Dynamic Models of Neurophysiological Systems

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. Biological experiments typically involve bits of the system: anatomical structures and their connections, effects of lesions on behavior, activity of single neurons in behaving animals. The missing element required to synthesize these pieces can be provided by neural network models: a method of generating working models of the complete system. New algorithms make it possible to derive networks that simulate dynamic sensorimotor behavior and incorporate anatomically appropriate 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 have three key properties (see RECURRENT NETWORKS: SUPERVISED LEARNING): First, the units are *dynamic*, meaning they can exhibit time-varying activity, which 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 composed of interconnected "sigmoidal" 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 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 and the desired target output activations is the error. The backpropagation algorithm optimally modifies the weights to reduce this error. This weight change implements a gradient descent of the error as a function of the weight. 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.

Applications

The applications for these dynamic recurrent networks fall into three general categories:

1. *Pattern recognition* applications involve identification 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 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 (Tsung, Cottrell, and Selverston, 1990; Rowat and Selverston, 1991).
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 and Shupe, 1990); the vestibulo-ocular reflex (Arnold and Robinson, 1991; Lisberger and Sejnowski, 1992; see also VESTIBULO-OCULAR REFLEX: PERFORMANCE AND PLASTICITY); and short-term memory tasks (Zipser, 1991). Recurrent networks can also simulate analytical transforms such as integration and differentiation of input signals (Munro, Shupe, and Fetz, 1994).

Oscillating Networks

Biological systems provide numerous examples of autonomously generated periodic motor activity: locomotion, mastication, respiration, etc. The neural circuitry underlying cyclic movements has been called a central pattern generator (CPG). Williams and Zipser (1989) trained dynamic networks to generate oscillatory activity with various frequencies. The smallest circuit that sustained quasi-sinusoidal oscillations consisted of two interconnected sigmoidal units.

Tsung, Cottrell, and Selverston (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. Rowat and Selverston (1991) developed algorithms to train oscillations in networks of units with biological properties like gap junctions and membrane currents, and trained networks with realistic constraints on connection weights to simulate known oscillatory activations. Simulations suggest that recurrent networks of sigmoidal units are quite robust in generating oscillatory activity, even to the point of meeting various constraints in phase and period. (But see CRUSTACEAN STOMATOGASTRIC SYSTEM for more subtle oscillator properties involving neuromodulation.)

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. Premotoneuronal (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), and dorsal root ganglion afferents—as well as 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 decrementing 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 motoneurons, we derived dynamic networks that generated as outputs the average firing rates of motor units recorded in monkeys performing a step-tracking task. Changes in target position were represented by step inputs to the network and/or by brief transient bursts at the onset of target changes. The input signals were transformed to either MU output patterns (phasic-tonic, tonic, decrementing, and phasic flexors and extensors) 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 and Shupe, 1990; Fetz, 1992 and 1993). In these simulations, the network solutions have several features which resemble biological situations but which were not explicitly incorporated:

1. Divergent connections of hidden units to different coactivated motor units are representative of divergent outputs of physiological PreM neurons (Fetz et al., 1989).
2. Many hidden units have discharge patterns resembling the outputs, but some have counterintuitive patterns that are 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 and Shupe, 1990; 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 also observed in physiological experiments. These networks can also be trained to scale their responses—i.e., to generate output activation patterns proportional to the size of the input. Moreover, their ability to generalize across stimulus dimensions 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 specific neurons in motor cortex, red nucleus and spinal cord (interneurons, and motoneurons) and afferent fibers was included with appropriate relative conduction delays; (2) the activity of representative subsets of these additional elements, where known, 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 have the ability to generate both types of behaviors, and have more realistic properties. Although these networks are still highly abstracted, they reflect many of the essential features of the biological system in the monkey. 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 cortical units that covary with muscles which 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 proven useful in elucidating the function of many puzzling features of biological networks, including some puzzling properties.

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 requires remembering 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 its 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 is illustrated in Figure 1. It consists of three excitatory and one inhibitory unit. The two inputs are the sample gate signal (S) and the random 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 nega-

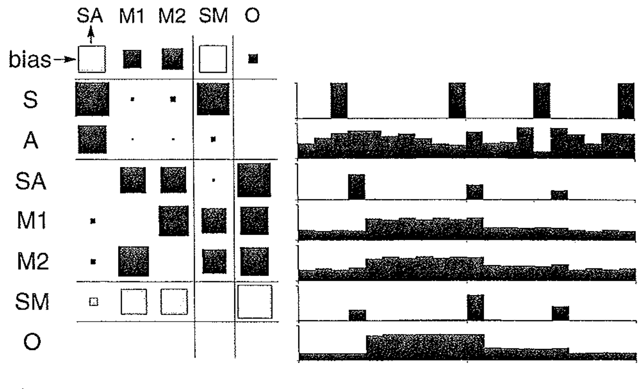


Figure 1. Reduced network performing a sample-and-hold function, simulating short-term memory. The units are indicated by abbreviations and their representative activation patterns, shown at right. The weights are indicated by squares (black = excitatory; grey = inhibitory) proportional to the connection 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.

tive 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 through 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*. As shown by its output weights, the function of *SM* is to subtract the previously held value from the integrating hidden units and from the output. This illustrates how the weights and activations of a complete network solution can reveal the underlying algorithm—in this case, an elegant use of nonlinearity and integration to yield the appropriate remembered value. It seems plausible that networks with more units implement a comparable algorithm in a distributed manner.

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). 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, which were interconnected to each other and to motoneurons. 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. Removing hidden units in a trained network reduced the time constant of integration, but the network could be readily retrained. The networks could also mimic more complex physiological responses, such as post-saccadic drift. Similarly Anastasio (see VESTIBULO-OCULAR REFLEX: PERFORMANCE AND PLASTICITY) trained dynamic networks of sigmoidal units with backpropagation to simulate velocity storage in the vestibulo-ocular system.

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 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. 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 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 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 techniques 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. This work was supported by ONR (grant N18–89–J–1240) and by NIH grants NS12542 and RR00166.

Road Map: Biological Networks
Background: I.3. Dynamics and Adaptation in Neural Networks
Related Reading: Reaching Movements: Implications of Connectionist Models; 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. ◆

Fetz, E. E., and Shupe, L. E., 1990, Neural network models of the primate motor system, in *Advanced Neural Computers* (R. Eckmiller, Ed.), Amsterdam: Elsevier North, pp. 43–50.

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, 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. Abst.*, 19:993.

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., 1991, Learning algorithms for oscillatory networks with gap junctions and membrane currents, *Network*, 2:17–41.

Tsung, F.-S., Cottrell, G. W., and Selverston, A. I., 1990, Experiments on learning stable network oscillations, in *Proceedings of the International Joint Conference on Neural Networks, 1990*, vol. 1, New York: IEEE, pp. 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.

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

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. ◆

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

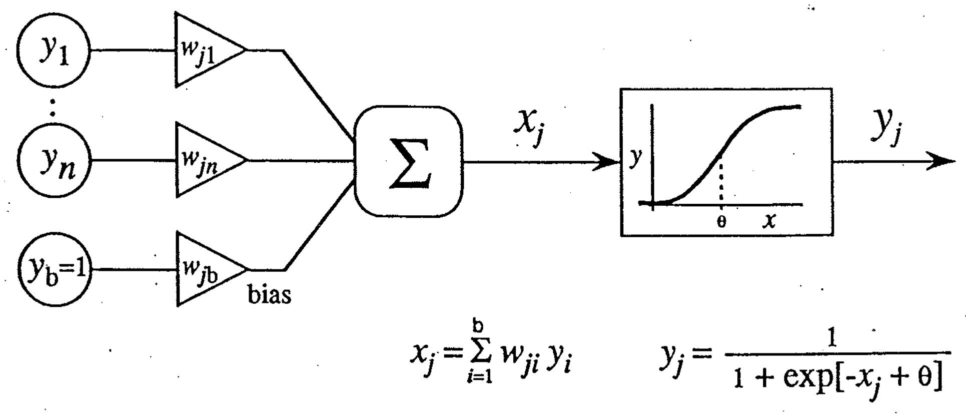

y_1
y_n
$y_b=1$
w_{j1}
w_{jn}
w_{jb}
bias
Σ
x_j
y
θ
x
y_j
$x_j = \sum_{i=1}^{b} w_{ji} y_i$
$y_j = \frac{1}{1+\exp[-x_j+\theta]}$

Fig. 1. Input-output properties for sigmoidal units used in dynamic recurrent networks. Each unit 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 "squashing function" to limit the output between 0 and 1.

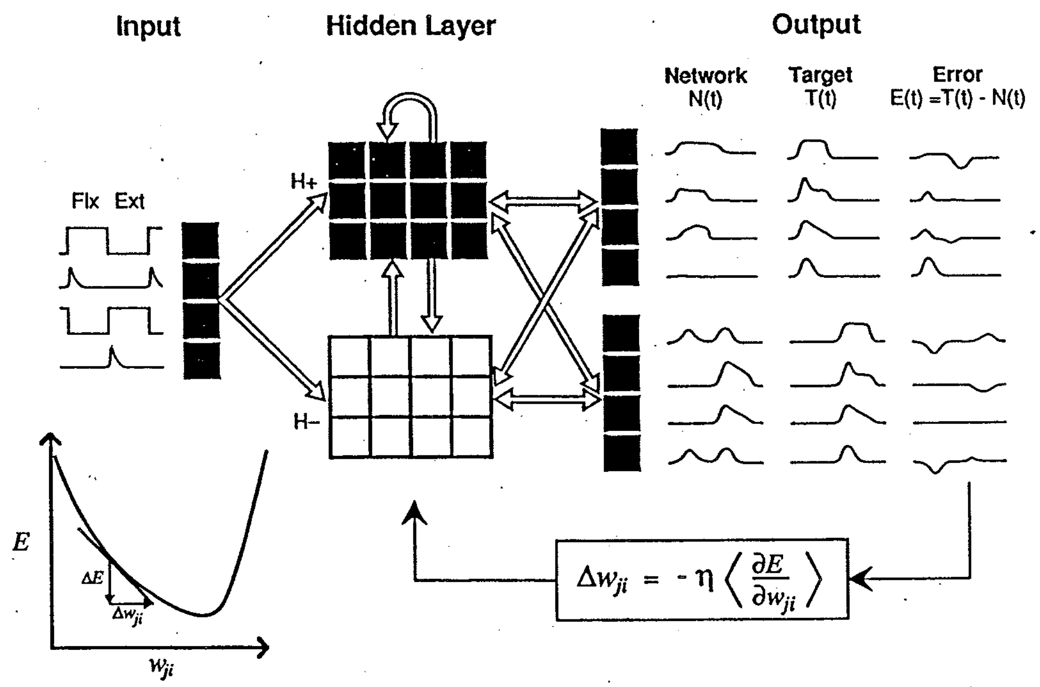


Fig. 2. Typical network architecture and training procedure used with dynamic recurrent networks. This network was used to simulate the step-tracking task (cf. Fig. 5 for a specific example). The network input consists of 4 representations of the step target position and target change; the output represents the firing patterns of eight representative motor units in flexor and extensor muscles. The connections of the intervening hidden units are 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.

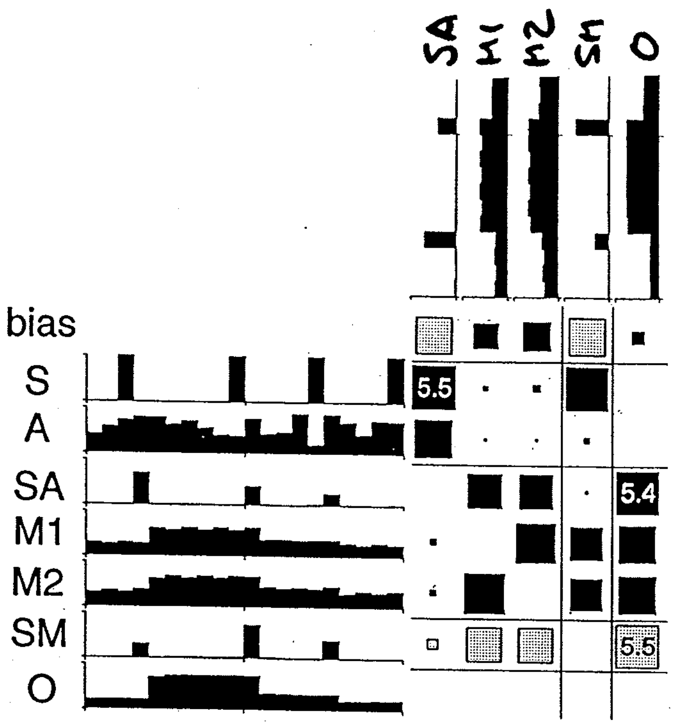
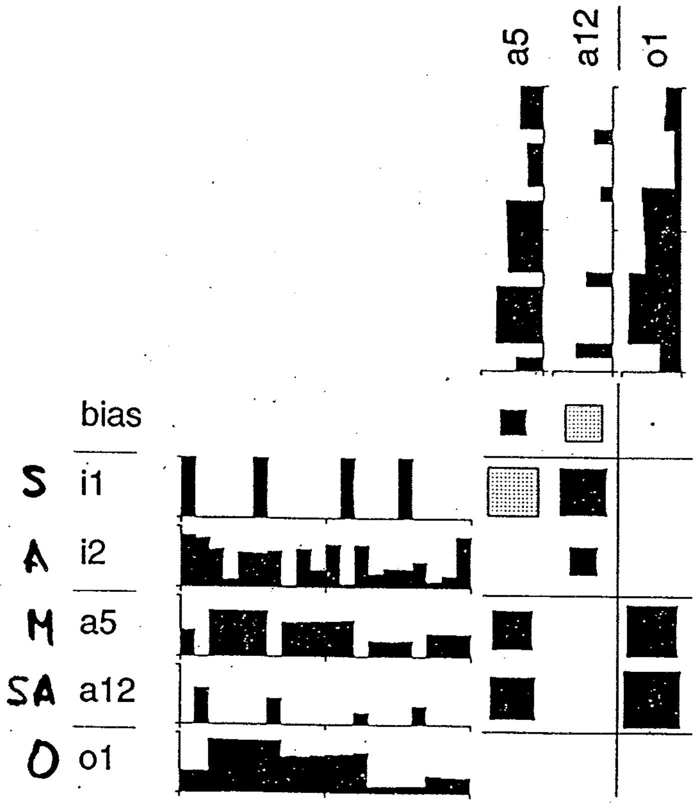


Fig. 10. Reduced network performing a sample-and-hold function, simulating short-term memory. A larger network with weight constraints was originally trained on the task, then reduced to the essential minimum by eliminating redundant and unnecessary hidden units. The two inputs are the sample signal (S) and a random analog value (A); the output (O) is the sustained value of the last sampled analog value.


a5
a12
o1
bias
S i1
A i2
M a5
SA a12
O o1