Control of Network Activity through Neuronal Response Modulation

Christian D. Swinehart, Kris Bouchard, Peretz Partensky, & L.F. Abbott
Volen Center for Complex Systems & Dept. of Biology,
Brandeis University, MS 13
Waltham, MA, 02454

Keywords: Neural Networks; Learning; Modulation; Supervision;

Abstract

Neural network learning is typically treated as the problem of setting synaptic connection strengths to better perform a task. This requires supervision. However, anatomical data suggest that direct synaptic modification by such a supervisor circuit would be unfeasible. We investigate supervision at the level of neurons rather than synapses. By modulating the response properties of cells in the network, this form of supervised learning is able to successfully train a network to perform a function approximation task. We examine the nature of this modulation, and consider its implications for supervised learning.

1 Introduction

Neural networks that perform a specific task must be developed through a learning procedure which reconfigures the network in a way that maps patterns of input to a desired output. A standard mechanism for inducing such a correct mapping is the use of supervision. The supervisor can be thought of as a neural circuit that monitors the network's success in performing its task, and, by adjusting the properties of the network, acts to minimize the difference between the desired and actual output.

Traditionally, the supervisor receives an error signal and adjusts the network by directly guiding synaptic plasticity. This approach has proven widely successful in training networks to perform a variety of tasks [9, 2, 4, 3]. However, there is little anatomical evidence that such a scheme could work under a biological implementation. In particular, this direct synaptic modification model presumes the existence of a large number of backprojections from the supervisor terminating on the synapses in the network. While massive feedback projections are characteristic of biological nervous systems, the proposed three-element synapses—though they do exists—are too rare for this mechanism to explain network learning on this scale.

Instead, we propose a supervision scheme in which adjustments are made not to synapses, but to the neurons themselves. This connectivity pattern would be consistent with known neuroanatomy, consisting of feedback connections from the supervisor to the neurons themselves. Additionally, these connections could be ordinary excitatory and inhibitory inputs [1], allowing for both the speed and input specificity necessary for rapid learning.

The question we address here is whether such a scheme can work. We first survey the range of network functions that can be learned based on this type of supervision. Subsequently, we examine the effects of the supervisor's modulatory input, and discuss the problem of learning within the search space where the supervisor operates.

2 Methods

We apply our proposed supervision scheme to the problem of function approximation, a classical neural network task with similarities to those performed by biological circuits [6]. The objective is to elicit a network output that is a specified function of a single stimulus variable, θ. A traditional supervision scheme would bring this about by modifying synaptic weights according to a synaptic learning scheme such as the delta rule [8], in which errors in the output approximation are used to strengthen or weaken connections based on their relative contributions to those errors. Here, a similar approach is used, but rather than adjusting synaptic weights, the supervisor modulates the response properties of individual network neurons.

Our model network consists of a two-layer feedforward architecture containing purely excitatory connections, with N input neurons projecting to a single output neuron (see figure 1). The input units are driven by currents that are Gaussian functions of the difference between a stimulus, θ, and a preferred stimulus θ_i. Preferred stimulus values are uniformly distributed across the stimulus space for the different input units.

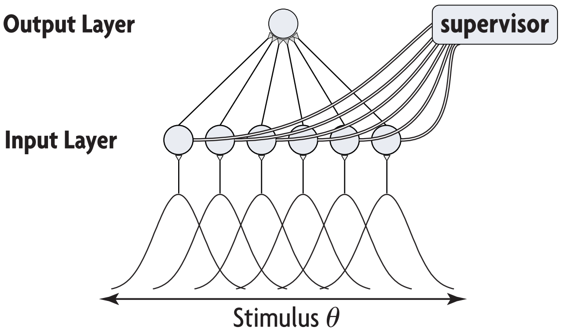

Figure 1: Network architecture. 460 input neurons receiving stimulus-tuned input currents project to a single output unit. An external supervisor is connected to all input units and modulates their response properties independently.

Each input unit's firing rate is calculated by passing its stimulus-tuned input current through a sigmoidal transfer function,

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

The parameter s_i, which we call the shift, controls the value of I_i at which r_i reaches half its maximal value, which has been set to the value 1. The parameter g_i, which we call the gain, determines the slope of the firing rate versus input curve at this point. These are set to the initial values $s_i = 1.0$ and $g_i = 3.0$ for all units, but are then changed by the supervisor.

The network's output consists of the firing rate of the single unit in the top layer of figure 1. This rate is also determined by the sigmoidal function in equation 1, but the input factor is instead a sum of the firing rates of the input layer units,

$$R(\theta) = \frac{1}{1 + \exp\left(-g_{out}\left(\sum_{i=1}^{N} r_i(\theta) - s_{out}\right)\right)} \,. \tag{2}$$

The goal of learning for this network is to match the output firing rate, $R(\theta)$, to an arbitrary set of stimulus-dependent target functions, $F(\theta)$, as closely as possible.

2.1 Neuronal Response Modulation

Instead of following the classical approach and having supervision occur through changes in the synaptic weights, our form of supervision involves changes in the shift and gain parameters (s_i and g_i) to each of the individual input cells, thereby controlling their response properties. By contrast, all weights are kept fixed at 1.0 throughout the simulation.

Performance error is calculated by comparing the output firing rate to the target function for each stimulus value. The supervisor then uses a stochastic gradient descent algorithm to modify the shifts and gains of the input units in order to reduce the error,

$$E(\theta) = \frac{1}{2}\left(R(\theta) - F(\theta)\right)^2 . \tag{3}$$

For each stimulus presentation, a random θ value is chosen in the range from 0 to 2π. The rates of the input and output cells are then computed and shifts and gains for the input cells are modified such that

$$s_i \rightarrow s_i - \epsilon\frac{\partial E(\theta)}{\partial s_i} \quad \text{and} \quad g_i \rightarrow g_i - \epsilon\frac{\partial E(\theta)}{\partial g_i} \,, \tag{4}$$

where ϵ is a small rate factor which constrains the response modulation. In our simulations, ϵ was fixed at 0.2.

3 Results

3.1 Range of Performance

We begin our analysis by showing that function approximation can be achieved in the absence of synaptic plasticity solely through changes to the input cells' response properties. Using our supervision regime, the network can approximate a wide variety of target functions, as depicted in figure 2. Distribution of shift and gain values for the modulated cells are plotted in the left column, and the corresponding target function and network approximation in the right column. In the center are the modulated firing rates of the input layer cells with respect to the stimulus.

In general, the network can approximate any smooth, continuous target function it is given. However, since the modulated firing rates of the input layer cells represent a basis set for the output approximation, problems arise when these response profiles cannot capture important features of the target function. This limitation can be seen in figure 2c where rapid changes in the target function cannot be reproduced by the network due to the Gaussian shape of the input unit response tunings.

It is worth noting that in its attempts to deal with the discontinuity, the supervisor has increased the gain of the cells whose response boundaries border the vertical portion of the target function in an attempt to match that contour.

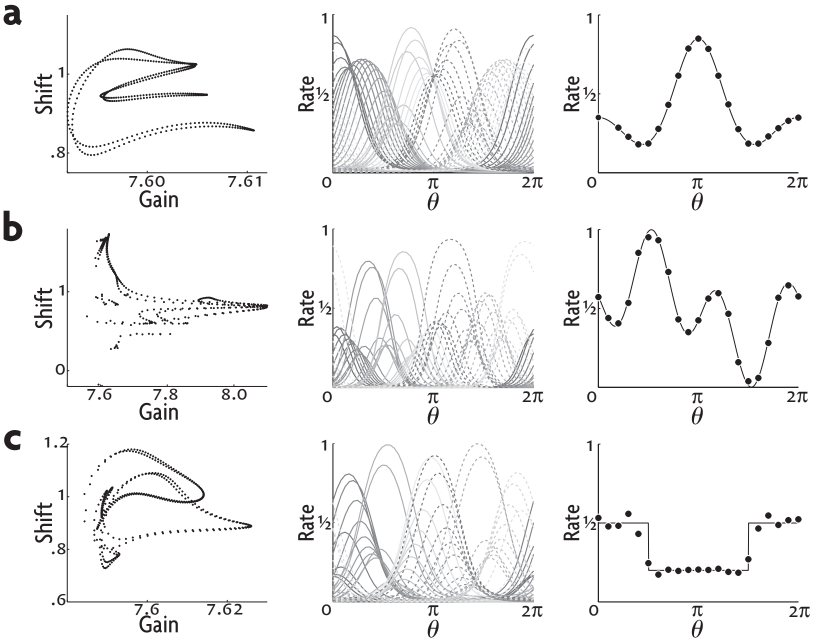

Figure 2: Examples of learning via response modulation for a variety of target functions. Plots in the left column depict the modulation state of the network, with each dot corresponding to the final shift and gain values for each input unit. The center column shows the responses of a sampling of input units to a range of stimulus values. The right column plots the network's approximation (dots) of the target function (line) for a selection of stimuli.

3.2 Shift vs. Gain

Following the observation that gain modulation was used to mitigate the approximation failure in the case of the square-wave target function, we examined the relative uses of shift and gain in the learning process. Figure 3 illustrates the result of learning by modulating shift alone, gain alone, or both in tandem. Figure 3a considers a low-frequency cosine target function, while figure 3b considers a high-frequency target.

In the low-frequency case, learning succeeds regardless of which parameters the supervisor was free to modulate. The top row shows the optimal case in which the supervisor relies approximately evenly on shift and gain, with shift contributing to the overall level of activity, and gain controlling the flatness of the response in the peak and trough and its steepness along the sloped portions. This reliance upon shift for setting the overall vertical level can be seen indirectly in the second row of figure 3a, corresponding to the gain-alone case. Here the shifts remain fixed, but a similar effect can be emulated by flattening the gain profiles of a few inputs which will in turn contribute a stimulus-independent baseline shift.

In comparing the target approximation in the right column of figure 3a, the results from learning with gain alone are virtually unchanged as compared to the shift-alone and shift-plus-gain con-

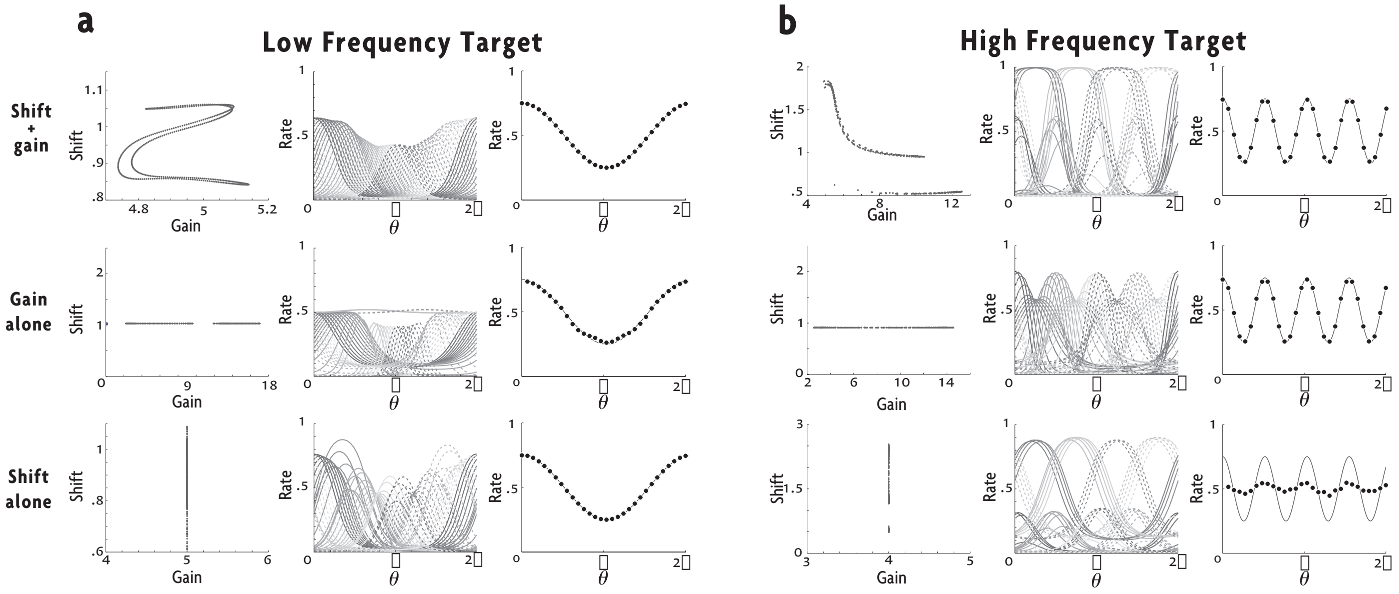

Figure 3: Relative contributions to learning of shift- and gain-based modulation under two stimulus regimes. As with figure 2, the columns depict (from left to right), the network modulation state, modulated firing rates of input cells, and the network's final approximation of the target function.

ditions. However, in the high-frequency case of figure 3b the differences between shift and gain become stark.

Again the top column of figure 3b shows that the function is readily matched by the supervisor when both shift and gain are adjusted. But whereas in the low-frequency case, learning was successful with only shift modulation, here it fails miserably. This failure is due to the breadth of the Gaussian inputs being wider than the half-period of the target function. Thus in order to properly approximate the height of the peaks, inputs must be shifted up, but in the process this causes the depth of the troughs to be overestimated as the Gaussian tails spill off to the sides. As a result, the ultimate approximation splits the difference between peaks and troughs by lying along the target function's mean value, with only small deflections toward the extremes.

It is clear that to successfully approximate this function, the breadth of the response profiles must be lessened, and this is precisely the result of learning in the gain-alone condition (middle row of figure 3b). In this case the network learns successfully even in the absence of shift modulation by boosting the gain of inputs with preferred stimuli near the peaks, and modestly decreasing those preferring the troughs.

3.3 Uniqueness

Learning is made easier in these networks by the fact that the set of shift and gain variables that leads to successful function approximation is not, in general, unique. An illustration that this is indeed the case can be seen in figure 4, in which the final network modulation states of two separate runs are plotted. In both cases the target function was the same, however in the first run the input cells were given initial values of $s_i = 0.91$ and $g_i = 5.0$, and in the second run $s_i = 1.14$ and $g_i = 8.4$. In both cases the supervisor was able to approximate the target perfectly, yet none of the modulation states in the two runs is the same. This suggests that the problem, from the supervisor's perspective, may be simpler than one would guess. Perhaps it is the overall balance of shifts and gains that is significant, and not simply finding a single, globally optimal solution.

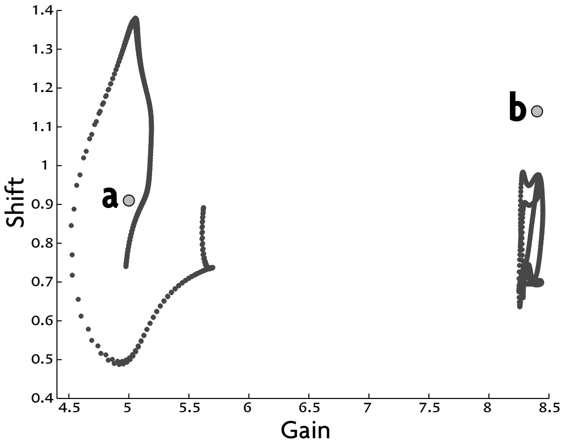

Figure 4: Function approximation solutions are not unique. The two clouds depict the final shift and gain states of the input units after learning from two different starting conditions. The cloud on the left corresponds to a starting position of (a), and the cloud on the right to starting position of (b). The target function in both cases was identical.

3.4 Interpolation

It is important to remember that the supervisor's control of the network corresponds to a pattern of excitatory and inhibitory input that it sends to the network. As a result, the same network could conceivably be *switched* [5] between different target functions simply by the supervisor changing this input pattern. This raises the possibility of the supervisor building up a repertoire of learned states over the course of learning, and then applying those as appropriate to the task at hand. It would be particularly useful if there were some consistent mapping between related modulation states and related target functions.

In fact this relationship seems to exist. Figure 5 shows the result of interpolating between learned states to yield an intermediate output. Figures 5a and 5b show the result of learning two sinusoids phase shifted by $\pi/2$. For figure 5c, no learning occurred. Instead, a set of shifts and gains was constructed by taking the vector average of the sets in the previous, learned trials. The resulting distribution produces an output quite close to the midpoint (in phase) between the learned functions.

This suggests an additional simplification of the supervisor's task since every new target function need not be learned from scratch. Instead, by consulting a library of previously learned approximations, the supervisor can start with a similar pattern and fine-tune the approximation from there. This result also places a limit on what could otherwise be unsustainable growth in the size of the library over time. Since similar functions yield similar network modulation patterns, it is not necessary to exhaustively store every single pattern experienced. Instead, a pared-down set of characteristic functions can be maintained, each as a basis for an entire group of targets rather than just

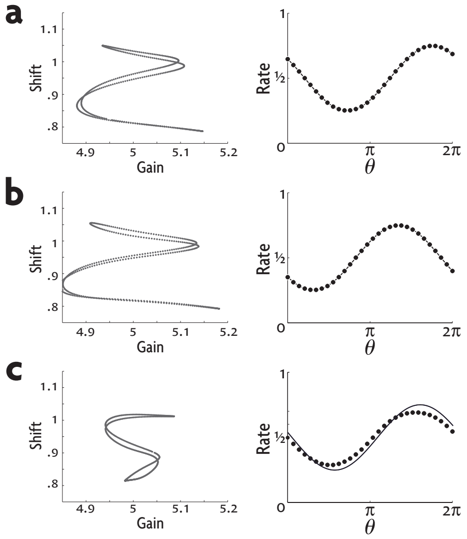

Figure 5: Interpolating between learned modulation states yields an interpolated output function. After learning to approximate two sinusoids shifted in phase (a and b), the resulting modulation states were averaged (c, left panel), resulting in an output function corresponding to a sinusoid with an intermediate phase (c, right panel).

a single one.

4 Conclusions

Though we have demonstrated that through its modulatory input the supervisor can put the network into an activity state suitable for the task, it should be noted that this does not result in a permanent change to the network itself. In the absence of modulation the network's response to stimuli will be identical to its response before 'learning' took place. For permanent changes that do not depend on the action of the supervisor, a form of synaptic plasticity is necessary.

In another publication [7] we have examined similar networks in which an unsupervised, Hebbian learning rule governing plasticity works in concert with supervised response modulation. This results in a pattern of learning similar to that discussed here, with the exception that learning is gradually transferred from the modulation pattern to the synaptic weights. Once this transfer is complete, the supervisory modulation may be removed altogether without affecting the network's performance.

5 Acknowledgments

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

References

[1] Chance, F.S., Abbott, L.F., Reyes, AD, Gain Modulation Through Background Synaptic Input. Neuron. 35 (2002) 773–782.

[2] Chauvin, Y., Rumelhart, D.E., eds., Back Propagation: Theory, Architectures, and Applications. (Erlbaum, Hillsdale, NJ, 1995).

[3] Dayan, P. and Abbott, L.F., Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. (MIT Press, Cambridge, MA, 2001).

[4] Hertz, J., Krogh, A., Palmer, R.G. Introduction to the Theory of Neural Computation. (Addison-Wesley, Redwood City, CA, 1991).

[5] Lukashin A.V., Wilcox G.L., Georgopoulos A.P., Overlapping neural networks for multiple motor engrams. Proc Natl Acad Sci USA. 9 (1994) 8651–8654.

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

[7] Swinehart, C.D. & Abbott, L.F. Supervised learning through neuronal response modulation. (Submitted, 2003).

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

[9] Widrow, B., Stearns, S.D. Adaptive Signal Processing. (Prentice-Hall, Englewood Cliffs, NJ, 1985).