nonlinear* ion channels, whose conductance depend on the cell voltage. There are slow positive feedback currents, such as calcium and persistent sodium currents, which tend to amplify large voltage excursions. There are also negative-feedback currents like those found for potassium, which tend to hyperpolarize the cell, acting like a kind of active inhibition or adaptation. These “active” currents can strongly influence a cell’s response to input, its input, but their strengths in real cells are often unknown. For a discussion of the computational significance of all of these, see Koch (1999).

Compartmental Models

The *single-point* models discussed previously assume that neurons do not have any significant spatial extent. However, most real neurons have intricate dendritic trees (where the synaptic input arrives), as well as an axon and its branches (where the output spike is carried away). The unique shapes of those dendrites and axons can distinguish between various cell classes.

Axons are usually thought of as delay lines without any significant information processing ability (see AXONAL MODELING). Dendrites, however, display a much richer repertoire of information processing operations. The simplest “passive” dendrite model, pioneered by Rall (1989), is a single capacitive, resistive cable. Its voltage is characterized by the *cable equation*, which has two main parameters: τ is the membrane time constant, and λ the electronic space constant, is a characteristic distance over which a steady-state voltage attenuates. Signals always attenuate and temporally

smooth as they spread from their sources in such a dendritic cable (Rall, 1989). For a review of compartmental methods, see the handbook by Koch and Segev (1998).

Because dendrites with branches are not as easily analyzed, modelers resort to discretizing the cable equation (like decomposing the dendritic tree into hundreds of simple electrical compartments connected by Ohmic resistors, as in Figure 3. Note the right branches of the dendritic tree contain arbitrary active membrane component.).

Computation with Passive Dendrites

Low-pass filtering. As predicted by linear cable theory, one important computational function that passive dendrites can perform is low-pass filtering. This operation removes high temporal frequencies from voltage response to input signal and generates voltage attenuation and temporal delay.

Synaptic Saturation. Real synapses—whether slow or fast, inhibitory or excitatory, passive or voltage gated—are best modeled as *conductance* in the cell membrane in series with driving potentials. Synaptic input is not a constant current or voltage source, but rather changes the electronic properties of postsynaptic membrane. Postsynaptic potential (PSP) saturates when synaptic input becomes stronger. As a result, the PSP for two synaptic inputs arriving together is usually smaller than the sum of PSPs of two independent input. When many synapses are simultaneously active, their increased conductance further attenuates distant input (by reducing λ) and makes the cell sensitive to fluctuations at a faster timescale (by reducing τ).

Shunting Inhibition. If the driving potential of the inhibitory synapse is close to the resting potential of the cell, it is called *shunting inhibition*. Such an inhibition can veto excitation locally *only* if the inhibition lies on the path between the location of the excitatory synapse and the cell body. Shunting inhibition enables local nonlinear computation at the dendrite level and endows single neurons with more powerful computational abilities (Figure 1D).

Computation with Active Dendrites

Experiments have revealed that dendrites contain not only passive membranes but also nonlinear “active” ones (Stuart, Spruston, and Häusser, 1999). These active membranes have voltage-dependent sodium, calcium, or potassium channels that readily perform local nonlinear computation (Figure 3). Given these active channels, action potential can backpropagate to remote dendrites from soma, providing a communication mechanism between soma and dendrites. Such a mechanism can be critical to the Hebbian learning rule-based synaptic plasticity.

Synaptic Input Amplification. Input impedances at distal arbors are usually higher than those of proximal large dendrites. Therefore, distal inputs generate higher local excitatory postsynaptic potential (EPSP), which will be amplified more by local active channels. This amplification can offset the attenuation along the cable so that synapses near and far from the cell body are equally effective. Recent evidence (Magee and Cook, 2000) shows synaptic conductance changes increasing as one moves along the apical dendrite away from the soma in CA1 pyramidal neurons. Such a progressive increase in conductance seems to be the mechanism rendering EPSP size insensitive to input location.

Coincidence Detection. Biophysically plausible proposals for coincidence detection exploit fast sodium action potential generation in spines and distal basal dendrites to achieve sub-millisecond res-

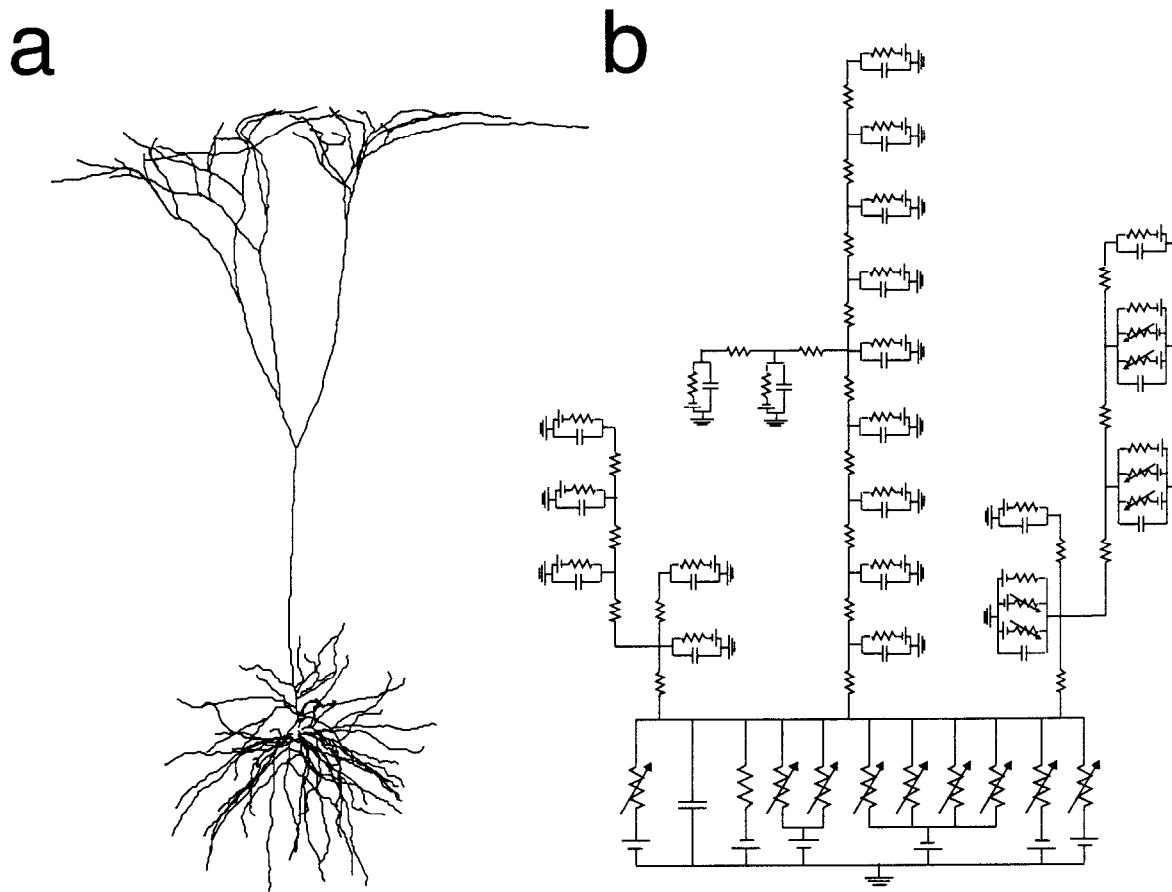


Figure 3. The most realistic form of single-neuron model numerically simulates the electrical properties of a branched cell membrane. First, the observed shape of the cell (A) is approximated by a collection of connected cylinders of the appropriate length and diameter. Then, each cylinder is simulated as a single electrical unit composed of an axial resistance and

membrane capacitances and resistances (B). Although it is computationally intensive, this numerical approach can treat the whole variety of cell shapes and nonlinear electrical properties, which are ignored by the traditional single-compartment models. Here the right branches of the dendritic tree contains voltage-dependent membrane conductance.

olution (Softky, 1994). Sakmann's group (Larkum, Zhu, and Sakmann, 1999) demonstrated dendritic coincidence detection at the 10 ms level in layer V pyramidal neurons. If an excitatory input at dendrites coincides with somatic-triggered backpropagation action potential, a powerful calcium spike may be triggered at the dendrite level. This long-lasting (10 ms or longer) calcium spike in turn triggers a burst of sodium potential at the soma.

Multiplication Operation. Multiplication is one of the simplest but most important nonlinear operations that serve as the basis for many computation models, such as those for optomotor response of insects and motion perception in primates. Nonlinear interaction by voltage-dependent channels at dendrites might provide a mechanism to implement multiplication.

Synaptic Clustering. The distribution of synapse on active dendrites can be very important because spatially close, simultaneous synaptic inputs will be amplified more by active channels. The difference in amplification is large enough to generate orientation selectivity in V1 cell models (Mel, Ruderman, and Archie, 1998).

Discussion

After several decades of research, we understand only some of the most fundamental functions of nerve cells.

Learning

The most remarkable feature of the nervous system is its ability to change its internal structure in response to previous input—that is, to learn. Theoretical aspects of learning are covered elsewhere (see the three road maps on learning in Part II of the *Handbook*). However, at the single-cell level, there are many issues left to resolve.

Strengthening or weakening a single synapse weight, w_{ij} , seems to involve complex events, including calcium accumulation on both sides of the connection. However, there are several different types of increase and decrease, occurring under different circumstances and lasting from seconds to days. These mechanisms are not yet well understood, especially in the cerebral cortex. Markram and colleagues (1997) controlled the relative timing of presynaptic and postsynaptic action potentials in a pair of excitatory-coupled neurons, measuring its effect on the strength of synaptic coupling between the two cells. If the presynaptic spike preceded the postsynaptic one by as little as 10 ms, synaptic strength increased (long-term potentiation, LTP). Conversely, if the postsynaptic spike preceded the presynaptic one, synaptic coupling decreased (long-term depression, LTD). This gives rise to powerful, temporally asymmetric Hebbian learning rules.

A potent (but poorly understood) type of learning could occur as axonal branches and connections “die off,” while other branches form elsewhere and connect to other cells. If such structural plas-

ticity participates in learning, then at least as much information could be stored in the existence/nonexistence of the synaptic connections as in their current-pulse amplitudes or “strengths.”

The Dendrites

Although most input to cortical cells arrives on an intricately branched dendritic tree, we do not understand how the tree processes the input. Whether those tiny dendrites smooth out brief, localized input fluctuations—or instead amplify them—depends on the type of nonlinear membrane properties to be found there. Nonlinear dendrites can make a single “neuron” function as a large collection of distinct, multiplicative subunits.

Slow Analog versus Fast Digital

There are two distinct, self-consistent interpretations of single-neuron function, which are diametrically opposed, but there is evidence for both (Figure 2).

In the most popular interpretation, the fundamental computation is like that of a Perceptron or a Hopfield neuron: a slow real-valued output resulting from slowly varying real-valued inputs, in which the timing of single spikes is inconsequential. Indeed, decades of experiments on most parts of the brain have found that only average rates—and not individual spike times—correlate with simple stimuli.

This corresponds to an integrate-and-fire neuron gathering many small inputs, smoothing out their irregularities in the dendritic tree, summing the results in the cell body, and producing a firing rate as output. Here the internal variable is the *average current* into the cell body, and the output firing rate can be interpreted as current into the *next* cell (the cell voltage is just a repeating ramp whose phase has no significance). This form of computation is most effective when the fluctuations in current are small, so that the cell fires in response to the mean current.

An alternative is McCulloch and Pitts’ original interpretation of their binary neuron: the “active” state corresponds to a *single spike* rather than to an prolonged firing rate, so that every spike carries some kind of independent message. Although such a model can in principle transmit information much faster than an analog neuron, there are four fundamental criticisms of it: Are binary computations more appropriate than analog ones? Is there any need for such a high bandwidth? Can that high bandwidth be implemented in a realistic cell? And how sensitive is such a cell to the noise that exists in the cortex?

Other neural systems—such as the fly’s visual system—can indeed carry significant information by single spikes. And there is a

need for *some* improved neural bandwidth to solve problems of segmentation and binding. But there is so far no solid evidence that most cortical areas need such fast temporal resolution.

This single-spike regime corresponds to an integrate-and-fire type of cell that fires according to its instantaneous (rather than average) voltage, so it is sensitive to input fluctuations. However, in order for the cell to respond reliably to fluctuations, the average current must not dominate them—otherwise the mean current will ramp up the voltage and fire the cell, fluctuations or not.

Road Maps: Biological Neurons and Synapses; Grounding Models of Neurons

Related Reading: Axonal Modeling; Biophysical Mechanisms in Neuronal Modeling; Ion Channels: Keys to Neuronal Specialization; Perspective on Neuron Model Complexity

References

- Arbib, M., 1987, *Brains, Machines, and Mathematics*, 2nd ed., New York: Springer-Verlag. ♦
- Hille, B., 1992, *Ionic Channels of Excitable Membranes*, 2nd ed., New York: Sinauer. ♦
- Hodgkin, A. L., and Huxley, A. F., 1952, A quantitative description of membrane current and its application to conduction and excitation in nerve, *J. Physiol. (Lond.)*, 117:500–544.
- Koch, C., 1999, *Biophysics of Computation: Information Processing in Single Neurons*, New York: Oxford University Press. ♦
- Koch, C. and Segev, I., Eds., 1998, *Methods in Neuronal Modeling: From Ions to Networks*, 2nd ed., Cambridge, MA: MIT Press.
- Koch, C., and Segev, I., 2000, Single neurons and their role in information processing, *Nat. Neurosci.*, 3:1171–1177.
- Larkum, M. E., Zhu, J. J., and Sakmann, B., 1999, A new cellular mechanism for coupling inputs arriving at different cortical layers, *Nature*, 398:338–341.
- Magee, J. C., and Cook, E. P., 2000, Somatic EPSP amplitude is independent of synapse location in hippocampal pyramidal neurons, *Nat. Neurosci.*, 3:895–903.
- Markram, H., Lubke, J., Frotscher, M., and Sakmann, B., 1997, Regulation of synaptic efficacy by coincidence of postsynaptic Aps and EPSPs, *Science*, 275:213–215.
- Mel, B., 1994, Information processing in dendritic trees, *Neural Computat.*, 6:1031–1085.
- Mel, B., Ruderman, D., and Archie, K., 1998, Translation-invariant orientation tuning in visual “complex” cells could derive from intradendritic computations, *J. Neurosci.*, 18:4325–4334.
- Rall, W., 1989, Cable theory for dendritic neurons, in *Methods in Neuronal Modelling* (C. Koch and I. Segev, Eds), Cambridge, MA: MIT Press, pp. 9–62.
- Softky, W., 1994, Sub-millisecond coincidence detection in active dendritic trees, *Neuroscience*, 58:15–41.
- Stuart, G., Spruston, N., and Häusser, M., Eds., 1999, *Dendrites*, New York: Oxford University Press.

Sleep Oscillations

Alain Destexhe and Terrence J. Sejnowski

Introduction

The brain spontaneously generates complex patterns of neural activity. As the brain falls asleep, the rapid patterns characteristic of aroused states are replaced by low-frequency, synchronized rhythms of neuronal activity. At the same time, electroencephalographic (EEG) recordings shift from low-amplitude, high-frequency rhythms to large-amplitude, slow oscillations. In what follows, we focus primarily on this slow-wave sleep, rather than

rapid eye movement (REM) sleep, whose oscillatory properties resemble those of wakefulness.

The thalamus and cerebral cortex are intimately linked by means of reciprocal projections. The thalamus is the major gateway for the flow of information toward the cerebral cortex and is the first station at which incoming signals can be blocked by synaptic inhibition during sleep. This shift contributes to the transition that the brain undergoes from an aroused state, open to influence from the outside world, to the closed state of sleep. The early stage of

quiescent sleep is associated with EEG spindle waves, which occur at a frequency of 7–14 Hz. As sleep deepens, waves with slower frequencies (0.1–4 Hz) appear on the EEG. This article summarizes what is known about the biophysical mechanisms underlying spindle oscillations.

The dramatic reduction in forebrain responsiveness during sleep, the pervasiveness of these changes, and the discovery of the underlying specific cellular mechanisms, suggest that sleep oscillations are highly orchestrated and highly regulated. Experimental and modeling studies have shown how sleep rhythms emerge from an interaction between the intrinsic firing properties of thalamic and cortical neurons and the networks through which they interact (Steriade, McCormick, and Sejnowski, 1993; Destexhe and Sejnowski, 2001). These advances have raised interesting possibilities regarding the function of sleep.

Biophysical Basis of Sleep Spindle Oscillations

Sleep spindles are characteristic of brain electrical synchronization at sleep onset, an electrographic landmark for the transition from

waking to sleep that is associated with loss of perceptual awareness. Spindle oscillations consist of 7–14 Hz waxing-and-waning field potentials, grouped in sequences that last for 1–3 s and recur once every 3–10 s. Spindle oscillations constitute an interesting and well-constrained problem to investigate by computational models for several reasons. First, these oscillations are generated in the thalamus, which is a well-known structure anatomically, with well-defined connectivity between the different cell types. Second, spindles are remarkably well documented experimentally and have been extensively characterized both *in vivo* and *in vitro* (reviewed in Steriade et al., 1993; Destexhe and Sejnowski, 2001). Third, this oscillation is generated by an interplay of complex cellular properties, such as burst firing, and synaptic interactions via multiple types of postsynaptic receptors (reviewed in Destexhe and Sejnowski, 2001). Computational models are needed to understand this complex interplay, as we summarize here.

The typical electrophysiological features of spindle oscillations are shown in Figure 1A. The two cell types involved, thalamocortical (TC) and thalamic reticular (RE) neurons, oscillate synchronously and display burst discharges according to a mirror

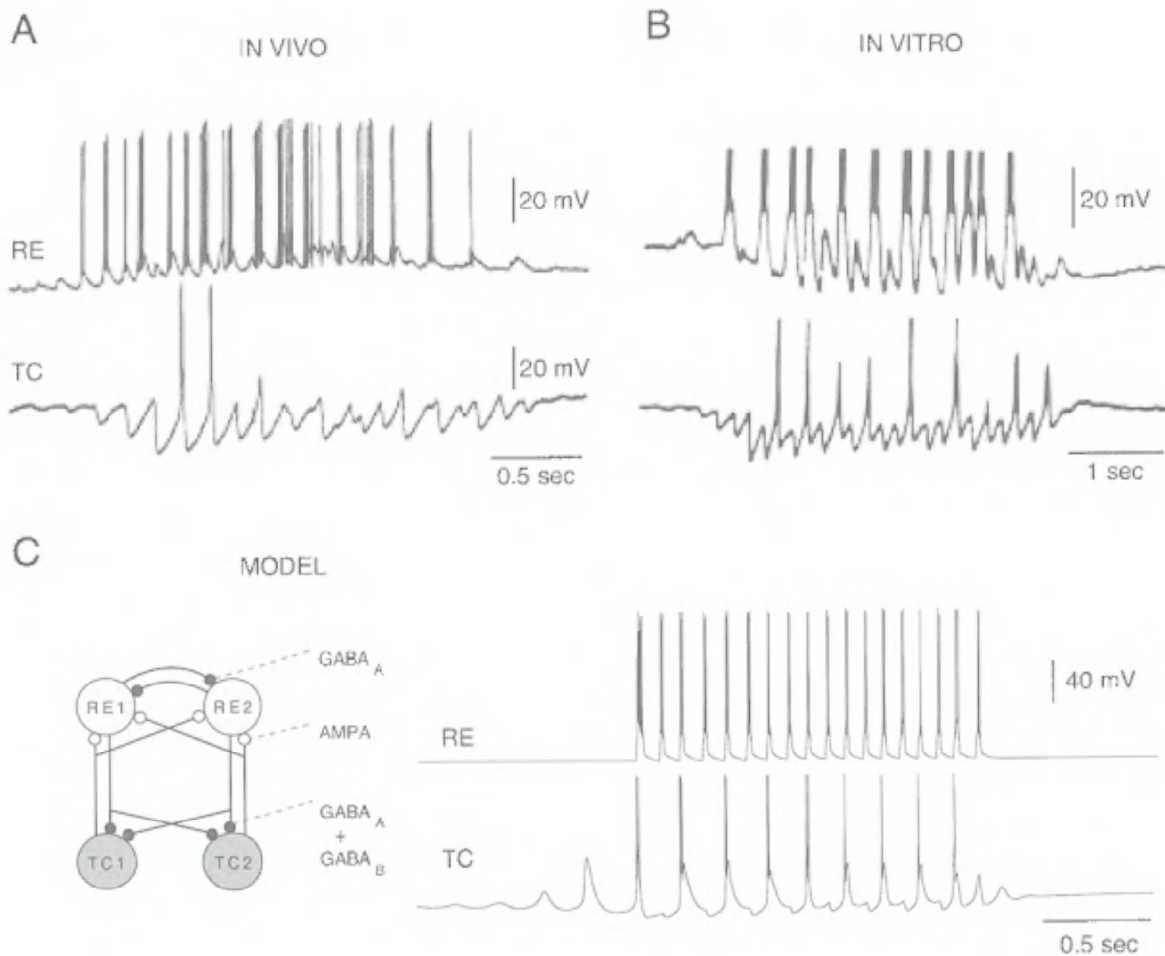


Figure 1. Spindle oscillations in thalamic circuits. *A*, Intracellular recordings of thalamic neurons during spindle oscillations *in vivo* (cats, barbiturate anesthesia; modified from Steriade et al., 1993). *B*, Intracellular features of spindle oscillations in ferret thalamic slices (spikes truncated; modified from Steriade et al., 1993). *C*, Model of spindle oscillations by interacting TC and RE cells. The intrinsic firing properties of each cell type was simulated by Hodgkin-Huxley type models for Na^+ , K^+ and Ca^{2+}

currents, and kinetic models of postsynaptic receptors (AMPA, GABA_A, and GABA_B; see scheme) were used to represent synaptic interactions (modified from Destexhe et al., 1996; see also Destexhe, Mainen, and Sejnowski, this volume). In all three examples, RE cells generated bursts following EPSPs while TC cells generated bursts following IPSPs once every few cycles.