Recurrent Networks: Neurophysiological Modeling

Eberhard E. Fetz and Larry E. Shapere

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 computations of the hidden units are time-variant (i.e., show 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 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 backpropagation error correction. The resulting neural network models, however, not only find the same solutions, but usually take longer (Mazmanian, Andersen, and Jordan, 1991; see also REINFORCEMENT LEARNING IN MOTOR CONTROL).

Neural networks that emulate particular dynamic behaviors basically transform spatiotemporal inputs into appropriate spatiotemporal outputs. These networks are usually comprised of interconnected “sensomotor” units (units whose outputs are spatiotemporal functions of their inputs); this mimics a biological neuron’s property of summing 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 a network pattern. The network has four input units representing the firing rates of motor units in monkeys tracking such targets. To train the network, the synaptic weights between units are initialized randomly and the output response of the network is determined. The difference between network output pattern N(t) and the desired target output activities 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.

Biology plausibly learns through the general network learning algorithms will also find the same solutions, but usually take longer (Mazzoni, Andersen, and Jordan, 1991; see also REINFORCEMENT LEARNING IN MOTOR CONTROL).

Other algorithms, such as genetic algorithms (see LOCCOMOTOR, VERTEBRATE Behavior: Smith, 1993), or simulated annealing (Penfield and Rob- inson, 1991), can also be applied when the 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 spatiotem- poral 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 (Cunningham, Morgan, and Macek, 1990). The vestibulo-ocular reflex (Arnold and Robinson, 1991; Lieberbrun and Sjeijns, 1992) and short-term memory tasks (Zipser, 1991; Moody et al., 1995). Recurrent networks can also simulate automatic transformations such as integration and differentiation of input signals (Mano, Shapere, and Fetz, 1994).

3. Oscillating Networks

Among the many examples of autonomously generated periodic motor activity to be found in biological systems are locomotion, mastication, and language. 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 simplest circuit that sustained quasi-sinusoidal oscillations consisted of only two units (Figure 1). Tung et al. (1990) trained a network with the connectivity and sign constraints of neurons in the lobster gastric mill circuit to simulate the oscillations characteristic of the gastric mill circuit. 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 associated with task-related neurons. Premotor (PreM) cells were identified by postspike facilita- tion of target muscle activity in spikes triggered averages of EMG. During alternating wrist movements, the responses of different PreM cells corticomotor (CM), rubromotor- neurons (RM) showed both artifacts, and PreM interneurons 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 periods. Some RM cells show phasic burst firing 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 the dynamic networks that integrate 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, representing the eight units consisting of intercon- nected excitatory and inhibitory units. The activation patterns and connection matrix of units in such networks are illustrated elsewhere (Fetz et al., 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 neu- rons (Fetz et al., 1989). (2) Some hidden units have commutative discharge patterns also seen in biological neurons, e.g., bidirec- tional spatio-temporal firing; both forward and backward, 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 input stimulus and analyzing the propagated network response. Because of the nature of the task, the performance depends on the time it is delivered, as is also observed in physiologic- al experiments. These networks can also be trained to scale their response appropriately to the speed of the targets.

To generate more realistic models of the primate motor system, the same approach has been used with networks incorporating ad- ditions of other neurons, e.g., the projection neurons of the central unit was required to be part of the solution; and (3) in addition to the active target-tracking task, the network was required to sim- ulate reflex responses to peripheral perturbations of the limb. The resulting networks can generate both types of behavior and have more realistic properties. Some complex activity patterns seen in PreM neurons of monkeys, such as bidirectional activity patterns, cells, also appear in the networks. Even some apparently paradox- ical relations seen in monkeys, such as PreM units that co-activate with muscles that they inhibit, appear in networks and make contribu- tions that are understandable in terms of other units: its activity selects out inappropriate components of bidirectional activity patterns. Thus, network simulations have proved useful in eluci- dated 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 CONNECTIVE 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 encodes 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 excitatory and one inhibitory unit and (2) eliminating units with negligible activation or weak connections, then (3) retaining 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 neurons (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 signal (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 appropriately remembered value.

More sophisticated recurrent networks have been derived that perform delay-matching-to-sample tasks (Moodie 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 Anderson, 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 modulate 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-ocular system. The input signals represented the reciprocal responses of opposed vestibular afferents to head movement; these connected to four interneurons that were interconnected to one another 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 postcataract drift.

Lusher 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 cerebel- lar 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. 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 oculo- motor 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 as- sure 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 se- lective 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. N00014-89-1-0420) and NIH grants NS25422 and RR00166.

Road Map: Biological Networks

Related Literature: Learning in Neural Networks; Recurrent Networks: Learning Algorithmic Short-Term Memory

References


Figure 2. Reduced network performing a sample-and-hold function, simulating short-term memory. The units are indicated by abbreviations and representative acti- vation patterns at rest. The weights are indicated by squares (black = excitatory; gray = inhibitory) pro- portional to the connection from row unit to column unit (e.g., arrows), The two inputs are the sample signal (S) and a random deviating value (A), the output (O) is the sustained value of the last sampled signal value.