Supervised Learning Through Neuronal Response Modulation

Christian D. Swinehart
cds@brandeis. edu.
L.F. Abbott
abbott@brandeis. edu.
Volen Center and Department of Biology, Brandeis University, Waltham, MA 02454, U.S.A.

Neural networks that are trained to perform specific tasks must be developed through a supervised learning procedure. This normally takes the form of direct supervision of synaptic plasticity. We explore the idea that supervision takes place instead through the modulation of neuronal excitability. Such supervision can be done using conventional synaptic feedback pathways rather than requiring the hypothetical actions of unknown modulatory agents. During task learning, supervised response modulation guides Hebbian synaptic plasticity indirectly by establishing appropriate patterns of correlated network activity. This results in robust learning of function approximation tasks even when multiple output units representing different functions share large amounts of common input. Reward-based supervision is also studied, and a number of potential advantages of neuronal response modulation are identified.

1 Introduction

Correlation-based, Hebbian mechanisms of synaptic plasticity have been used with considerable success to explain the spontaneous development of selectivity and sensory maps in neural circuits (see Miller, 1996). However, when such plasticity mechanisms are applied to the development of networks that perform specific functions, rather than simply represent input data, a problem arises. To guide correlation-based synaptic plasticity, the activity of a naive neural circuit must be correlated in a manner similar to that of the final, functioning circuit. But such correlations usually arise only after the synapses of the circuit have been appropriately adjusted. The consequence of this is a chicken-and-egg problem: Which comes first, correlations or synaptic modifications?

The traditional answer to this question is that synaptic modifications come first, guided by a supervisor. The supervisor is a hypothetical neural circuit that assesses network performance, computes an error signal, and uses it to direct synaptic plasticity within the network. Such schemes work extremely well for many tasks (Widrow & Stearns, 1985; Chauvin & Rumelhart, 1995; Hertz, Krogh, & Palmer, 1991; Dayan & Abbott, 2001), making them attractive models for learning in biological systems. However, for biological applications, it is important to identify the pathways through which the supervisory circuit controls synaptic plasticity. In some cases, such as climbing fiber input to cerebellar Purkinje cells, such a mechanism appears to be in place. In other systems, such as cerebral cortex, an appeal must be made to some form of modulatory (perhaps dopaminergic; see Schultz, Dayan, & Montague, 1997) control of synaptic plasticity that is largely conjectural. Furthermore, modulatory pathways tend to be slow and nonlocal, making them poorly suited for the rapid, precise control of synaptic plasticity needed during task learning. These considerations lead us to explore the possibility that supervision of synaptic plasticity takes place indirectly rather than directly.

The scheme we study corresponds to correlations coming first. In other words, the supervisor modulates neuronal excitability in order to introduce correlations into network activity. These correlations then generate the synaptic plasticity needed to learn a task through Hebbian synaptic modifications that are not themselves subject to direct supervision. We are interested in supervision through response modulation because it is easy to see how this scheme could be realized in cortical circuitry. The response modulations that we consider can be generated through standard excitatory and inhibitory synaptic input. Therefore, in this scheme, the supervisory circuit can guide learning through the feedback projection pathways that are characteristic of cortical circuitry, and no appeal must be made to as-yet-undiscovered forms of modulation. It is important to realize that we are not proposing this scheme as an algorithmic improvement. Indeed, such indirect supervision of synaptic plasticity has disadvantages, and an important element of our study is to determine how detrimental these are.

In summary, we consider a network in which synaptic plasticity is purely Hebbian, a form typically used in unsupervised learning applications. We ask whether it is possible to implement supervised learning in such a network solely by communicating error signals to the network along conventional excitatory and inhibitory feedback pathways that modulate neuronal responsiveness but do not directly affect synaptic plasticity. Such a scheme is not optimal, so its virtues are not efficiency or elegance. Rather, we take this minimalist approach so that we can determine whether these well-established elements of cortical circuitry provide a sufficient basis for implementing supervised learning.

2 Response Modulation and Synaptic Plasticity

Neural networks used for supervised learning consist of units with nonlinear response functions connected together through interactions characterized by synaptic weights. The response r_i of network unit i to an input I_i is typically determined by a sigmoidal function,

$$r_i = \frac{1}{1 + \exp\left(-g_i(I_i - s_i)\right)} \,. \tag{2.1}$$

In biophysical terms, this can be thought of as the normalized firing rate generated by an input current I_i. The parameter s_i, which we call the shift, controls the value of I_i at which r_i reaches one-half its maximal value, while g_i, which we call the gain, determines the slope of the firing rate versus input curve at this point. Input currents are typically computed by multiplying presynaptic responses by synaptic weight factors and summing over all inputs.

The role of the supervisor is to compute an error by comparing actual and desired network output and to use this error to direct the modification of network parameters such that network performance improves. Conventionally, the major targets of this process are the synaptic weights. For example, weights can be modified to produce a stochastic gradient descent of the error function. We deviate from this procedure by employing a standard Hebbian synaptic modification rule that is not directly affected by the supervisor. At each stimulus presentation, the synaptic weight connecting unit i with response r_i to unit a with response R_a, w_{ai}, is augmented by a term proportional to the product of the pre- and postsynaptic activities,

$$w_{ai} \rightarrow w_{ai} + \epsilon_w R_a r_i \,, \tag{2.2}$$

where the parameter ϵ_w controls the learning rate. In addition, to prevent the runaway excitation that results from this positive feedback rule, divisive normalization is included. This consisted of dividing all the weights by factors that maintain the sums (for all a)

$$\sum_{i=0}^{N} w_{ai} = \alpha \tag{2.3}$$

at a constant value α. The important point here is that neither of the above rules, 2.2 or 2.3, involves the error function or any other form of supervisory signal.

All the supervision in our network takes place at the level of the gain and shift parameters governing the input-output function of equation 2.1. It is not unusual for supervised learning schemes to modify such parameters, particularly shift parameters. Furthermore, in our scheme, supervision of shift and gain parameters takes place through the same type of stochastic gradient-descent procedure used in conventional supervised learning algorithms. The novel element in our approach is that these parameters are the only targets of supervised modification. The reason that we restrict supervision to the shift and gain parameters of neuronal response functions is that, unlike the supervision of synaptic plasticity, such supervision can be accomplished by ordinary, fast excitatory and inhibitory synapses from neurons of the supervisory circuit onto neurons of the function-approximation network. Changes in the shift variable s_i correspond to having the supervisor provide either net excitatory or net inhibitory input to network neuron i. It has been shown that balanced, parallel modulations of excitatory and inhibitory input can modify the gain of a postsynaptic neuron (Doiron, Longtin, Berman, & Maler, 2001; Chance, Abbott, & Reyes, 2002; Prescott & De Koninck, 2003), and on the basis of this result we argue that the supervisor can also control and modify the gain variable g_i.

In summary, the supervisory circuit in our model can modify both the shift and the gain variables for each of the neurons in the network (though in our examples, only the input neurons are modulated, and modulating the output units alone is not effective) through normal excitatory and inhibitory synaptic pathways. To reiterate what was said in section 1, our goal is not to introduce a new algorithm, but rather to see if existing algorithms can still operate when supervision is restricted to well-established cortical pathways.

3 Function Approximation

We apply the proposed mechanism of supervised learning to function approximation, a well-studied task in the artificial neural network literature with obvious applications to biological systems (Poggio, 1990). In this task, network neurons are driven by a stimulus characterized by a single variable θ. The goal of learning is to produce a network output that matches a specified function or set of functions of θ. This is a very easy task for neural network learning that can be accomplished with a single layer of synapses modified, for example, by a delta learning rule (Widrow & Hoff, 1960). We consider this task because it allows us to illustrate clearly the features and limitations of the scheme we are studying.

Specifically, we consider a two-layer feedforward network architecture with purely excitatory connections, as shown in Figure 1. The network consists of N input units, responding to a stimulus variable θ (which takes values in the range from 0 to 2π), that drive M output units. The input units of the network, indicated by the lower row of circles in Figure 1, are driven by currents that are gaussian functions of the difference between θ and a preferred stimulus value, which is different for each input unit. Specifically, the input to unit i, I_i, is given by

$$I_i = G(\theta - \theta_i) + G(\theta - \theta_i - 2\pi) + G(\theta - \theta_i + 2\pi)\,, \tag{3.1}$$

where

$$G(\theta) = 1.5 \exp\left(-\frac{\theta^2}{2}\right) - 0.5\,. \tag{3.2}$$

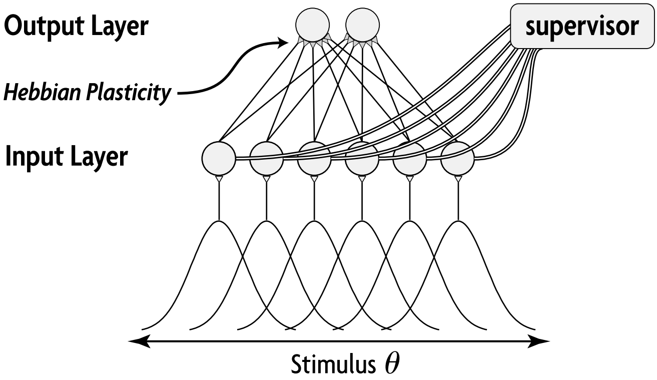

Figure 1: The function approximation network. Input units (lower row of circles) receive input tuned to the value of a stimulus variable θ, as indicated by the gaussian curves. The input units drive output units, shown at the top of the figure, through synaptic connections that are subject to Hebbian plasticity. A supervisor modifies the response properties of the input units through feedback projections. The task is to induce the firing rates of the output units to match specified functions of the stimulus variable.

The three terms appearing in equation 3.1 impose an approximate periodicity on the network, which is convenient (though not essential) because it removes edge effects. The values of the preferred stimulus parameters, θ_i for $i = 1, 2, \ldots, N$, are uniformly distributed over the range from 0 to 2π. The response of input unit i to stimulus θ, $r_i(\theta)$, is given in terms of the input I_i by equation 2.1.

The output of the network consists of the firing rates of the units appearing at the top of Figure 1. These are determined by the same firing-rate function as in equation 2.1, but their inputs are given by a weighted sum of the firing rates of the input units. Specifically, using $R_a(\theta)$ to denote the response of output unit a (for $a = 1, 2, \ldots, M$) to stimulus θ,

$$R_a(\theta) = \frac{1}{1 + \exp\left(-g_a\left(\sum_{i=1}^{N} w_{ai} r_i(\theta) - s_a\right)\right)}. \tag{3.3}$$

Here, w_{ai} is the weight of the connection from input unit i to output unit a. When we consider networks with a single output unit, we drop the output index and denote the weight from input i simply as w_i. The goal of learning for this network is to match the outputs $R_a(\theta)$, as closely as possible, to a set of stimulus-dependent target functions $F_a(\theta)$.

3.1 Action of the Supervisor. The supervisor in our network model computes an error by comparing the firing rates of the output units to the values of the target functions for each stimulus. It uses a stochastic gradient-descent algorithm to adjust the gain and shift values for the input units of the network of Figure 1 in such a way that the error,

$$E(\theta) = \frac{1}{2}\sum_{a=1}^{M} (R_a(\theta) - F_a(\theta))^2 \,, \tag{3.4}$$

is reduced after each stimulus presentation. Here, R_a is the response of output unit a, and F_a is the target response for that unit.

As stated above, synaptic weights in the network are subject to Hebbian synaptic plasticity as described by equations 2.2 and 2.3, with $\epsilon_w = 0.03$ and $\alpha = 5.5$. Error-based supervision is used to vary the shift and gain parameters (s_i and g_i) that control neuronal responsiveness. These are set to the initial values $s_i = 1$ and $g_i = 3$ for all units, but are then changed by the supervised learning algorithm. During each run, a stimulus value θ is chosen randomly in the range from 0 to 2π, and the resulting output rates are computed. Then the shifts and gains for all the input units of the network are updated according to the rules

$$s_i \rightarrow s_i - \epsilon_s \frac{\partial E(\theta)}{\partial s_i} \quad \text{and} \quad g_i \rightarrow g_i - \epsilon_g \frac{\partial E(\theta)}{\partial g_i} \,, \tag{3.5}$$

where ϵ_s and ϵ_g are small parameters that control the rate of response modulation. For our simulations, these took the values $\epsilon_s = \epsilon_g = 0.2/M$. This process is repeated until performance stops improving.

We could also adjust the corresponding parameters s_a and g_a for the output units, but for the examples we give, this is unnecessary. Instead, these have been held at their initial values $s_a = 1$ and $g_a = 3$ for all a. The adjustment of output shifts and gains is unnecessary in the examples we present because we have chosen parameters so that the mean of the output response, averaged across all stimuli, is equal to the stimulus average of the target function. This is not essential; it was done primarily to simplify the presentation.

It is useful to compare and contrast our approach with the conventional use of the delta rule in this situation. In the conventional approach in which synaptic plasticity is supervised, the error in equation 3.4 is differentiated with respect to the synaptic weight w_{ai}. This weight is then updated according to the rule (assuming a gain of one)

$$w_{ai} \rightarrow -\epsilon_w \frac{\partial E}{\partial w_{ai}} = \epsilon_w \left(F_a - R_a\right) R'_a r_i \,, \tag{3.6}$$

where R'_a stands for the derivative of the response of output unit a with

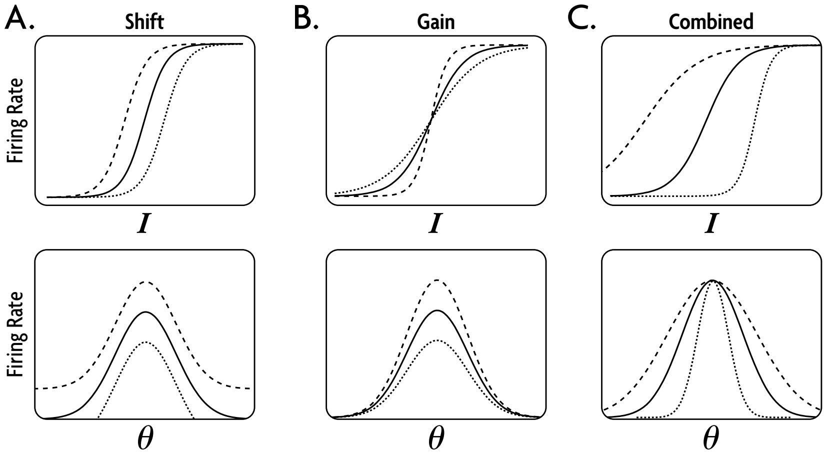

Figure 2: Effects of shift and gain modulations on firing rates and response tuning curves. The upper plots show neural responses as a function of the input current I, and the lower plots show them as a function of the stimulus parameter θ. (A) Changing the shift variable slides the response current curve to the left or right and moves the tuned response up and down. (B) Changing the gain variable changes the slope of the response current curve and has a roughly multiplicative effect on the tuned response. (C) Changing both variables changes the width of the tuned response.

respect to its input current. The term $(F_a - R_a)R'_a$ can be thought of as an error signal sent to output unit a that, in conjunction with the presynaptic firing rate r_i, controls modification of the weight w_{ai}.

In contrast, the error signals in our scheme, given by the derivatives in equation 3.5, are "sent" to the input units of the network rather than to the output units. Furthermore, these guide the modification of parameters affecting neuronal responses, not synaptic weights. Although the supervised learning rules in equations 3.5 and 3.6 may look similar in terms of mathematical abstraction, we stress that the modification described by equation 3.5 can be generated by normal, ionotropic synaptic transmission from the supervisory circuit to the targeted neuron, whereas those of equation 3.6 cannot. This is why we are considering such a modified form of delta rule learning.

The ability to change both the shift and gain variables that determine neuronal excitability provides considerable flexibility in modulated neuronal responsiveness. The different effects of shift and gain modulations on the firing rate of a model neuron, both as a function of its input current and of the stimulus variable, are shown in Figure 2. Changing the shift parameter translates the firing-rate curve right and left or, plotted as a function of the stimulus variable, shifts the tuning curve up and down (see Figure 2A). Changing the gain variable modifies the slope of the firing-rate curve and modulates the firing-rate tuning curves in a roughly multiplicative manner (see Figure 2B). Adjusting both variables allows the width of the tuning curve to be changed without an iceberg effect (see Figure 2C).

4 Results

In studying supervised learning through response modulation, we separately consider networks with a single output unit and networks with multiple output units. Obviously, the case of a single output unit provides less of a challenge than multiple outputs to any learning algorithm. Nevertheless, we consider it here because it provides a clear example of the interaction between supervised response modulation and unsupervised synaptic plasticity. We begin by showing that supervised response modulation, acting by itself without any accompanying synaptic plasticity, leads to a solution of the function approximation problem with a single output unit. This has some implications for network switching. However, it does not provide a satisfactory long-term solution because the supervisor-induced modulations do not produce any permanent changes in the network. This means that the task can be performed only, even after learning, with continuous input from the supervisor. This problem is resolved by adding Hebbian synaptic plasticity to the learning scheme. This allows the supervisor-induced modulations to be transferred into changes of synaptic strength. Ultimately, this transfer allows the network to function properly even in the absence of supervisory input.

Supervised learning through response modulation is more difficult in networks with multiple output units. In the multi-output case, situations often arise in which response modulation, acting without synaptic plasticity, cannot solve the function approximation task. As an example, consider an input unit that projects to two output units that are supposed to represent two different functions. For one of these functions, it might be appropriate to enhance the response of this input unit, while for the other, it may be necessary to decrease its responsiveness. Clearly, without access to the separate synapses that connect this single input unit to its multiple output targets, both of these criteria cannot be satisfied. In such situations, Hebbian plasticity does not merely act as a way of transferring supervisory modulation into permanent network changes; it must act in concert with response modulation for the task to be learned at all. This is indeed what happens. We find that a combination of supervised response modulation and unsupervised synaptic plasticity allows networks with multiple outputs to compute multiple functions, provided that the connection probability between the input and output layers is less than about 95%.

4.1 Networks with a Single Output Unit.

4.1.1 Learning and Switching Through Response Modulation. Because response modulation is the pathway through which supervision affects network responses in our studies, it is useful to start off by considering what happens when response modulation acts alone, without the Hebbian synaptic plasticity that will be added later. Therefore, we begin the study of networks with a single output unit by showing that function approximation can be accomplished solely on the basis of response modulation. For Figure 3, synaptic connection strengths were held fixed, while a gradient-descent supervisor varied the shifts and gains of the input units. In other words, we used equation 3.5 but not equation 2.2 during learning. Figure 3A shows the initial state of the network in which the output response is independent of the stimulus (upper panel), because all the input units have identical shifts and gains, as revealed by the identically shaped response curves in the lower panel. After the gradient-descent response modulation algorithm has acted, the output response matches the target function (upper panel of Figure 3B) due to the modulation of responses revealed by the modified response curves seen in the lower panel. To match the cosine-like target response, the input units selective for stimuli near zero and 2π have been upregulated by the supervisor, while those selective for stimuli near π have been downregulated.

Using supervised response modulation, the network can approximate a wide variety of functions (some examples are shown, along with the distributions of shift and gain values that produce them, in Figure 4). It is important to keep in mind that the distributions of shift and gain variables shown in the left column of this figure could arise from specific excitatory and inhibitory inputs generated by a supervisor circuit. Thus, each function computed by the network corresponds to a specific pattern of activity within the hypothetical supervisory circuit. If these patterns of activity are remembered and later recreated within the supervisor circuit, this will induce the function approximation network to compute the target function related to that pattern of activity. Thus, after learning has taken place, the supervisor can act as a controller, rapidly switching the input-output relationship of the function approximation network between prelearned states. Although we do not consider this form of switching further in this article, it provides an interesting mechanism by which one neural circuit can control, activate, and switch the function of another (for a related discussion, see Lukashin, Wilcox, & Georgopoulos, 1994).

In this network, the input unit responses act as basis functions for representing the output response. Because they do not provide a complete set for arbitrarily high frequencies, there are limits to the types of functions that can be accurately approximated. Limitations arise when the target function varies rapidly, as seen in Figure 4. Although these limitations exist, they are less severe than they would be in a function approximation network

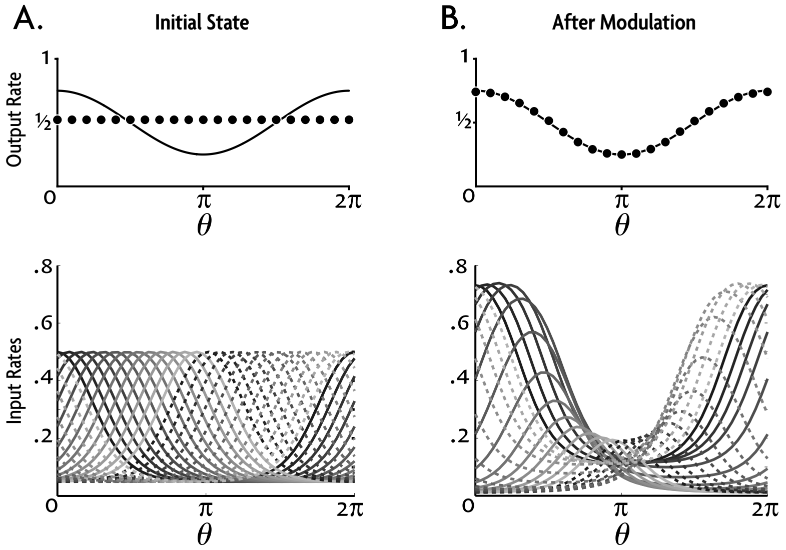

Figure 3: Function approximation by supervised response modulation acting alone without synaptic plasticity. Upper curves show the response of the single output unit to various stimulus values (dots) and the target function (line). The lower panels show a sampling of the responses of the 230 input units as a function of the stimulus value. (A) State of the network before learning. All input responses have the same shifts and gains, and the output is independent of the stimulus. (B) State of the network after learning. The output unit responses match the target function due to modulation of the input unit responses.

that relied solely on synaptic modification. This is because the tuning curve narrowing seen in Figure 2C can somewhat ameliorate problems with approximating rapidly varying functions.

In the following examples, we choose to approximate sinusoidally varying functions and do not present examples with other types of functions. All the networks shown can produce equivalent results with any target functions for which the input responses provide an adequate basis.

4.1.2 Transfer of Learning to Synapses. In the previous section, we considered supervised response modulation acting alone. We now add to this a Hebbian plasticity mechanism. In other words, we now use both equation 3.5 and equation 2.2 during learning. The combined effect of supervised response modulation and unsupervised synaptic plasticity is illustrated in Figure 5. As before, the network is initialized with uniform weights and all shifts and gains set to the same values. The supervisor then modifies response properties to minimize the output error. Early on during the learn-

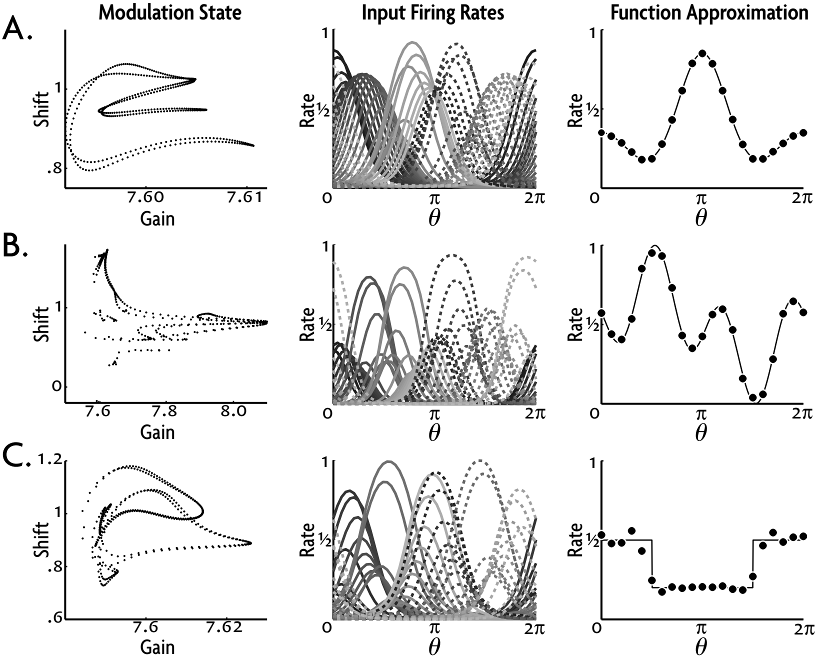

Figure 4: Examples of learning through response modulation. Three different functions are approximated by response modulation. The left column shows the shift and gain variables for all 460 of the input units of the network, each dot representing one input unit. Gains and shifts can also be seen in the sampling of input unit response curves in the middle column. The right column shows the output unit response (dots) and target function (line) plotted against the stimulus value.

ing process (top row of plots), the performance of the network relies almost entirely on the response modulation of the input units produced by the supervisor (illustrated by the distribution of input responses in the top row, left panel). At this point, the weights have hardly changed from their initial values (as seen in the top row, center panel). However, as the simulation progresses, the weight changes become progressively larger (second row, center panel), and the response modulations become progressively smaller (second row, left panel). Ultimately, the weights take on the cosine shape of the target function (third row, center panel), and the responses are almost uniform for all the input units (third row, left panel), as they were at the beginning of the learning process. Note that a stable equilibrium is reached when Hebbian modification and response modulation act together.

Once the response modulation and Hebbian plasticity have equilibrated, the supervisory input can be removed altogether, returning all shifts and

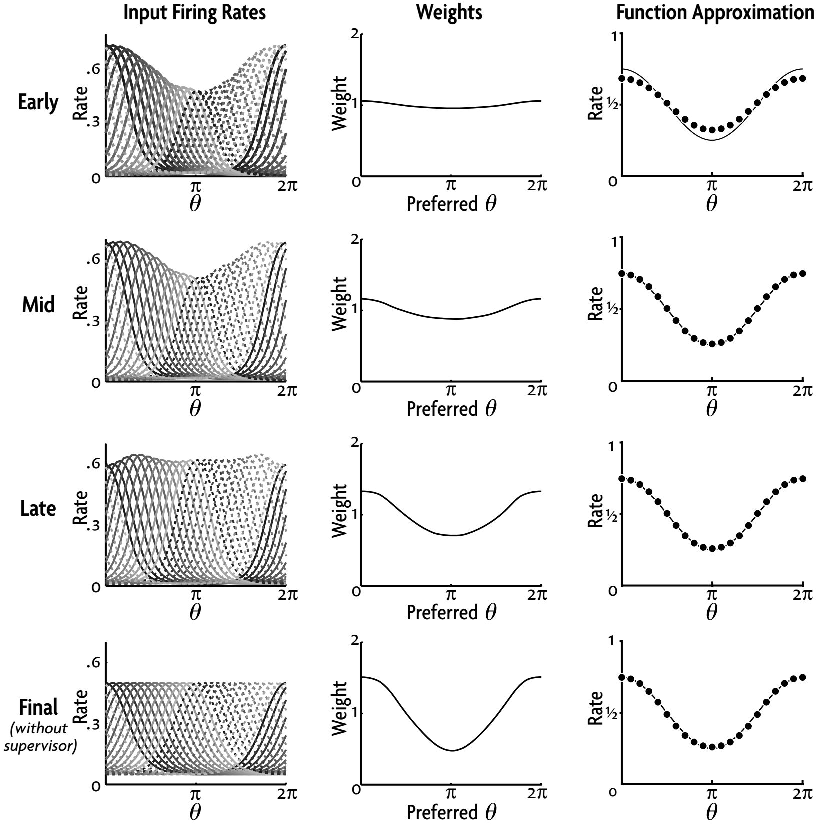

Figure 5: Supervised response modulation along with unsupervised synaptic plasticity allows for the transfer of learning to the synapses. The left column shows a sampling of the responses of the 230 input units as a function of the stimulus value. The middle column depicts the synaptic weights of the network plotted as a function of the preferred stimulus value for the presynaptic neuron. The right column shows the responses of the output unit (dots) and the target function (line), plotted against the stimulus value. The top three rows of plots, from top to bottom, show the gradual transfer of learning as the network changes from relying primarily on response modulation (top row of plots) to relying primarily on the pattern of modified synaptic weights (third row of plots). The bottom row illustrates that the network can perform fairly well even when response modulation is totally eliminated, once the appropriate pattern of synaptic weights has been established.

gains to their default values, and yet the network can still generate a good approximation of the target function (bottom row of Figure 5) (although, for stability, this necessitates the deactivation of Hebbian plasticity). Unsupervised synaptic plasticity thus allows the supervisor to contribute progressively less as the burden of representing the target function is taken up by the synapses.

Supervised response modulation plays three critical roles in guiding the Hebbian development of synapses capable of performing the function approximation task. First, because supervised response modulation acting alone can solve the task, the supervisor can act through the input units to effectively clamp the output to the correct response profile while Hebbian plasticity is taking place. Second, by increasing the responsiveness of appropriate input units while clamping the output to the target function, supervised response modulation sets up the appropriate pattern of correlation across the synapses of the network to guide Hebbian modification. For example, input units that are important contributors to the correct output response will be pushed to high levels of responsiveness by the supervisor, enhancing their correlations with the correctly clamped output unit. This causes the synapses connecting such units to the output to grow rapidly. Input units not needed for the task will be made unresponsive by the supervisor, so their synapses to the output unit will not be enhanced by the Hebbian modification rule. Instead, these synapses will be weakened due to the synaptic normalization constraint.

Finally, we consider a third role for supervised response modulation on the basis of an analysis of Hebbian modification. The form of synaptic plasticity we are using, Hebbian synaptic modification in conjunction with divisive normalization, ultimately sets synaptic weights in this case equal to

$$w_i = \frac{\alpha\langle Fr_i\rangle}{\sum\limits_j \langle Fr_j\rangle}, \tag{4.1}$$

where

$$\langle Fr_i\rangle = \frac{1}{2\pi}\int_0^{2\pi} d\theta\, F(\theta) r_i(\theta)\,. \tag{4.2}$$

To simplify the analysis, we consider a linear approximation for the response function of the output unit rather than the full sigmoidal form of equation 3.3. In this case and for these weights, the condition that the output response matches the target function,

$$R = \sum_{i=0}^{N} w_{ai} r_i = F\,, \tag{4.3}$$

requires that

$$\alpha \sum_{i=1}^{N} r_i(\theta) r_i(\theta') = \delta(\theta - \theta') \sum_{j=0}^{N} \langle F r_j \rangle \,. \tag{4.4}$$

The third role of supervised response modulation is to make this equation as near to an equality as possible. The accuracy with which the sum on the left side of this equation can match a δ function profile depends on the narrowness of the tuning curves of the input units. This places a limit on the degree to which rapidly varying target functions can be reproduced, but modifications in the gain and shift variables can improve this situation by narrowing the input tuning curves. More important, supervised response modulation acts to ensure that the normalization condition implied by equation 4.4 is met, and this is what ultimately allows Hebbian plasticity to solve the problem (Salinas & Abbott, 2000). Thus, by acting on these multiple levels, supervised response modulation guides Hebbian plasticity to a solution of the function approximation task.

4.2 Networks with Multiple Output Units. Networks with multiple output units present a greater challenge to the form of supervised learning we are proposing than do single-output networks. In particular, situations frequently arise where the representation of different functions by different output units cannot be achieved by response modulation alone due to shared input. Two cases are simple to analyze. If the connectivity between the input and output units is all-to-all with equal weights, response modulation alone is clearly unable to produce different responses in the output units. With all-to-all coupling, all the output units receive the same total drive, and whatever modulation is done at the input level affects all of the output units in the same way. Unsupervised synaptic plasticity does not help because the input correlation structure seen by the synapses to each output unit is identical, so the synapses will all be modified in an identical manner. Basically, the problem with all-to-all coupling is symmetry; all the output units are equivalent, and supervised modulation of input responses is not sufficient to break this symmetry and allow the output units to respond differently to the stimulus. Although we have assumed that the synapses take identical values, setting the initial synaptic weights to different values does not fix this problem.

At the opposite extreme, if the coupling from input to output units is so sparse that each input unit projects to just a single output unit, the situation reduces to multiple copies of the single-output case, and the analysis becomes a trivial extension of what was done in the previous section. In this section, we consider intermediate cases where the input-to-output connectivity is not all-to-all, but there is nevertheless considerable overlap in the input to different output units. We start by considering the case of two-output units and construct networks with various amounts of overlap in

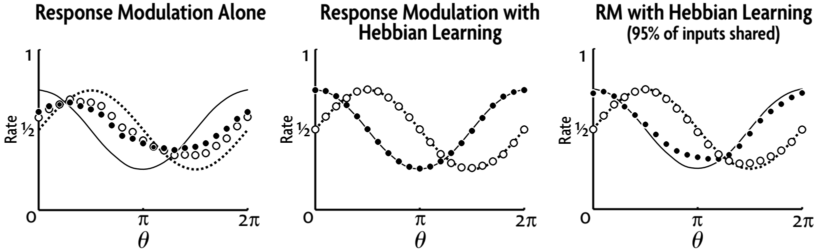

Figure 6: Unsupervised synaptic plasticity allows learning to succeed where interference causes supervised response modulation, acting without synaptic plasticity, to fail. In the left panel, a network of 460 input units connected with 88% overlap to two output units was simulated with response modulation acting without synaptic plasticity. The responses of the two output units, denoted by filled and open dots, failed to match the two target functions, indicated by dashed and solid curves, as a function of the stimulus value. When Hebbian plasticity was included, the two output responses accurately matched the target functions (middle panel). The combination of supervised response modulation and Hebbian plasticity continued to produce accurate outputs until the common input to the output units was increased to 95% (right panel).

the projections they receive from the input units. For an overlap of q, the number of input units that project to both output units is qN, and the number that project to only a single output unit is $(1 - q)N$. We ask whether, in such cases, unsupervised synaptic plasticity can exploit small differences in the drive to each output unit to break the symmetry and allow the output units to represent different functions.

Figure 6 illustrates the ability of unsupervised synaptic plasticity to play the role of a symmetry-breaking mechanism. In the first panel, supervised learning acting without synaptic plasticity has set the shifts and gains to their optimal values, but due to the degree of interference caused by shared input, the approximation is quite poor, and neither target function has been matched. The network has essentially split the difference between the two functions, with only small disparities between the responses of the two output units. However, when Hebbian plasticity is activated, it is able to exploit and amplify these small differences to improve performance dramatically. This ultimately leads to a match of the two different target functions (see the center panel of Figure 6). At this point, the supervisory input is no longer necessary (provided that the Hebbian process is halted). Thus, the combination of supervised response modulation and unsupervised synaptic plasticity allows the network to perform this task at a level that could not be achieved via response modulation alone.

The problem of indirectly supervising the plasticity of NM synapses by modulating only N neurons might at first appear to be a crippling limitation of response modulation. The example of Figure 6 shows at least one case in which this problem is not nearly as severe as might have been imagined. Separation of the two output units could still be achieved when they shared up to 95% of their inputs. However, it is critical to the success of supervision by response modulation that the requirement of a unique component for the input to each output unit scale appropriately as the size of the network and the number of output units increase. Initially, we investigate this issue by varying the amount of shared input in a network with two output units.

The result of varying the proportion of shared inputs in two-output networks of different sizes, when both supervised response modulation and unsupervised synaptic plasticity are active, is shown in Figure 7. In this figure and in Figure 8, performance is quantified by computing the error of equation 3.4, divided by the number of output units. The left panel of this figure shows that when learning occurs via response modulation alone without synaptic plasticity, errors begin to grow once the proportion of shared inputs exceeds 50% (dashed curves in Figure 7A). Performance is virtually identical for different input population sizes. With Hebbian plasticity included, the required number of unique inputs decreases dramatically, with little error accumulating until 90% to 95% of inputs are shared (solid curves in Figures 7A and 7B). The point of the transition from small errors to large errors appears to be roughly the same for all the network sizes studied (see Figure 7B). The main effect of increasing the number of input units is to make the transition point, where the function approximation network fails, sharper. This suggests that a discontinuous phase transition occurs at a critical percentage of about 94% shared input in the $N \to \infty$ limit.

We now extend these results to networks with more than two output units. In this case, the proportion of shared inputs (q) is not appropriate for describing all the different possibilities for sharing projections from the input units. Instead, we use the connection probability (p) to characterize the networks we study. To construct these networks, we introduce a connection between any one of the N input units and any one of the M output units with probability p. If such a connection is formed, it is subject to Hebbian plasticity. If no connection forms during this stochastic initial wiring, the connection remains absent for the entire duration of the simulation. The connection probability controls the sparseness of the network in that small values of p correspond to sparse connectivity.

Figure 8 shows function approximation errors for networks with different numbers of output units as a function of the connection probability p, for two sizes of input unit populations. In this case, both response modulation and synaptic plasticity are activated. As in the two-output case, interference does not become a serious impediment to learning until p reaches the .9 to .95 range, indicating that truly unique inputs are not necessary. Rather, the requirement is a certain degree of sparseness. Also noteworthy is the fact that the output population size can approach one-third of the total number

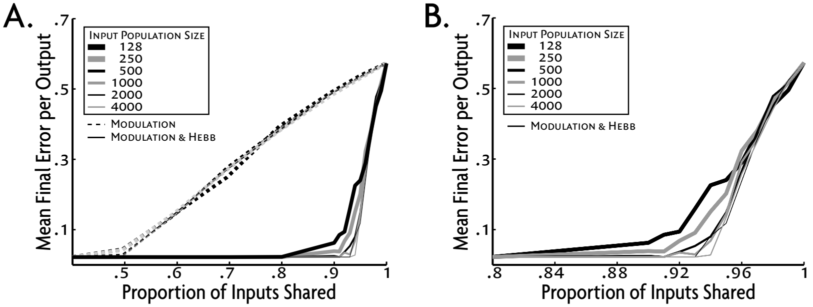

Figure 7: Error as a function of the proportion of shared input for a two-output network with different numbers of input units. Dashed lines show the mean error per output unit resulting from supervised response modulation without synaptic plasticity. Solid lines are from runs that also employed Hebbian plasticity. Individual lines correspond to different numbers of input units (N). The functions being approximated are cosine and sine. (A) Network performance degrades as the proportion of inputs projecting to both outputs increases. (B) Detailed view of the results for supervised response modulation with Hebbian plasticity shown in panel A.

of inputs before performance begins to suffer from interference (provided p values are not too large).

Our results indicate that connection probability is the dominant factor that controls whether a network with multiple output units, using both supervised response modulation and Hebbian plasticity, can function properly. The required sparseness in the connectivity is not stringent. Furthermore, approximation of multiple functions is possible even when output population size is a significant fraction of the total input population size.

The analysis of networks with multiple output units is more difficult than in the single-output case, but some of the same basic principles apply. In this case, the supervisor cannot clamp the output units to their target functions, but the existence of even a small number of symmetry-breaking synapses is sufficient to break this impasse. These synapses initially get quite strong and drive the output units away from the degenerate state in which they are all the same, which starts off the combined response modulation Hebbian learning process. Through this process, the bulk of the synapses ultimately come to obey the multi-output generalization of equation 4.1,

$$w_{ai} = \frac{\alpha \langle F_a r_i \rangle}{\sum\limits_j \langle F_a r_j \rangle} . \tag{4.5}$$

Similar to the result in the one-output case and making the same linear

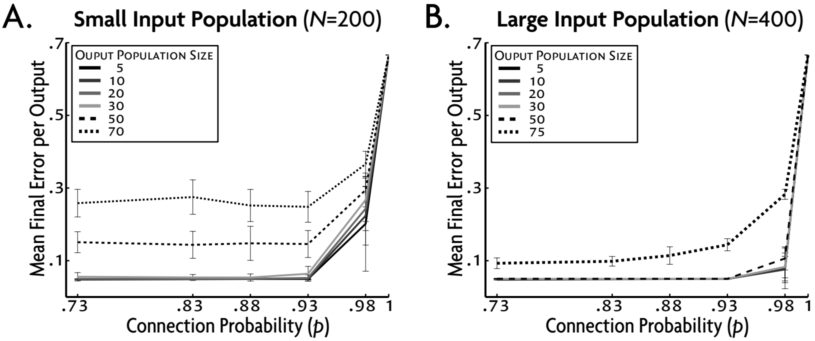

Figure 8: Error as a function of network connection probability for different numbers of output units. The functions being approximated are cosines with phases ranging from 0 to 2π, equally spaced between the output units. (A) A network with 200 input units. Error increases rapidly for $p > 0.93$ regardless of the number of output units. For 50 and 70 output units, the error is larger for all p values. (B) For a population of 400 input units, the results are similar, except that overall performance only degrades when there are 75 output units.

approximation, matching of the target function with these synaptic weights requires that

$$\alpha \sum_{i=1}^{N} r_i(\theta) r_i(\theta') = \delta(\theta - \theta') \sum_{j=0}^{N} \langle F_a r_j \rangle \, . \tag{4.6}$$

Subject to the same constraints on the approximation of the δ function, these equations represent M constraints that need to be satisfied by appropriate adjustment of the $2N$ shift and gain variables of the input units, which should be possible to satisfy when $M < N$. The key to making the combined response modulation synaptic plasticity scheme work is that Hebbian modification reduces the problem of setting pNM synaptic weights to the problem of satisfying the M constraints appearing above, and this can be done by the supervision through its control of the $2N$ shift and gain variables.

4.3 A Stochastic Supervisor. The supervisor used in the simulations discussed thus far employed a gradient-descent algorithm to modify intrinsic response properties on the basis of the error generated by each stimulus. A biological supervisor circuit is more likely to operate under a reinforcement-based scheme. As a first attempt at constructing such a supervisor, we have implemented a model using a stochastic search guided only by a reward signal that reflects network performance. Related ideas have been applied to the supervision of synaptic plasticity (Barto, Sutton, & Anderson, 1983; Mazzoni, Anderson, & Jordan, 1991; Jabri & Flower, 1992; Williams, 1992;

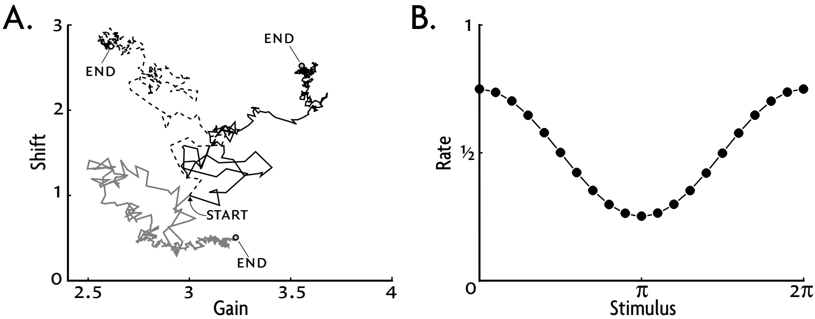

Figure 9: Learning under the random walk supervisor. (A) The paths in modulation space of three input units controlled by the supervisor. All three units started with the same shifts and gains (marked Start), but these then diverged as the supervisor found values that accomplished the function approximation task (End). (B) The end result is a good approximation (dots) of the target function (line) by the output unit as a function of stimulus value. In this simulation, 230 input units drove a single output unit.

Cauwnberghs, 1993; Doya & Sejnowski, 1995; O'Reilly, 1996; Xie & Seung, 2004; Seung, 2003).

For stochastic reward-based supervision, two N-dimensional "modification" vectors, v^s and v^g, of unit length were generated randomly—one for shifts and one for gains. For all i values, the shift and gain of unit i was incremented by an amount proportional to component i of the appropriate modification vector,

$$s_i \rightarrow s_i + v_i^s \quad \text{and} \quad g_i \rightarrow g_i + v_i^g \,. \tag{4.7}$$

Simulations were divided into epochs of 20 stimulus presentations and error evaluations. After each epoch, the sum of the 20 errors was compared to the summed error from the previous epoch. If this total error was less than it was previously, the modification vectors were left unchanged. If the summed error increased from the previous epoch, new modification vectors, v^s and v^g, were generated randomly. In either case, the resulting modification vector was then used to further increment the shifts and gains, as described above. In this study of random walk learning through response modulation, we do not include any Hebbian synaptic modification.

This strategy has the effect of steadily, although slowly, reducing the average error. The paths through modulation space of three input units over the course of a run are plotted in Figure 9A. The improvement in performance can be seen in the reduction in the lengths of the line segments seen in the traces. At the beginning of the run, most of these segments are relatively long as the network makes coarse adjustments to approach the target function. Later, more frequent trajectory changes appear as the network approaches a solution and makes fine adjustments.

Figure 9B illustrates that this crude strategy is capable of solving the task, given enough time (in this case 600 iterations). Thus, a random walk supervisor strategy that requires much less information and algorithmic sophistication than gradient descent can, at least in simple cases, provide adequate supervision.

There are clearly severe limitations on the sizes of networks that can be trained by this random walk algorithm. As the network grows in size, the algorithm gets prohibitively slow. In section 5, we propose ways that this problem might be addressed to achieve better scaling of performance with network size.

5 Discussion

The novel feature of the supervised learning scheme we have proposed is that supervision takes place at the level of neuronal responsiveness rather than synaptic plasticity. Two apparent disadvantages of this scheme—that it does not lead to permanent network modification and that it severely limits the number of elements being supervised—appear to be far less severe than might have been imagined at first. By guiding synaptic plasticity that is otherwise unsupervised, supervised response modulation can lead to permanent changes that allow a network to operate effectively, even in the absence of supervision. Furthermore, Hebbian plasticity can take advantage of small inhomogeneities in randomly coupled networks to allow independent changes in synapses to output units that share presynaptic input.

Given that it works, there are some potential advantages of supervising neuronal excitability rather than synaptic plasticity. First, supervision can occur through ordinary feedback projections that can act rapidly and can target individual neurons independently. Additional advantages concern the nature of the supervisory circuit. We have not attempted to construct a realistic model of this circuit, but we envision it as a network capable of maintaining a continuum of stable, self-sustained patterns of activity (Compte, Brunel, Goldman-Rakic, & Wang, 2000; Seung, Lee, Reis, & Tank, 2000). Such networks tend to drift, especially if provided with noisy input. Thus, it might be possible to implement the random walk supervisor as a network with self-sustained activity and random drift, with the rate of drift controlled by noisy inputs that are suppressed by reward.

Whatever the form of the supervisory circuit, modulating neuronal responsiveness instead of synaptic plasticity has a number of tactical advantages. We considered two approaches to supervision: gradient descent, which involves more information and mathematical analysis than we would expect from a neural circuit, and a random walk model that uses less. A real circuit should lie somewhere between these extremes. From the point of view of the supervisor, the fact that there are far fewer neurons than synapses to supervise changes from a disadvantage to an advantage. The supervisor must search in the space of the parameters it is modifying for a solution to the problem at hand. By reducing the dimension of this space, supervision of neuronal responses, provided that it works (and we have shown that it does), is far easier than supervision of more numerous synapses.

Another advantage of supervising neuronal responses is that the supervisor can monitor the activities that it is modulating in a way that is impossible with supervised synaptic plasticity. It is almost inevitable that the function approximation network, which receives input from the supervisor circuit, would also send projections to it. Such reciprocal connectivity is a typical feature of neuroanatomy. These projections allow the supervisor to monitor the activity of the units it is supervising and use this information to guide learning. For example, this information could be used to reduce the dimension of the space in which the supervisor must search for solutions of the task being learned. Consider, for example, two input units in the function approximation network that have almost totally overlapping response profiles. It is rather wasteful for the supervisor to vary the response properties of these two neurons independently, and yet this is what was done in the random walk model we studied. A more "intelligent" supervisor would use information about the correlations between the units it is modulating to find strategies that are most likely to produce large changes in the network being supervised, and to avoid wasting time generating modulations that have little effect. Thus, projections from the supervised units to the supervisor could be part of a secondary modulatory process that allows the supervisor to learn about learning.

Acknowledgments

This research was supported by the National Science Foundation (IBN-0235463) and NSF-DGE-9972756 and the Sloan-Swartz Center for Theoretical Neurobiology at Brandeis University.

References

Barto, A. B., Sutton, R. S., & Anderson, C. W. (1983). Neuron-like adaptive elements that can solve difficult learning control problems.*IEEE Trans. on Systems, Man and Cybernetics, 13*, 834–846.

Cauwnberghs, G. (1993). A fast stochastic error-descent algorithm for supervised learning and optimization. In Col. Giles, S. J. Hanson, & J. D. Cowan (Eds.), *Advances in neural information processing stystems, 5* (pp. 244–251). San Mateo, CA: Morgan Kaufmann.

Chance, F. S., Abbott, L. F., & Reyes, A. D. (2002). Gain modulation through background synaptic input. *Neuron, 35*, 773–782.

Chauvin, Y., & Rumelhart, D. E., (Eds.) (1995). *Back propagation: Theory, architectures, and applications*. Hillsdale, NJ: Erlbaum.

Compte, A., Brunel, N., Goldman-Rakic, P. S., & Wang X. J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cereb. Cortex, 10*, 910–923.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: Computational and mathematical modeling of neural Systems*. Cambridge, MA: MIT Press.

Doiron, B., Longtin, A., Berman, N., & Maler, L. (2001). Subtractive and divisive inhibition: Effect of voltage-dependent inhibitory conductances and noise. *Neural Comput., 13*, 227–248.

Doya, K., & Sejnowski, T. J. (1995). A novel reinforcement model of birdsong vocalization learning. In G. Tesauro, D. Touretzky, & T. Leen (Eds.) *Advances in nueral information processing systems, 7* (pp. 101–108) Cambridge, MA: MIT Press.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Redwood City, CA: Addison-Wesley.

Jabri, M., & Flower, B. (1992). Weight perturbation—an optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks. *IEEE Trans. on Neural Networks, 3*, 154–157.

Lukashin, A. V., Wilcox, G. L., & Georgopoulos A. P. (1994). Overlapping neural networks for multiple motor engrams. *Proc. Natl. Acad. Sci. U.S.A., 9*, 8651–8654.

Mazzoni, P., Andersen R. A., & Jordan M. I. (1991). A more biologically plausible learning rule for neural networks. *Proc. Natl. Acad. Sci. U.S.A., 88*, 4433–4437.

Miller, K. D. (1996). Receptive fields and maps in the visual cortex: Models of ocular dominance and orientation columns. In E. Domany, J. L. van Hemmen, & K. Schulten (Eds.), *Models of neural networks* (Vol. 3, pp. 55–78). New York: Springer-Verlag.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation, 8*, 895–938.

Poggio, T. (1990). A theory of how the brain might work. *Cold Spring Harbor Symposium on Quantitative Biology, 55*, 899–910.

Prescott, S. A., & De Koninck, Y. (2003). Gain control of firing rate by shunting inhibition: Roles of synaptic noise and dendritic saturation. *Proc. Natl. Acad. Sci. U.S.A., 100*, 2076–2081.

Salinas, E., & Abbott, L. F. (2000). Do simple cells in primary visual cortex form a tight frame? *Neural Comp., 12*, 313–336.

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science, 275*, 1593–1599.

Seung, S. (2003). Learning in spiking neural networks by reinforcement of stochastic synapatic transmission. *Neuron, 40*, 1063–1073.

Seung, H. S., Lee, D. D., Reis, B. Y., Tank, D. W. (2000). Stability of the memory of eye position in a recurrent network of conductance-based model neurons, *Neuron 26*, 259–271.

Widrow, B., & Hoff, M. E. (1960) Adaptive switching circuits. *WESCON Convention Report, 4*, 96–104.

Widrow, B., & Stearns, S. D. (1985). *Adaptive signal processing*. Englewood Cliffs, NJ: Prentice Hall.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning, 8*, 229–256.

Xie, X., & Seung, S. (2004). Learning in neural networks by reinforcement of irregular spiking. *Physical Review, 69*, 041909.

Received October 2, 2003; accepted July 28, 2004.