Fetz EE. Dynamic neural network models of sensorimotor behavior.
In: Gardner D, ed. The neurobiology of neural networks.
Cambridge, Mass: MIT Press; 1993:165-190.

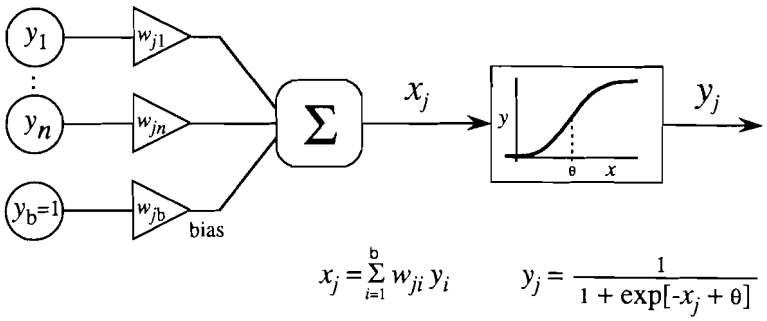
Dynamic Neural Network Models of Sensorimotor Behavior

Eberhard E. Fetz

The goal of systems neurophysiology is to understand the neural mechanisms underlying behavior. Traditional approaches toward this goal have involved anatomical and physiological techniques that reveal the connectivity or activity of specific components of the system. Yet, despite innumerable studies documenting how lesions affect behavior, how stimulation evokes movement, and how single units fire in behaving animals, we still have no consensus about how biological neural networks actually generate movement. The basic reason that the experimental data have not yet converged on a definitive causal explanation of motor behavior is simple: The traditional techniques inevitably provide only limited samples of the system. For example, recordings of single unit activity indicate their response patterns, but nothing about their relation to the rest of the system. What is missing is what artificial neural network models can provide: a method of generating working models of the complete system.

Two general approaches to neural network modeling should be distinguished at the outset. The bottom-up approach involves synthesizing experimental details about cellular mechanisms into a working network model. This approach appeals to physiologists concerned with preserving biological reality, who wish to demonstrate how physiological neurons could function when combined into circuits; however, the bottom-up approach runs into problems dealing with the whole organism, since many parameters of the nervous system are largely unknown and must be arbitrarily assigned. Moreover, modeling the whole system with "realistic" neurons soon becomes prohibitively complex. Alternatively, recent advances in neural network algorithms now make it possible to derive top-down simulations of sensorimotor behavior in networks of appropriately chosen connectivity. The latter techniques provide neural networks that perform specified behaviors, derived entirely on the basis of examples of the behavior itself.

The first examples of such top-down simulations involved feedforward networks trained with backpropagation to transform spatial patterns. For example, feedforward networks have been trained to classify the shape of a surface from shaded images (Lehky and Sejnowski 1990).



Input-output Properties for Sigmoidal Units used in Dynamic Recurrent Networks

Figure 7.1 Each unit j generates a continuous activation y_j as output. Its net input x_j is the weighted sum of activities of its input units y_i and the bias, weighted by synaptic connection strengths w_{ji}. This sum is transformed by the sigmoidal input-output or "squashing" function f to limit the output: $0 < y_j < 1$. The offset θ reduces output to near zero in the absence of input.

Another such network combined the retinal coordinates of a visual stimulus with eye displacement to provide a code for the position of the stimulus in head coordinates (Zipser and Andersen 1988a). Interestingly, the hidden units in such networks often have response properties resembling those seen in biological neurons recorded in animals. The shape-from-shading networks developed hidden units with center-surround receptive fields like visual cortex cells, even though the simulated task did not explicitly require such properties or present stimuli with edges. The reasons that units in network simulations of a task often have such close correspondence with units in biological systems may be more than coincidental; it may be related to basic properties of identification models, as discussed by Zipser (1992).

The supervised training procedures have now been extended beyond feedforward networks, which transform spatial patterns through serial layers, to dynamic recurrent networks, which can deal with spatiotemporal patterns and have feedback connections. Results obtained from top-down simulations of behavior with dynamic recurrent networks are the subject of this chapter.

Three key properties distinguish dynamic recurrent networks from other modeling approaches:

First, the units are *dynamic*, meaning they can exhibit time-varying activity. The activation of each unit is a temporal variable, which can represent the firing rate of neurons, their synaptic potentials or some relevant time-varying sensory or motor parameters. This means that the models can incorporate physiologically recorded activity as activation patterns that the network can be required to generate.

Second, the networks have a *recurrent* architecture, which allows unrestricted connectivity. In addition to feedforward connections between successive layers of units, the networks can have feedback and

cross-connections. Thus, the models can incorporate recurrent anatomical pathways.

Third, the networks that simulate a particular sensorimotor behavior can be derived from examples of the behavior by *gradient descent* methods such as backpropagated error correction. The resulting network 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 sensorimotor behaviors—i.e., that transform spatiotemporal inputs to appropriate outputs—can be derived by various training strategies. These networks usually comprise the units illustrated in figure 7.1, often named "sigmoidal" or "logistic" for the fact that the unit's output y is a sigmoidal function of its input x. The unit's input consists of the summed output activation of all other cells connected to that unit times their synaptic weight; a steady input may also be provided by a bias element:

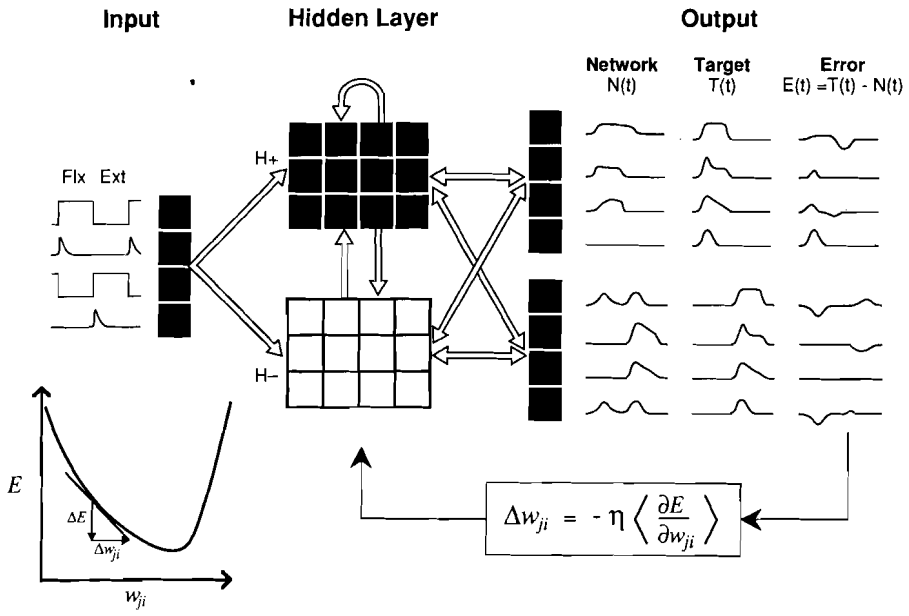
$$x_j = \sum_{i=1}^{b} y_i w_{ji} \tag{7.1}$$

(see also equation 3.1). The sigmoidal input-output function f mimics a biological neuron's property of saturating at maximal rates for high levels of input, and decreasing to zero at low input levels:

$$y_j = f(x_j) = \frac{1}{1 + e^{[-x_j + \theta]}} \,. \tag{7.2}$$

Additional properties of these units are provided in chapter 3; chapter 5 describes not only the units but also the input-output function. The function f often includes an offset term θ to ensure that units generate negligible output in the absence of input.

To illustrate the training procedure, figure 7.2 shows a representative network of such units, with network input and output patterns that simulate the step-tracking task described below. The four input units carry signals representing the changing target locations; these signals are transformed to eight output activation patterns, representing the firing patterns of typical motor units in the muscles of a monkey tracking the target. To train the network, the synaptic weights 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 modifies the weights in such a way as to optimally reduce this error. As shown by the equation, the optimal weight change can be computed from the error when the functions are differentiable. As illustrated by the inset in figure 7.2, this weight change implements a *gradient descent* of the error as a function of the weight. After each weight correction the input patterns are presented again and the new outputs



Typical Network Architecture and Training Procedure Used with Dynamic Recurrent Networks

Figure 7.2 This network was used to simulate the step-tracking task (see figure 7.5 for a specific example). 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 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 the use of the gradient of this function to determine the appropriate weight change.

determined. This process of changing the weights in proportion to the error gradient is repeated iteratively until the network converges on a solution with minimal error.

Several types of algorithms have been used to implement such gradient descent methods for dynamic recurrent networks. Watrous and co-workers derived backpropagation techniques to train dynamic networks to identify acoustic features from speech data (Watrous and Shastri 1986). Their temporal flow model involved a hidden unit layer between input and output that had recurrent connections. Pearlmutter (1989) developed an algorithm for recurrent networks of units carrying continuous temporal signals, and applied this to training a network to generate trajectories in phase space. Williams and Zipser (1989a, 1990) provided a real-time recurrent learning (RTRL) algorithm for training recurrent networks carrying time-varying signals. RTRL updates the weights after each time step, rather than after the end of a sequence, allowing the training to occur in real time, with less storage require-

ments. A useful variant of this algorithm is teacher-forced learning, in which the target activations are substituted for the actual unit outputs during training; this has proven more effective for training networks to generate oscillations. A comparison of various training methods for recurrent networks is presented in Williams and Zipser (1990).

APPLICATIONS

The applications for these dynamic recurrent networks can be classified into three broad categories:

Pattern recognition applications involve identification of spatiotemporal input patterns into discrete categories. A set of input units receiving time-varying signals can be considered to represent a spatiotemporal pattern. Pattern recognition applications include speech recognition (Watrous and Shastri 1986), conditional delay tasks (Williams and Zipser 1989a, b) and recognition of finite state grammars (Smith and Zipser 1989).

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 (Pearlmutter 1989; Williams and Zipser 1989a, b) and simulations of central pattern generators (Rowat and Selverston 1991; Tsung et al. 1990), described below.

Pattern transformation networks convert spatiotemporal input patterns into spatiotemporal outputs. Examples include simulations of the leech withdrawal reflex (Lockery and Sejnowski 1990, 1992), step-target tracking in the primate (Fetz and Shupe 1990; Fetz et al. 1990), the vestibulo-ocular reflex (Anastasio 1991a, b; Arnold and Robinson 1991; Lisberger and Sejnowski 1992a), and short-term memory tasks (Zipser 1990, 1991). Recurrent networks have also been trained to simulate analytical transforms such as integration and differentiation of input signals (Munro et al. 1991) and to emulate special-purpose Turing machines (Williams and Zipser 1989b).

Given the generality and power of dynamic recurrent networks, it seems remarkable that the applications are still sufficiently few to summarize in the scope of this chapter.

Oscillating Networks

Biological systems provide numerous examples of autonomously generated periodic motor activity: locomotion, mastication, respiration, etc. The neural circuitry underlying cyclic periodic movements has been dubbed a central pattern generator (CPG). Additional reviews of CPGs are discussed in chapters 5 and 6. Besides motor activity, oscillatory activity of various frequencies also appears in cortical circuitry during the alpha and gamma waves of neocortex and the theta waves of hippocampus. Networks that generate oscillatory activity have been con-

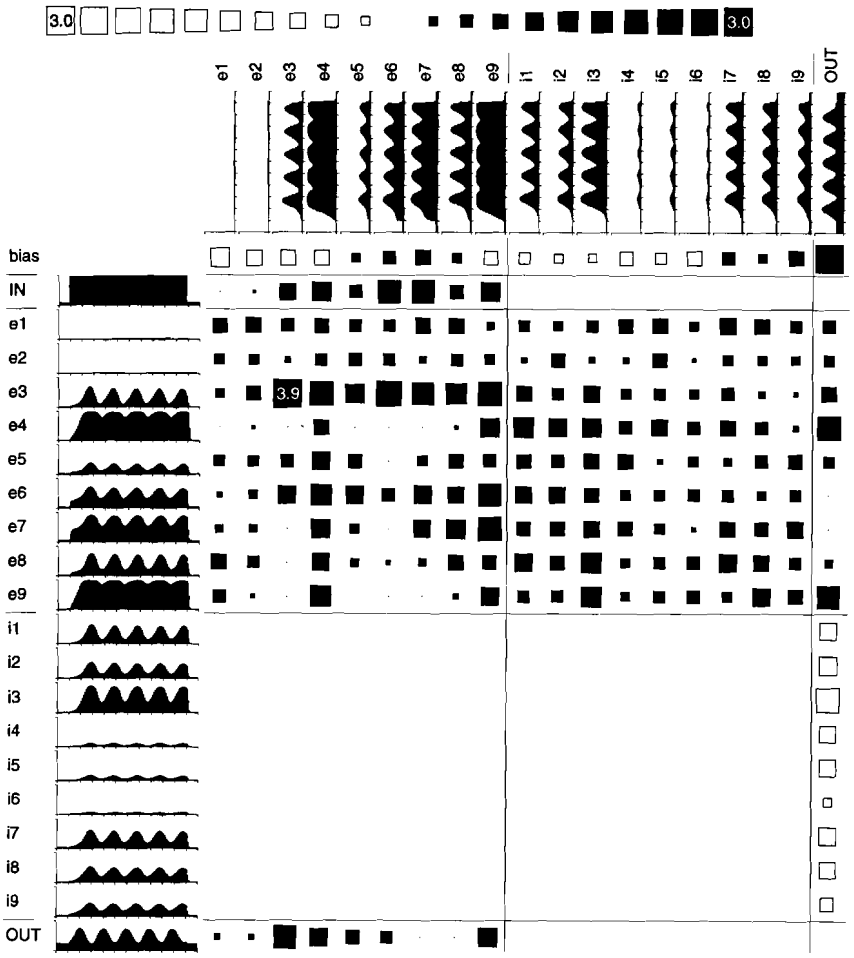
structed with bottom-up strategies (e.g., Buchanan 1992; Grillner et al. 1991), but our interest here is in networks derived by backpropagation techniques.

Williams and Zipser (1989a, b) trained dynamic networks of sigmoidal units to generate oscillatory activity with various specified frequencies. Boolean oscillations between two binary states were readily achieved, but only with teacher-forced training. Dynamic networks could also generate sine-like oscillations with periods of 25 timesteps or less. The smallest circuit that sustained quasi-sinusoidal oscillations consisted of two interconnected sigmoidal units. The frequency of the free-running network was about 10% lower than the trained frequency sustained by the teacher signal. Selverston and co-workers (Tsung et al. 1990) trained pairs of sigmoidal units to oscillate with specific phase shifts. The two units preferred to oscillate with similar (but not identical) phase; training them to exhibit larger phase shifts involved increasing numbers of training cycles. A 15-unit interconnected network without sign constraints could readily be trained to oscillate with equidistant phases (i.e., phase shifts of $2\pi/15$, or 24°).

To simulate the oscillatory activity of neurons in the lobster gastric mill circuit, Tsung et al. (1990) trained a network with the connectivity and sign constraints of known neurons in the mill circuit; this network replicated the correct phase relations of the oscillating interneurons. After 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.

Rowat and Selverston (1991) have developed learning algorithms to train oscillations in dynamic recurrent networks of units with biological properties like gap junctions and membrane currents. They used a generalization of the Williams and Zipser teacher-forced learning procedure to train networks to simulate known oscillatory activations and to observe sign and magnitude constraints on connection weights.

My colleague Larry Shupe and I have also derived autonomous oscillators through dynamic recurrent networks. The input was a gating step that turns network activity on, and the target output was activity oscillating at a prescribed frequency. Figure 7.3 shows a typical recurrent network that generates gated oscillations. The discharge pattern of each unit is shown along the left, next to the row of output weights of that unit. From top to bottom these include the bias (a constant), the gate signal (IN), nine excitatory hidden units (e1–e9), nine inhibitory hidden units (i1–i9) and the oscillating output (OUT). The hidden units and output are shown again at the top, above the column of weights representing the input connections to each unit. The squares in the matrix symbolize the strength of the synaptic connection from the row unit to the column unit. Black squares designate excitatory weights and open squares represent inhibitory weights. (Note that this convention is opposite from that of the Hinton diagrams in chapter 5.) The size of each square

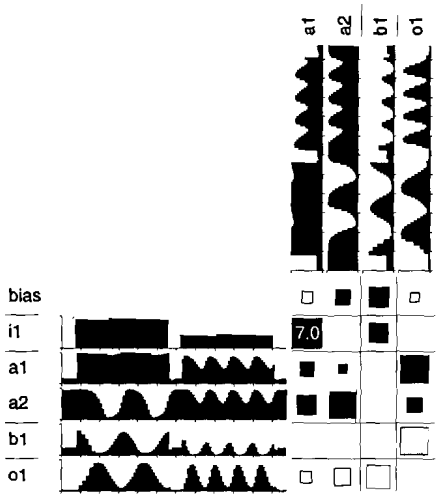


Network Trained to Generate Gated Sinusoidal Oscillation

Figure 7.3 The weight matrix gives the strength of connections from the row to column units (scale at top; excitatory black). The units are represented by name and activation; from top to bottom: the bias (a constant maximal activation of 1, not illustrated), the input gate signal (IN), the excitatory hidden units (e's), inhibitory hidden units (i's) and output (OUT).

is proportional to the strength of the connection, except those that exceed the calibrated scale at the top, which are designated numerically. In this simulation the excitatory hidden units are connected to each other, to the inhibitory group, and reciprocally to the output cell; the inhibitory hidden units connect only to the output. The network was trained without teacher forcing and achieved an oscillatory period of 20 time bins.

The network in figure 7.3 has more units than required to generate the oscillations. Many units have similar activations and connections (e.g., e4 and e9), and other units have negligible activity (e.g., e1 and i6). To simplify the network and elucidate the neural mechanisms, it is



Reduced Network That Generates Oscillations at Different Frequencies, Depending on the Level of Tonic Input

Figure 7.4 After the network was trained on the two illustrated input levels (0.4 and 0.9), testing with intervening levels of input generated oscillations at intervening frequencies.

often useful to reduce the network to its minimal size (LeCun et al. 1990; Mozer and Smolensky, 1989). A network of many hidden units can be reduced by (1) combining units with similar responses and connections into one equivalent unit and (2) eliminating units with negligible activation or weak connections. This can be done manually or by incorporating an automatic weight decay algorithm. Such reduced networks can then be retrained to perform the same operation. Figure 7.4 illustrates a reduced version of a network trained by my co-workers A. Pralle and L. Shupe to generate oscillations at two different frequencies, depending on the level of input. This network was trained to oscillate at a low frequency for an input of 0.9 and a high frequency for an input of 0.4. When presented with input levels between the training levels, this network also generates intervening frequencies. The interactions generating the oscillations can be analyzed by determining how the activation of each unit is determined by its inputs.

These simulations suggest that recurrent networks of sigmoidal units are quite robust in generating oscillatory activity, even to the point of meeting a variety of constraints in phase and period.

Reflex Responses of the Leech

Biological systems provide many examples of spatiotemporal transforms in reflex responses to localized stimuli. The leech withdrawal reflex has been modeled with dynamic recurrent networks by Lockery and Sejnowski (1990; 1992). The network inputs were stimuli applied to dorsal and ventral cutaneous receptors; the output signals were the

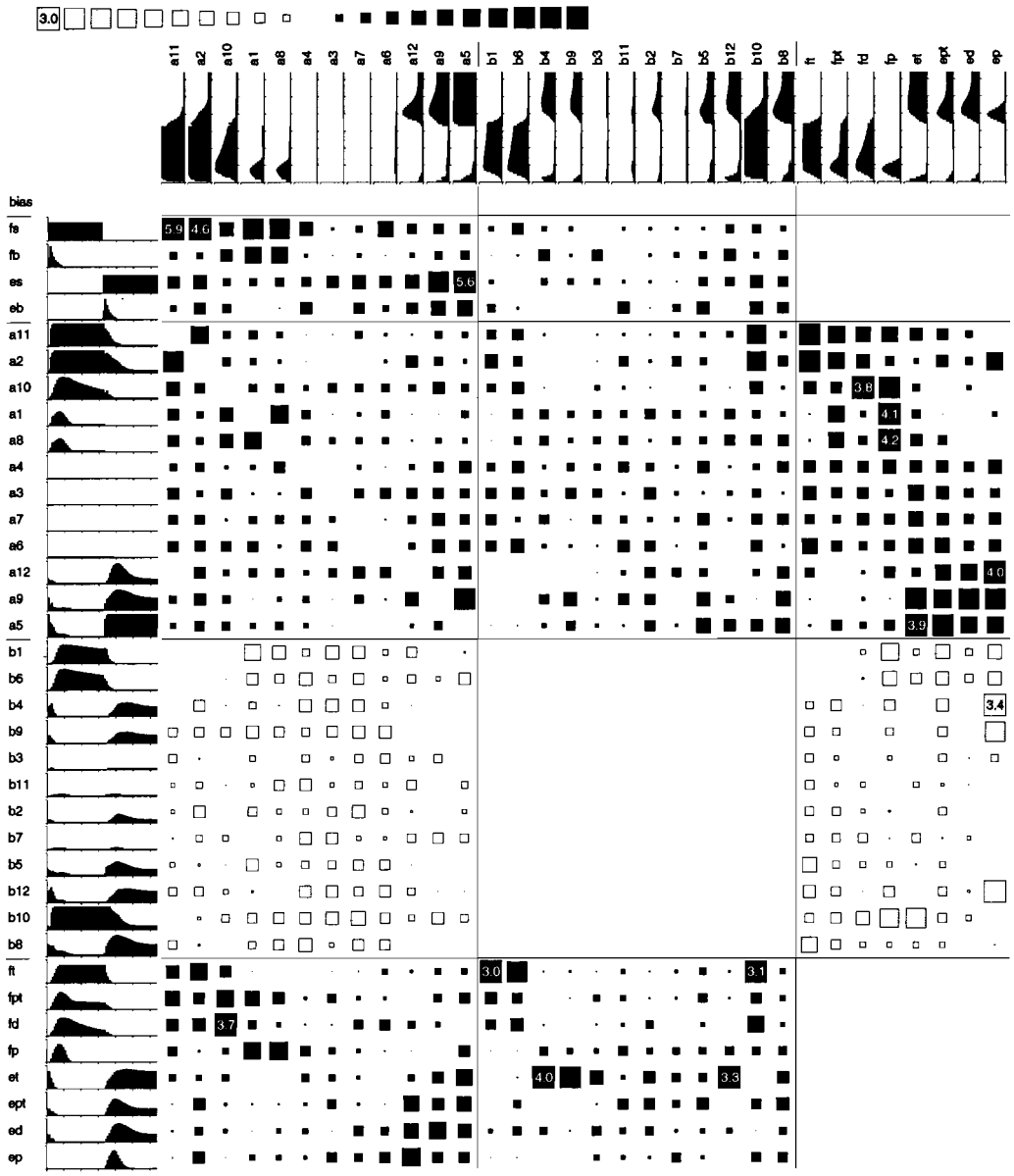
postsynaptic potentials evoked in representative motor neurons. These networks also incorporated electrical synapses and time constants which generated fast and slow components of synaptic potentials. The trained networks implemented distributed reflex pathways, as seen in the leech. Further details are provided in chapter 5.

Primate Step Tracking

The primary interest of my laboratory in dynamic recurrent networks derives from a desire to understand the neural circuitry controlling forelimb muscles of the primate. This will be discussed in some detail to illustrate the types of insights provided by network simulations. In monkeys performing a step-tracking task, previous experiments have documented the physiological discharge patterns and output connections of task-related neurons. The premotoneuronal (PreM) cells which affect muscle activity were identified in behaving monkeys by post-spike facilitation of target muscles in spike-triggered averages of electromyographic recordings. The response patterns of three groups of PreM cells—corticomotoneuronal (CM), rubromotoneuronal (RM), and dorsal root ganglion afferents—have been documented during a simple alternating flexion/extension task designed to relate their activity to changing and sustained force (Fetz et al. 1989; Flament et al. 1992). The discharge patterns observed in PreM cells, as well as in single motor units (MU) of agonist muscles (Palmer and Fetz 1985), fall into specific classes (Fetz et al. 1989). During a ramp-and-hold movement, all three groups include cells that show phasic-tonic discharge. The phasic component is related to the changing force and the tonic component is proportional to the amount of static force exerted. All groups also include tonic cells which show steady discharge throughout the hold period in proportion to the active force. Each group also has cells with unique firing properties. A large proportion of motor units show decrementing discharge, which decreases gradually through the static hold period. The RM population has cells that fire during both flexion and extension, and some cells that are unmodulated with the task and provide a constant bias.

To investigate the possible functional role of all these 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 7.5 shows a neural network that simulates the input-output transformation in the step-tracking task. The monkey sees a step change in target position, alternating between flexion and extension target zones. This is represented by sustained step inputs to the network for both flexion (fs) and extension (es). Since many visual system cells discharge



Dynamic Network Simulating Step-tracking Behavior Documented in Monkeys

Figure 7.5 The network transforms step and transient inputs to firing patterns of motor units at output. Unit activations during a flexion-extension cycle are shown at left and along the top. The rows represent, from top to bottom, the bias (which was eliminated for this simulation), the input signals (fs to eb), the excitatory hidden units (a's), inhibitory hidden units (b's) and the output flexor and extensor motor units (ft to ep). The target output patterns for each group are tonic (ft and et), phasic-tonic (fpt and ept), decrementing (fd and ed) and phasic (fp and ep). Hidden units were sorted in order of contribution to flexion vs. extension outputs. (Reprinted from Fetz and Shupe 1990, with permission.)

transiently, we also provided brief transient input at the onset of each target change (fb and eb). The network transforms these input signals to the observed response patterns of motor units at the output. The four types of motor unit patterns observed experimentally—tonic, phasic-tonic, decrementing, and phasic—are generated for both flexor and extensor movements (ft to ep). The intervening hidden units consist of twelve excitatory and twelve inhibitory neurons (a and b units, respectively).

Figure 7.5 illustrates the activation patterns and the connection matrix of all units after the network produces the eight different output patterns (2000 training iterations were sufficient). The discharge pattern of each unit is shown along the left, next to the row of output weights of that unit, and is shown again at the top of the column of weights representing the input connections to that unit. Self-recurrent connections (corresponding to weights on the diagonal) were excluded. To better visualize the relationships between units, the hidden units were sorted in order of the strength of their contribution to the phasic-tonic output units. The sorting algorithm used the product of activation and weight to the flexion phasic-tonic motor unit minus activation times weight to the extension phasic-tonic unit. Thus, the first hidden unit (a11) makes the largest relative contribution to the flexion phasic-tonic output unit (fpt). This hidden unit developed the strongest weights to the flexion tonic output unit (ft), but was also connected to the other flexor units. Such divergent connections to different motor units, as well as to synergist muscles are representative of CM cells (Fetz et al. 1989).

The activation patterns of the hidden units show several interesting features. The discharge patterns in the hidden units involve some recognizable variants of the output patterns, including tonic, phasic, phasic-tonic, and decrementing patterns. Secondly, although the activation profiles of the target motor units are identical for the flexion and the extension groups, the network solution involves a different assignment of hidden units devoted to each. There are more excitatory and fewer inhibitory hidden unit activations related to flexion than to extension, yet they produce essentially identical output effects. In general, different network simulations with the same architecture and trained on the same task, but starting with different initial weights usually converged on different solutions. (See chapter 3 for a relation of this property to biological variability.)

Another interesting feature of the weight matrix is the preferentially strong connections within the sets of units with sustained and transient activity. Thus, the first two flexor hidden units exhibiting tonic activity (a11 and a2) are strongly interconnected with each other and receive potent input from the flexion step and connect strongly to the tonic output unit. Similarly, the transient input (fb) is most strongly connected to the phasic hidden units (a1 and a8), which are strongly inter-

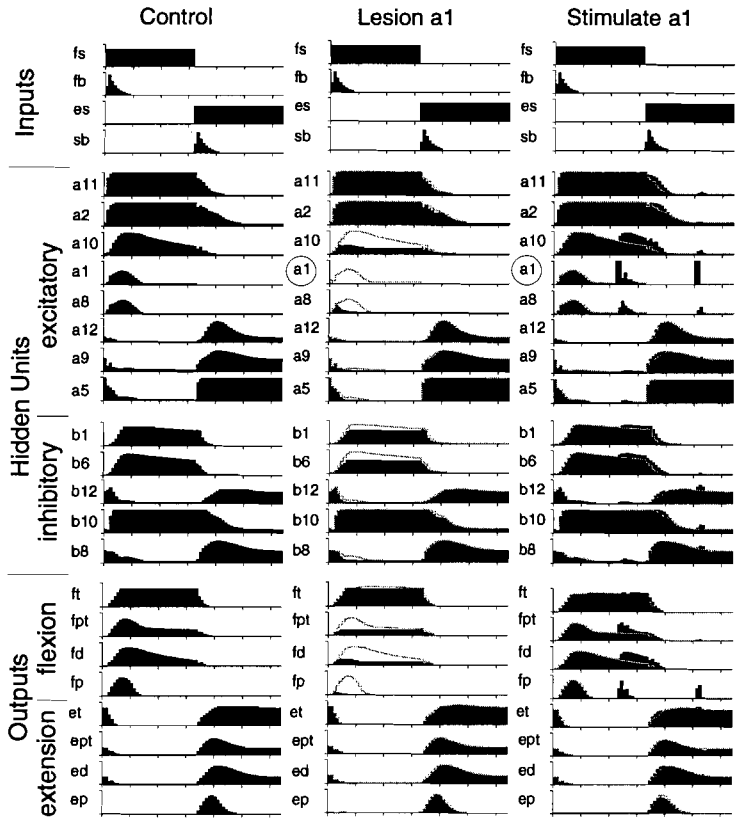
connected with each other and which also have strong reciprocal connections with the phasic output unit (fp).

Despite this tendency for sustained and transient signals to propagate through segregated pathways, most cells receive a mixture of phasic and tonic input signals. For example, the phasic flexor motor unit (fp) receives input not only from the phasic hidden units (a1 and a8), but also from cells with sustained activity (a11 and a10); indeed, some of the phasic activity of fp is derived from the difference between excitatory and inhibitory tonic cells with different onset times (e.g., a11 and b1). Such a subtraction of a delayed tonic pattern is used to produce phasic output in network simulations that receive only step inputs (Fetz et al. 1990).

We expected the network to develop reciprocal inhibition between the flexion and extension groups. Although a few inhibitory units activated during one phase of movement (e.g., extension) do affect primarily reciprocally activated motor units (e.g., b8 and b5), just as many units inhibit coactivated motor units (e.g., b12). Interestingly, the major drive on the tonic inhibitory units with connections to coactivated motor units is derived from recurrent feedback from the motor units (e.g., et to b4 and b9, which in turn inhibit ep). This recurrent feedback makes these inhibitory hidden units more analogous to the Renshaw cells of the spinal cord than to the Ia inhibitory neurons. Another group of inhibitory hidden units develop connections to both flexor and extensor motor units (b1 and b10), a pattern that has no known correlate in physiological studies.

The function of hidden units in the network can be tested by making selective *lesions*—by eliminating the activation of particular units and analyzing the behavior of the remaining network. Figure 7.6 illustrates the effects of removing a phasic hidden unit. The first column shows the control activations of representative units in the intact network. Removing the contribution of phasic unit a1 (middle column) eliminates the phasic activity of the output flexor units (fpt, and fp), as well as eliminating activity of the other phasic hidden unit (a8). It also reduces the sustained activity of the decrementing units (a10 and fd). These selective lesions have substantial effects because this network has relatively few units carrying a particular pattern, and these units are strongly interconnected.

The output effects of a given unit can also be tested by delivering a simulated *stimulus* and analyzing the propagated network response. The right column in figure 7.6 shows the effect of brief activation pulses delivered to a1 during the flexion and extension phase of the activity. The second pulse, during extension, evokes a response in the phasic output unit as well as in some of the active inhibitory hidden units. In contrast, the first pulse, delivered during the flexion phase, evokes a stronger response in the phasic output unit, as well as more prolonged



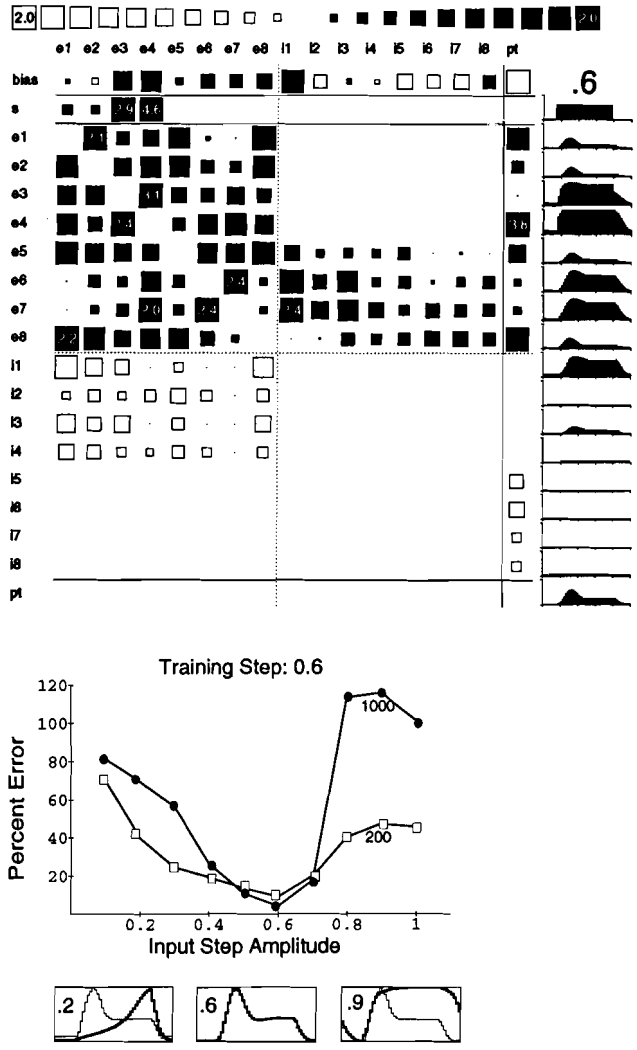
Effect of Lesioning and Stimulating a Hidden Unit in the Network of Figure 7.5

Figure 7.6 The activations of representative units are shown for the intact network (control, left column); after lesioning the phasic excitatory hidden unit a1 (middle column); and after stimulating the phasic excitatory hidden unit a1 (right column). The modified activations are shown in black, superimposed on a stippled profile of the control activations. (Reprinted from Fetz and Shupe 1990, with permission.)

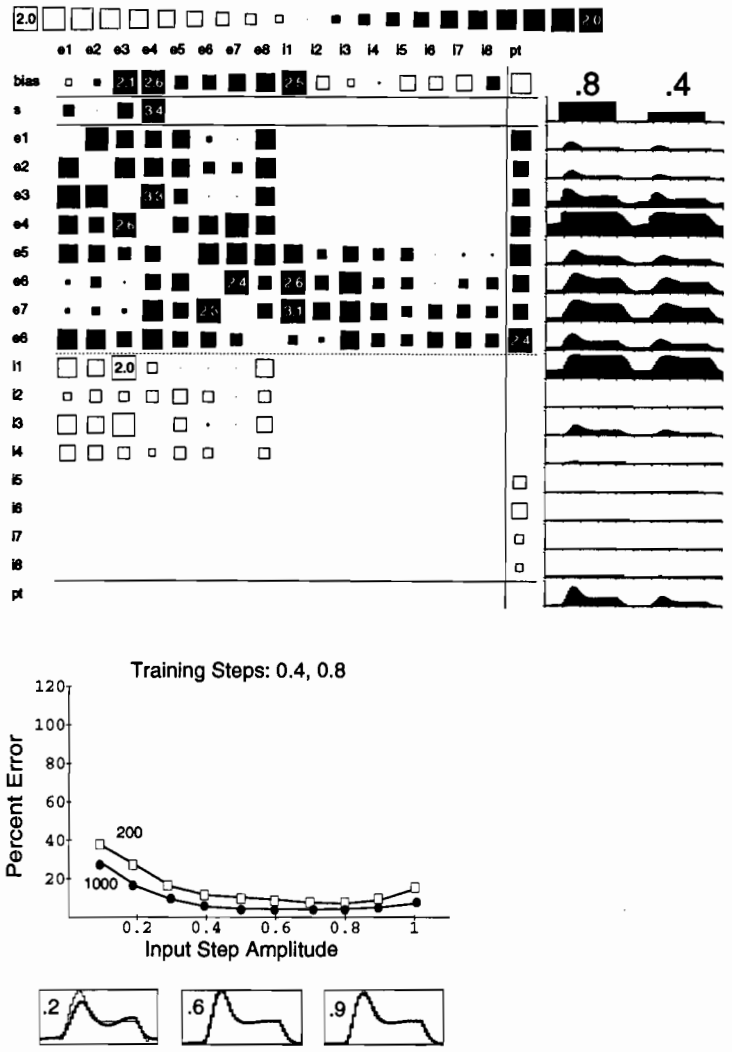
responses in the phasic-tonic and decrementing units (a10, fd), suggesting that the decrementing units carry a leaky integral of the phasic burst.

The fact that evoked responses were larger in certain phases of the movement than in others is clearly related to the gating function of the activation of the intervening units. Such modulation of evoked responses during cyclic motor activity has been well documented in physiological experiments. Plotting the evoked response as a function of time in the flexion-extension cycle revealed modulations that were not simply proportional to the activation of the stimulated or target cell, but were more complex functions of the activations and connections of the network (Fetz and Shupe 1990).

These networks were trained to generate the appropriate output patterns in response to specific and repeated inputs. We also investigated a network's ability to *generalize* across different stimulus-response



magnitudes—to generate output activation patterns proportional to the input step. In the step-tracking task, monkeys learned to generate force levels proportional to the size of the target step, and did so by proportionally scaling the discharge levels of the task-related neurons. For example, the discharge pattern of phasic-tonic motor units was proportional to the active force levels (Palmer and Fetz 1985). Such linearity was tested in a simple network that transforms a step input to a phasic-tonic output (figure 7.7). If a network was trained on only one force level and tested with inputs representing other force levels it typically generated outputs that deviated drastically from a proportional phasic-tonic pattern (figure 7.7, left). Going from 200 to 1000 training iterations produced a slight improvement in accuracy at the training level (0.6) but resulted in greater deviations at other levels. We obtained a network that generates a phasic-tonic output pattern in proportion to the input



Magnitude Scaling in Neural Networks Transforming Step Input to Phasic-tonic Output

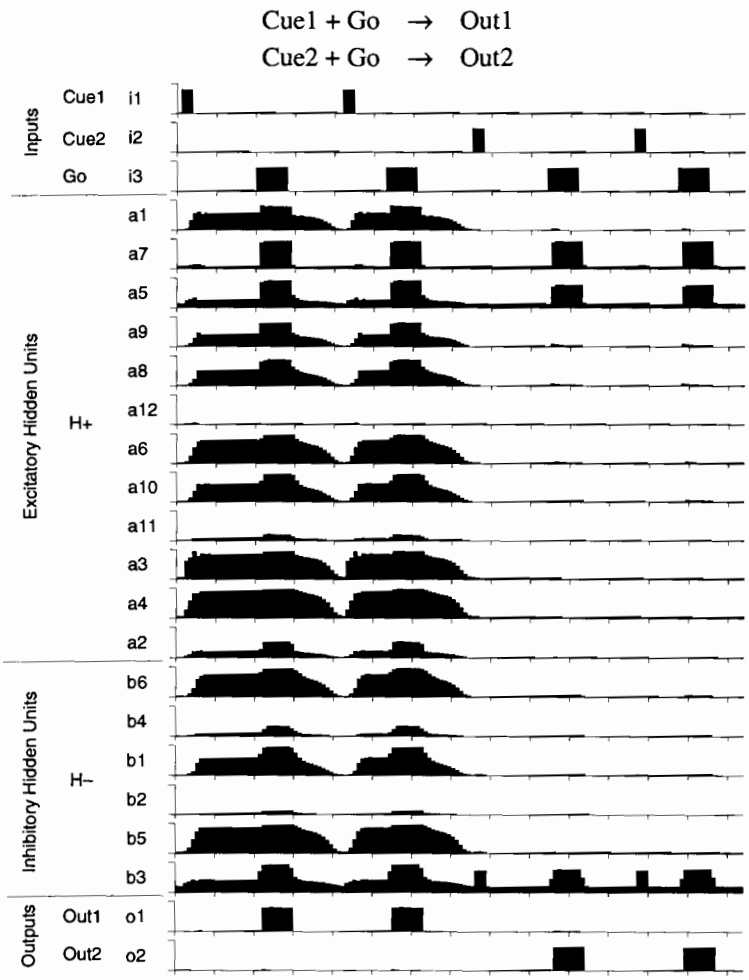
Figure 7.7 (Left) Network trained on one step size (0.6). (Right) Network trained from same starting point on two step sizes (0.4 and 0.8). Bottom graphs give percent error between the actual output and a phasic-tonic pattern proportional to the step input, after 200 training iterations (white boxes) and 1000 iterations (black circles). Samples below show normalized outputs (dark lines) superimposed on the phasic-tonic pattern (light lines) for input step sizes 0.2, 0.6 and 0.9, using the networks after 1000 iterations. (Reprinted from Fetz et al. 1990, with permission.)

step by simultaneous training at two step levels (0.8 and 0.4) (figure 7.7, right). When tested with intermediate input steps, the second network still generated proportional outputs. However, when tested with steps outside the range straddled by the training levels, the output still deviates slightly from a proportional response. Interestingly, the network that generalized the transform had a connection weight matrix similar to the network trained at a single level (both were trained from the same initial state); however, the network which generalized had a more equitable distribution of weights from the excitatory hidden units to the output (compare left and right networks in figure 7.7).

These simulations are obviously too simplistic to be taken as realistic models of the primate motor system. Nevertheless, even these highly abstracted networks reflect many of the features of the biological system in the monkey. For example, many different combinations of hidden units contribute to the same output. A phasic output pattern, for instance, has been generated by phasic hidden units, tonic excitatory activity minus delayed tonic inhibitory activity, and phasic-tonic excitatory input minus inhibitory bias. Furthermore, some complex activity patterns seen in premotor neurons of monkeys, such as bidirectional responses of RM cells, also appear in the networks. Even certain apparently paradoxical relations seen in monkeys, such as cortical units that covary with muscles which they inhibit, appear in networks and make contributions that are understandable in terms of other units. Thus, the networks have been useful in elucidating the function of many puzzling features of biological networks.

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. One variant of these tasks begins with an instruction cue that the monkey has to remember during a delay period. At the end of the delay a *go* stimulus signals the time to execute the appropriate response indicated by the cue. A simulation of this paradigm in a dynamic recurrent network is illustrated in figure 7.8. The network receives three inputs, consisting of the two separate cues and the go signal; the network has two outputs, corresponding to the response appropriate for each cue. Thus, following either cue 1 or 2, the go signal will generate activation of the corresponding output unit, 1 or 2. Figure 7.8 illustrates activations of the units in a trained network for several trials involving different delays. The connectivity matrix shows that the hidden units are configured to form a flip-flop that routes the go signal to the appropriate output. This network simulates the operation of short-term memory during the delay; some hidden unit activations (e.g., b3) carry mixed representations of cue and movement signals, which has been seen in premotor and prefrontal cortex neurons.

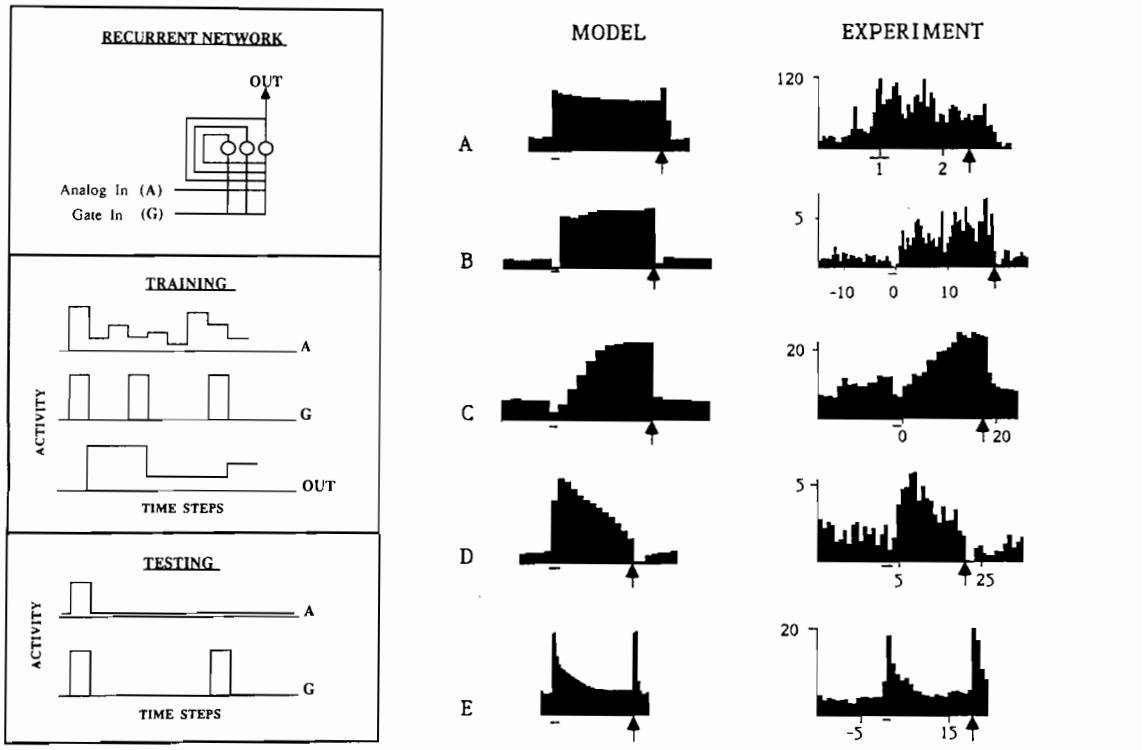


Activity Patterns in Network Simulating the Instructed Delay Task

Figure 7.8 Input pulses on inputs i1 and i2 are cues for the network to generate outputs o1 and o2, respectively, when the go signal i3 is presented after a delay.

In this simulation the output pulse is a replica of the go signal; this output pulse can be readily shaped to generate activation profiles resembling muscle activity by further training, or by cascading a second network that transforms the output pulse (as in figure 7.7).

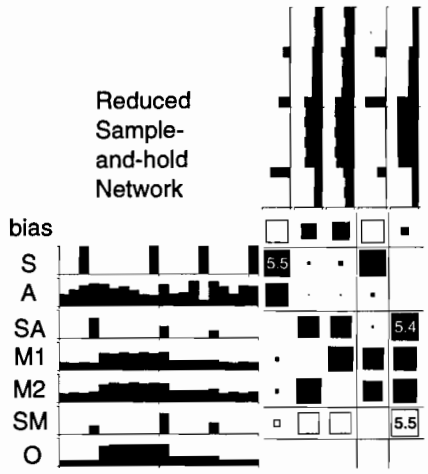
Another type of instructed delay task involves the requirement to remember the value of a particular stimulus. Williams and Zipser (1989a) trained recurrent networks to perform delayed nonmatch to sample task, in which the sampled variable to be remembered had binary values. More generally, the remembered variable can range over a continuum of values. Zipser (1990, 1991) has trained recurrent networks to simulate short-term memory of an analog value during the delay; in essence the network implements a sample-and-hold function (figure 7.9, left). The network has two inputs: an analog signal representing the stimulus



Sample-and-hold Delayed Response Task of Zipser (1990, 1991)

Figure 7.9 (Left) Simplified representation of input-output operation of the model, and waveforms used for training and testing. (Right) Comparison of activations of hidden units of the model and neural activity recorded in cerebral cortex of monkeys performing conditional delay tasks. The brief bar and arrow under the time axes represent the initial and final gate for the model, and the cues at the beginning and end of the delay period for the experimental data. The experimental histograms represent: (A) a neuron from posterior parietal area LIP during a delayed saccade task (Gnadt and Andersen 1988); (B) an inferotemporal neuron during a visual delayed match-to-sample task (Fuster et al. 1985); (C and E) frontal neurons in the principal sulcus during a delayed match-to-sample task (Fuster 1984); (D) a frontal cortex neuron during a delay choice task (Quintana et al. 1989). (Reproduced with permission from Zipser 1990, 1991; see these references for further information.)

value to be remembered and a gate signal specifying the sample and output times. The network output is the value of the analog input at the time of the previous gate. During the delay the activity of many hidden units resembles the response patterns of cortical neurons recorded in monkeys performing comparable instructed delay tasks (figure 7.9, right). The activity patterns of hidden units, like those of cortical neurons, fall into three main classes: activation profiles proportional to the remembered analog value, often with a decay; activation during the gate signal; and some combination of the two. The network simulations allow the function of each of these patterns observed in the animal to be interpreted in terms of its role in the task.



Reduced Network Performing a Sample-and-hold Function, Simulating Short-term Memory

Figure 7.10 A larger network with weight constraints was originally trained on the task, then reduced to the essential minimum by eliminating redundant and unnecessary hidden units. 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.

My laboratory has also investigated such short-term memory networks in order to further analyze their operation. To elucidate the underlying computational algorithm we have found it useful to constrain units to have either excitatory or inhibitory output weights (not both), and to reduce the network to the minimal essential network that performs the same function. A reduced network performing the sample-and-hold function with sign constraints on the weights is illustrated in figure 7.10. It 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 an elegant 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 held in memory by two excitatory units (M1 and M2), which maintain their activity by reciprocal connections and which feed their summed activity to the output. 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 reveals an elegant use of nonlinearity and integration to yield the appropriate remembered value.

It seems likely that networks with more units implement a comparable algorithm in a distributed manner.

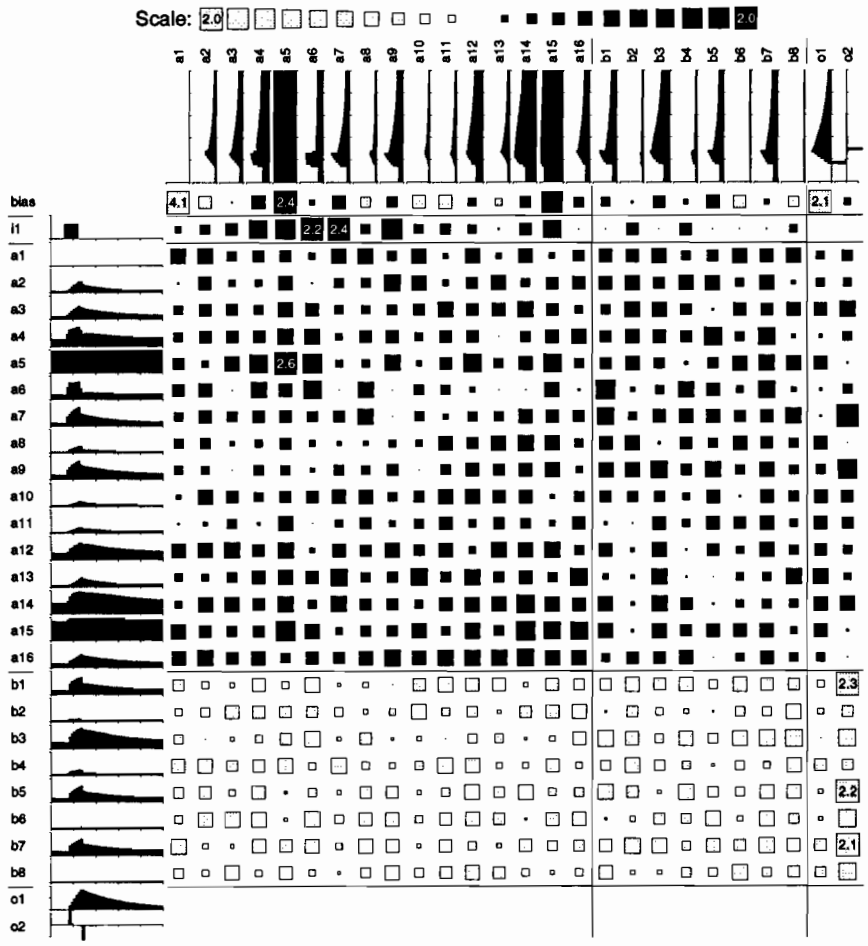
Neural Integration and Differentiation

Biological systems provide many examples of networks that integrate and differentiate neural activity. Munro et al. (1991) trained dynamic recurrent networks with backpropagation to generate the integral and differential of arbitrary input patterns. The connectivity of these networks was not designed to model any particular biological system; instead, our purpose was to investigate the mechanisms of neural computation of analytical functions and test the possibility of implementing multiplexed functions. Recurrent networks with architectures similar to that in figure 7.2 were trained with inputs of quasi-sinusoidal waveforms to produce either the leaky integral or the differential of the input or both. The nonrepeating input signal was a sinusoidal waveform whose frequency was varied up and down in random steps. The desired target output waveforms were calculated as the differential or the leaky integral delayed by two timesteps. Varying the frequency of the training signal insured that the network performed accurately over a broad range of frequencies. The trained networks compute these transforms for any arbitrary inputs.

By eliminating unnecessary units and combining redundant units and retraining the remaining network, it was possible to derive reduced networks that performed integration or differentiation or both. The computational algorithm was often clearer in the reduced networks. (However, smaller networks could not be trained from random starting weights to perform these functions.) The differentiating networks implemented a strategy of delayed subtraction of the input, and their hidden units often carried signals resembling the input. The integrator networks used recurrent connections between the excitatory hidden units, whose activation all resembled the output—i.e., the integral of the input.

Networks could also be trained to produce the integral and differential simultaneously on two separate outputs, as in figure 7.11 (Munro et al. 1991). These networks combined the strategies used by the pure differentiator and integrator in a distributed and overlapping manner. In general, a large number of hidden units were actively involved in the multiplexed computation and many of these contributed significantly to both outputs (compare vertical column of input weights to o1 and o2). The recurrent connectivity of the excitatory units is similar to that obtained for a simple integrator network. The activity patterns of most hidden units (both excitatory and inhibitory) carry a component that resembles the leaky integral of the input.

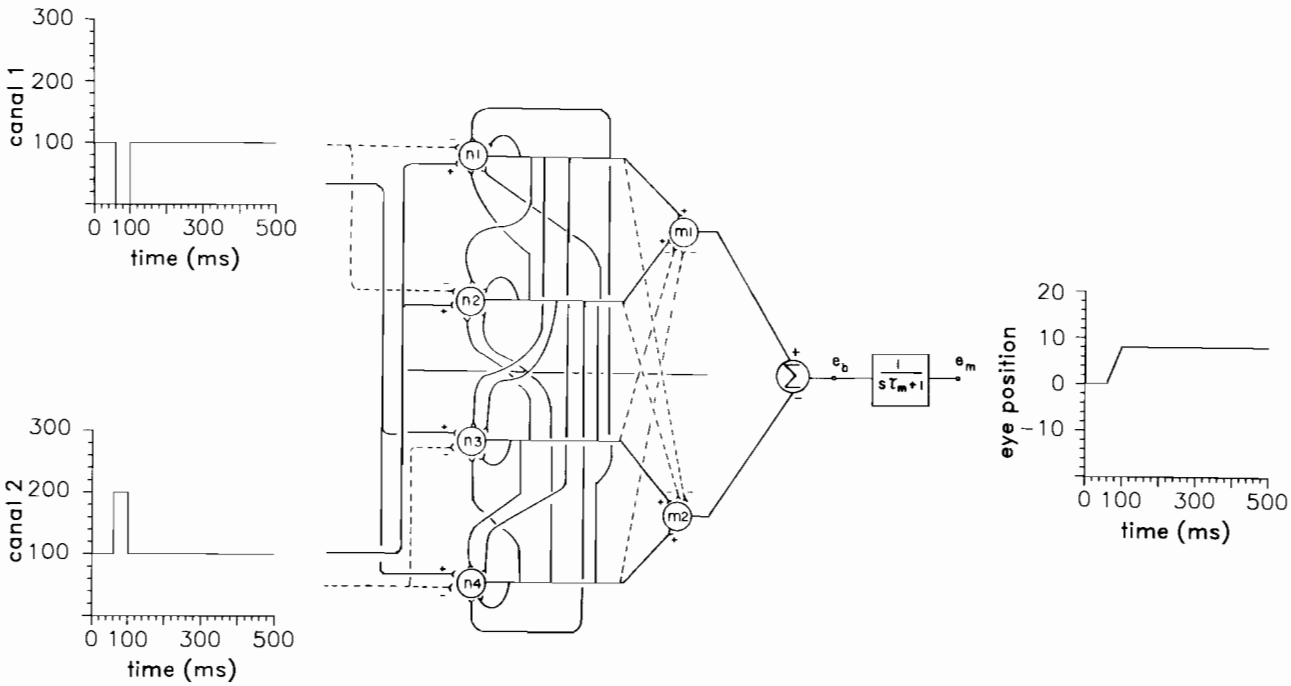
In biological motor systems, neural integrators have been postulated to transform transient commands into sustained activity, and to mediate the vestibulo-ocular reflex (VOR). Arnold and Robinson (1991) modeled



Dynamic Recurrent Network Generating Integral and Differential of the Input

Figure 7.11 Activity of a dynamic recurrent network trained with nonrepeating input waveforms to generate both the leaky integral (o1) and the differential (o2) of the input. The illustrated input waveform was a pulse, to show the activations of the hidden and output units.

the vestibulo-ocular integrator with a recurrent network whose connections resembled those of the vestibulo-ocular system (figure 7.12). The two input signals were representations of the reciprocal responses of opposed vestibular afferents to head movement; these connected in push-pull fashion to four interneurons which were interconnected to each other and to motor neurons. Since the 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 an exponential decay and a rectifying input-output characteristic. To train units with this nondifferentiable function they used a gradient descent method that involved tweaking the weights individually by a small amount, deter-



Connectivity of Dynamic Network Simulating Vestibulo-ocular System

Figure 7.12 Two opponent vestibular canal inputs are connected to four recurrently connected interneurons, which in turn connect to antagonist oculomotor neurons. Connections of units can be both excitatory (solid lines) and inhibitory (dashed lines). Eye position is derived from the difference of the motor neurons' outputs. (Reproduced from Arnold and Robinson 1991, with permission.)

mining the effect on the error, and using this operationally derived estimate of the error gradient to update all weights. The resulting network is shown in figure 7.12 with the activity profiles of vestibular canal inputs and resultant eye movements. This network allowed units to have both excitatory and inhibitory output connections. Integration was performed through positive recurrent connections between the interneurons. Removing a hidden unit in a trained network reduced the time constant of integration, but the network could be readily retrained. The network could also mimic more complex responses, such as postsaccadic drift.

Anastasio (1991a, b) trained dynamic networks of sigmoidal units with backpropagation to simulate velocity storage in the vestibulo-ocular system. A three-layer network model was trained to produce compensatory long-duration activity of oculomotoneurons in response to brief head velocity inputs. The network learned to produce the prolonged response, termed velocity storage, by developing lateral inhibitory interactions between the interneurons. These interneurons exhibited many of the response properties of VOR interneurons in the vestibular nucleus—i.e., low baseline firing rates, long time constants, and rectified and skewed responses. The model indicates that these properties are related and are the result of lateral inhibition. The model further replecated the consequences of vestibular lesions.

Lisberger and Sejnowski (1992a) used dynamic networks to investigate mechanisms of learning in the vestibulo-ocular system (see also the extensive discussion in Churchland and Sejnowski 1992). 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 tracking. These requirements led to changes in the weights of connections at two specific sites: the vestibular input to the flocculus and to the brainstem neurons controlling oculomotoneurons. A change in the balance of input from phasic-tonic and tonic vestibular efferents contributed to a change in the gain of the VOR, because the transient signal was integrated by a positive feedback loop (Lisberger and Sejnowski 1992a, b). 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.

FUTURE DIRECTIONS

These examples illustrate the use of dynamic recurrent networks to simulate sensorimotor behavior and to elucidate the underlying neural computations. To better understand the operations of biological networks, future development of these techniques can be directed toward incorporating more realistic neural properties into the units. The sigmoidal input-output function of the units is a mathematically convenient but artificial approximation to the response of biological neurons. One important property of physiological neurons is that their activity depends not only on immediate inputs, but also on their prior activity. History-dependent unit activity such as adaptation and postinhibitory rebound arises from the kinetics of time- and voltage-dependent ionic channels (Schwindt 1992; Spain et al. 1991; see also chapter 2). These properties have been amply documented for many neuronal types (reviewed by Llinás 1988). The activation kinetics of ionic channels make the neuron's activity depend on its past behavior as well as on its instantaneous input. These intrinsic modulations may be quite significant for the network solutions. For example, in monkeys performing the step-tracking task, the decrementing motor units may simply be exhibiting an adapting response to a tonic input. Our simulations to date using sigmoidal units were forced to generate this decrementing pattern by creating appropriate simultaneous input from the hidden units. Simulations with adapting units could generate both the decrementing and tonic motor unit responses to the same tonic inputs. The basic property of history-dependent activation can be modeled to a first approximation by passing the summed synaptic input through a leaky integrator in parallel with a direct gain element (Schwindt 1992). This has been accomplished by incorporating a passive decay in many simulations (Arnold and Robinson 1991; Lisberger and Sejnowski 1992a; Lockery and Sejnowski 1992). As physiological information becomes more detailed it will be possible to incorporate more specific types of time dependencies into network units.

A second feature of real neurons that may be quite significant for information processing is *discrete spiking*. The sigmoidal units transform a continuous function representing mean firing rate; in this sense, the units may be more properly interpreted as representing the mean activity of a large population of asynchronously spiking neurons. Real neurons integrate synaptic input to threshold and fire discrete spike trains, which may have functional consequences. For example, synchronous activity in spiking neurons can enhance the effectiveness of transmission of signals (Kenyon et al. 1990). Unfortunately, networks with spiking neurons cannot be trained using backpropagation. Other training methods, reviewed below, could be used to change the network weights. However, the training is much slower because many iterations

must be averaged before the results of each weight change can be assessed.

Although backpropagation is very effective in searching out solutions in weight space, it is fraught with uncertainty about finding the best solutions, and can only be used with units whose input-output function f is differentiable. Other training strategies can be used to find network solutions with more realistic units. For example, *random search strategies* (Baba 1989; Choi et al. 1991) sample a number of randomly selected points distributed in weight space and then pursue the best solutions with random perturbations of the weights. This strategy is computationally simpler and faster *per step* than backpropagation, and can be applied for networks of units with nondifferentiable response functions.

Another important reinforcement strategy is $A_{R\text{-}P}$ *training* (for *associative reward-penalty*), in which all the weights are modified in proportion to the output error and in proportion to local presynaptic and postsynaptic activations (Barto and Jordan 1987). This modified Hebb rule has greater biological plausibility, since reward and punishment would generally have global effects rather than specifically changing each weight in the direction of reducing error. In direct comparisons with backpropagation, $A_{R\text{-}P}$ training converged to similar network solutions, albeit more slowly (Mazzoni et al. 1991a, b). These two training algorithms are both applicable to neurons which have biologically plausible and nondifferentiable characteristics.

CONCLUSIONS

The unique insights provided by neural network simulations ensure their continued use in elucidating the operations of neural systems. The basic limitation of conventional physiological and antomical data is that they provide a very selective sample of a complex system, leaving a wide 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 interpretation of the data (Fetz 1992). A more objective method of bridging the gap is with simulations that provide complete network models. These models can incorporate the observed responses of units and can help explain the functional meaning of neural patterns. Thus, systems neurophysiologists can profitably use a combination of unit recording techniques and neural modeling to elucidate the network mechanisms generating sensorimotor behavior. Unit recordings can provide important information on the activity of related neurons, but the network models can provide working examples of complete solutions to sensorimotor behavior. To the extent that models can incorporate anatomical and physiological constraints, they can provide plausible explanations of the neural mechanisms underlying behavior.

ACKNOWLEDGMENTS

I gratefully acknowledge the invaluable help of Mr. Larry Shupe, who wrote the software for our dynamic recurrent network simulator, and has helped explore many of its applications. Others who worked on various simulations described here include Edmund Munro, Soren Impey, Venky Murthy, and Arnd Pralle. This work was initially supported by the Office of Naval Research (contract number N00018-89-J-1240), and by NIH grants RR00166 and NS12542.

References

Anastasio, T. J. 1991a. Neural network models of velocity storage in the horizontal vestibulo-ocular reflex. *Biological Cybernetics* 64:187-196.

Anastasio, T. J. 1991b. A recurrent neural network model of velocity storage in the vestibule-ocular reflex. ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 3, R.P. Lippmann, J.E. Moody and D.S. Touretzky, eds., Morgan Kaufmann Publishers, San Mateo, CA, pp. 32-38.

Arnold, D.B. and Robinson, D.A. 1990. A neural network that learns to integrate oculomotor signals. *Proc IJCNN-90,* II:693-696.

Arnold, D.B. and Robinson, D.A. 1991. A learning network model of the neural integrator of the oculomotor system. *Biologzcal Cybernetics* 64: 447-54.

Barto, A.G. and Jordan, M.I. 1987. Gradient following without back-propagation in layered networks, *Proc.* IEEE *First International Conference on Neural Networks,* II 629-636. San Diego CA, June 1987.

Baba, N. 1989. A new approach for finding the global minimum of error function of neural networks, *Neural Networks* 2: 367-373.

Buchanan, J.T. 1992. Neural network simulations of coupled locomotor oscillators in the lamprey spinal cord. *Biologzcal Cybernetics* 66: 367-374.

Churchland, P.S. and Sejnowski, T.J. 1992 THE COMPUTATIONAL BRAIN, MIT Press, Cambridge MA.

Fetz, E.E., 1992 Are movement parameters recognizably coded in activity of single neurons? *Behavioral and Brain Sciences, 15:* 679-690.

Fetz, E.E., Cheney, P.D., Mewes, K. and Palmer, S. 1989. Control of forelimb muscle activity by populations of corticomotoneuronal and rubromotoneuronal cells, *Progress in Brain Research* 80: 437 - 449.

Fetz, E.E. Shupe, L.E. and Murthy, V. 1990. Neural networks controlling wrist movements, *Proc IJCNN-90,* II: 675-679.

Fetz, E.E. and Shupe, L.E. 1990. Neural network models of the primate motor system, in ADVANCED NEURAL COMPUTERS, R. Eckmiller, ed., Elsevier North Holland/ Amsterdam, pp. 43-50.

Fuster, J. 1984. Behavioral electrophysiology of the prefrontal cortex *Trends in Neuroscience 7:* 408-414.

Fuster, J. M., Bauer, R.H., and Jervey, J.P. 1985. Functional interactions between inferotemporal and prefrontal cortex in a cognitive task. *Brain Res.* 330: 299-307.

Gnad, J.W. and Andersen, R.A. 1988. Memory related motor planning activity in posterior parietal cortex of macaque *Exp. Brain Res.* 70: 215-220.

Grillner, S, Wallen, P. and Brodin, L. 1991. Neuronal network generating locomotor behavior in lamprey: circuitry, transmitters, membrane properties and simulation *Annual Reviezus of Neuroscience 8:* 263-305.

Kenyon, G.T., Fetz, E.E. and Puff, R.D. 1990. Effects of firing synchrony on signal propagation in layered networks, ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 2, D.S. Touretzky, ed., Morgan Kaufmann Publishers, San Mateo, CA, pp. 141-148.

LeCun, Y. Denker, J, Solla, S., Howard, R.E. and Jackel, L.D. 1990. Optimal brain damage, in ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 2, D.S. Touretzky, ed., Morgan Kaufmann Publishers, San Mateo, CA, pp 598-605.

Lehky, S.R. and Sejnowski, T.J. 1990. Neural network model of visual cortex for determining surface curvature from images of shaded surfaces. *Proc. R. Soc. Lond.* B240, 251-278.

Lisberger, S.G. and and Sejnowski, T.J. 1992a. Computational analysis suggests a new hypothesis for motor learning in the vestibulo-ocular reflex. Technical Report INC-9201, Institute for Neural Computation, U.C. San Diego.

Lisberger, S.G. and and Sejnowski, T.J. 1992b. A novel mechanism of motor learning in a recurrent network model based on the vestibulo-ocular reflex. *Nature*, 360: 159-161.

Llinas, R. 1989. Intrinsic electrical properties of mammalian neurons: involvement in central nervous system function, *Science* 242: 1654-1666.

Lockery, S.R., Fang, Y. and Sejnowski, T.J. 1990. A dynamical neural network model of sensorimotor transformations in the leech, *Neural Computation* 2: 274-282.

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, *Journal of Neuroscience* 12: 3877-3895.

Lockery, S.R. and Sejnowski, T.J. in press. Realistic network models of distributed processing in the leech, in THE NEUROBIOLOGY OF NEURAL NETWORKS, Gardner, D. (Ed.) MIT Press, Cambridge MA.

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., *Cerebral Cortex 1*: 293-307.

Mozer, M.C. and Smolensky, P. 1989 Using relevance to reduce network size automatically *Connection Science 1*: 3-16.

Munro, E. Shupe, L. and Fetz, E. 1994. Integration and differentiation in dynamic recurrent neural networks *Neural Computation 6:* 405-419.

Palmer, S.S. and Fetz, E.E. 1985. Discharge properties of primate forearm motor units during isometric muscle activity. *Journal of Neurophysiology 54*:1178-1193.

Pearlmutter, B.A. 1989. Learning state space trajectories in recurrent neural networks. *Neural Computation* 1: 263-269.

Pineda, F.J. 1989 Recurrent backpropagation and the dynarnical approach to adaptive neural computation. *Neural Computation 1:* 161-172.

Pineda, F.J. 1987 Generalization of backpropagation to recurrent neural networks. Physical *Review Leetters* 18: 2229-2232.

Quintana, J., Fuster, J.M. and Yajeya, J. 1989. Effects of cooling parietal cortex on prefrontal units in delay tasks. Brain Res. 503: 100-110.

Rowat, P.F. and Selverston, A.I. 1991. Learning algorithms for oscillatory networks with gap junctions and membrane currents, *Network* 2: 17-41.

Schwindt, P. C. 1992. Ionic currents governing input-output relations of Betz cells, in SINGLE NEURON COMPUTATION, T. McKenna, J. Davis & S.F. Zornetzer, eds. Academic Press pp. 235-258.

Smith, A.W. and Zipser, D. 1989. Learning sequential structure with the real-time recurrent learning algorithm *Int. J.* Neural Systems 1: 125-131.

Spain, W.J., Schwindt, P. C. and Crill, W.E. 1991 Post-inhibitory excitation and inhibition in layer V neurones from cat sensorimotor cortex, *J. Physiol 434*: 609-626.

Tsung, F.-S., Cottrell, G.W. and Selverston, A.I. 1990. Experiments on learning stable network oscillations, Proc *IJCNN-90*, I:169-174.

Watrous, R. L. and Shastri, L. 1986. Learning phonetic features using connectionist networks: an experiment in speech recognition. *Technical* Report MS-CIS-86-78, Linc Lab 44, University of Pennsylvania, Philadelphia, PA.; see also IEEE *Int.* Conf on *Neural Networks*, San Diego, CA 1987.

Weinrich, M. and Wise, S.P. 1982. The premotor cortex of the monkey, Journal of *Neuroscience* 2: 1329-1345.

Williams, R. J. and Zipser, D. 1989a. A learning algorithm for continually running fully recurrent neural networks. Neural Computation 1: 270 - 280.

Williams, R. J. and Zipser, D. 1989b. Experimental analysis of the real-time recurrent learning algorithm. Connection Science *1*: 87-111

Williams, R. J. and Zipser, D. 1990. Gradient-based learning algorithms for recurrent connectionist networks. Technical Report *NU-CCS-90-9*, College of Computer Science, Northeastern University, Boston MA

Zipser, D. and Andersen, R.A. 1988. A back propagation programmed network that simulates response properties of a subset of posterior parietal neurons. Nature 331: 679 - 684.

Zipser, D. 1989. A subgrouping strategy that reduces complexity and speeds up learning in recurrent networks *Neural* Computation 1: 551-557.

Zipser, D. 1990. Short-term active memory: a recurrent network model of the neural mechanism Report 9003, Department of Cognitive Science, University of California San Diego, San Diego CA.

Zipser, D. 1991. Recurrent network model of the neural mechanism of short-term active memory *Neural* Computation 3: 179-193.

Zipser, D. 1992. Identification models of the nervous system. *Neuroscience* 47: 853-862.