

FUNCTIONAL SIGNIFICANCE OF LONG-TERM POTENTIATION FOR SEQUENCE LEARNING AND PREDICTION

L.F. Abbott and Kenneth I. Blum*

Center for Complex Systems
Brandeis University
Waltham, MA 02254

Abstract

Population coding, where neurons with broad and overlapping firing rate tuning curves collectively encode information about a stimulus, is a common feature of sensory systems. We use decoding methods and measured properties of NMDA-mediated LTP induction to study the impact of long-term potentiation of synapses between the neurons of such a coding array. We find that, due to a temporal asymmetry in the induction of NMDA-mediated LTP, firing patterns in a neuronal array that initially represent the current value of a sensory input will, after training, provide an experienced-based prediction of that input instead. We compute how this prediction arises from and depends on the training experience. We also show how the encoded prediction can be used to generate learned motor sequences, such as the movement of a limb. This involves a novel form of memory recall that is driven by the motor response so that it automatically generates new information at a rate appropriate for the task being performed.

Published in *Cerebral Cortex* **6**:406-416 (1996).

Introduction

Although it is widely believed that long-term potentiation (LTP) plays a central role in learning and memory, the specific mechanisms by which changes in synaptic efficacy affect behavior are unknown. Synaptic modification rules inferred from experiments on LTP suggest that the pattern of synaptic potentiation produced by training experience reflects correlations in the firing patterns of neurons. How is the information stored in these synaptic enhancements read out to affect future behavior? A much-studied conjecture is that the stored information is encoded in the firing patterns of fixed-point attractors of the neural network dynamics (Marr, 1971; Hopfield, 1982 & 1984; Grossberg, 1988; Kohonen, 1984 & 1988; Rolls, 1989; Amit, 1989; Hertz *et al.*, 1990). In these models memory recall is associative and quasi-static. We provide another suggestion: that long-term potentiation allows a neural network to generate continuously from sensory input data an experience-based prediction of the future value of an encoded quantity (Levy, 1989; Muller & Kubie, 1989). Our proposal relies on two basic features of neuronal circuitry: 1) Ensemble coding of sensory information by arrays of neurons with broadly tuned average-firing rate curves and 2) Long-term potentiation of synapses between encoding neurons that is temporally asymmetric. We find that the temporal properties of NMDA-mediated LTP naturally cause a population of neurons that represent a quantity to start predicting that quantity after training. In addition to providing a model for experience-based prediction, our work suggests a novel mechanism for learning and generating sequences of motor actions.

Ensemble coding of sensory information by large arrays of neurons is extremely widespread (Knudsen *et al.*, 1987; Konishi, 1987; Sejnowski, 1988; Eichenbaum, 1993). Examples arise in all sensory modalities (Orban, 1984; Maunsell & Newsome, 1987; Steinmetz *et al.*, 1987; Kalaska *et al.*, 1983; Knudsen & Konishi, 1978; Konishi, 1991; O'Neill & Suga, 1982) and in multimodal, highly processed information such as spatial position represented by hippocampal place cells (O'Keefe & Dostovsky, 1971; O'Keefe & Nadel, 1978; O'Keefe, 1979). In many cases it has proven possible to decode the information represented by neural firing in such arrays (for reviews see Salinas and Abbott, 1994; Abbott, 1994) so that an estimate of the encoded quantity can be computed from measured firing rates. Georgopoulos and collaborators developed one such approach by defining the population vector relating activity in the motor cortex of the monkey to the direction of arm movements (Georgopoulos *et al.*, 1986 & 1988). Wilson and McNaughton (1993) have decoded the output of place cells in the hippocampus of the rat to determine the spatial location represented by their firing.

The ability to decode the information represented by neuronal ensemble firing is a key element of our work. We use decoding only as a tool and do not require that the decoding methods employed resemble mechanisms actually operating in the nervous system. We do, however, assume that the information obtained from a mathematical decoding procedure provides an accurate measure of the information represented by an array of neurons and that this information is available to downstream neural networks (Salinas & Abbott, 1995). We model how the value of an ensemble-coded quantity is affected by potentiation of recurrent collateral synapses between neurons within the coding array. Our computational procedure is to decode place cell activity before and after a simulated training period. The synaptic changes that occur during the training period are computed using characteristics of NMDA-mediated LTP observed in hippocampal slice preparations (Bliss & Collingridge, 1993; Malenka & Nicoll, 1993; Levy and Steward, 1983; Gustafsson *et al.*, 1987). We then calculate how these training-induced changes of synaptic strength alter network firing, and

decode the modified firing patterns. We find that training experience produces a shift in the information encoded by ensemble firing activity that has clear behavioral implications. Our calculations provide specific predictions that can be tested experimentally.

Since our results are based on general properties of LTP and on a common form of population coding, they are applicable to a wide variety of neural systems. Examples to which our ideas could be applied include neurons responding to the position or velocity of a limb (Kalaska *et al.*, 1983), neurons characterizing the motion of a visual image (Maunsell & Newsome, 1987; Steinmetz *et al.*, 1987) or location of a sound source (Knudsen & Konishi, 1978; Konishi, 1991) or hippocampal place cells representing spatial position within the environment (O’Keefe & Dostovsky, 1971; O’Keefe & Nadel, 1978; O’Keefe, 1979). To keep the discussion general, we initially consider an array of neurons responding to a sensory input without specifying a particular neural circuit or sensory modality. We use the term ‘sensory input’ rather than ‘stimulus’ so that cases like proprioceptive information from a moving limb or hippocampal-encoded spatial location are included along with more conventional sensory stimuli. However, we also consider specific applications and systems. In a separate paper, we have shown how LTP between hippocampal place cells can lead to a shift in the position coded by hippocampal ensemble activity that can be used to guide navigation in tasks like the Morris water maze. (Blum & Abbott, 1996). Here, after the general discussion, we consider the application of our ideas to the learning of motor sequences, for example, a learned movement of an arm or other limb. In this case, the sensory input being discussed is the position or velocity of the limb.

Methods

We consider an array of neurons that collectively encode information about a sensory input. Since we are interested in how experience changes coded information, we focus exclusively on experience-dependent modifications in synaptic weights and their effect on coding. As a result, we do not attempt to model how coding arises in the neuronal array. For example, the firing rate tuning curves of individual neurons are used directly in the model and no attempt is made to explain how these tuning curves arise. Tuned neuronal responses may be produced by the pattern of afferent synaptic weights or the structure of the connections between coding neurons. Our methods can handle either case or any combination of these two mechanisms.

We characterize the sensory input to the coding array by a vector \vec{x} that could be a two- or three-dimensional spatial vector representing direction of motion or spatial position or an arbitrary collection of numbers that characterize other attributes of a sensory input. These could include a large number of quite abstract features. Collecting these together in a vector form is merely a notational convenience.

Table of Symbols

The important variables and functions used in our discussion are:

r_i = the firing rate of neuron i in an array of neurons responding to some sensory input.
 \vec{x} = a set of numbers characterizing relevant properties of the sensory input driving the array.
 $f_i(\vec{x})$ = the average firing rate of neuron i , before training, when the sensory input is equal to \vec{x} .

\vec{s}_i = the sensory input that produces the maximum average firing rate in neuron i .
 \vec{p} = the population vector which is a set of numbers computed from the firing rates of the network neurons that identify the information represented by a particular pattern of firing.
 $\vec{X}(t)$ = a particular sequence of sensory inputs presented during training.
 ΔW_{ij} = the change in synaptic efficacy between presynaptic neuron j and postsynaptic neuron i arising from training.

We use two symbols, r_i and f_i , to denote the firing rate of neuron i . f_i refers to the average firing rate before training has modified synaptic weights. By definition, f_i is not changed by the training experience. We use r_i to denote the firing rate both before and after training so this rate is modified by experience. r_i is equal to f_i before training but different from it afterward. The population vector \vec{p} characterizes the neuronal response to a sensory input while the vector \vec{x} is a mathematical characterization of that input independent of the neuronal response. \vec{p} depends on \vec{x} but is not necessarily equal to it. In fact, the key to sequence learning and prediction in our model is that training experience causes \vec{p} to differ from \vec{x} .

Decoding

Decoding is a useful technique for determining the physical and behavioral significance attached to a particular pattern of network firing. In order to determine what information is being represented by a neuronal array, we use two different population decoding methods (Salinas & Abbott, 1994; Abbott, 1994). These methods construct a population vector \vec{p} that characterizes the information represented by ensemble firing activity in a manner similar to the way that the vector \vec{x} characterizes the sensory input. One way we compute the population vector is a variant of the method developed in the study of monkey motor cortex (Georgopoulos *et al.*, 1986). This approach has been applied successfully in a number of cases to a variety of systems (Georgopoulos *et al.*, 1986, 1988 & 1993; Caminiti *et al.*, 1991; Kalaska *et al.*, 1983; Fortier *et al.*, 1989; Gilbert & Wiesel, 1990; Steinmetz *et al.*, 1987; Young & Yamane, 1992; Salinas and Abbott, 1994). In this case, the population vector \vec{p} corresponding to a particular set of firing rates r_i is the ‘center of gravity’ of the neuronal firing pattern given by

$$\vec{p} = \frac{\sum_i r_i \vec{s}_i}{\sum_i r_i} . \quad (2.1)$$

Here \vec{s}_i is the value of the sensory input vector that produces the maximum response in neuron i . To check that our results do not depend on any particular decoding scheme, we also use another procedure that is related to maximum likelihood estimation (Salinas & Abbott, 1994). This is the least-squares method where \vec{p} is defined as the vector giving the best fit of the average firing rates f_i to the observed rates r_i so that

$$\sum_i [r_i - f_i(\vec{p})]^2 = \text{minimum} . \quad (2.2)$$

Both methods give similar, though not identical, results.

Synaptic Modification Rule

Our basic program is to determine how synaptic modifications induced by training experience affect the population vector that is ensemble-coded by network firing. To do this we need to model how training modifies synapses. As mentioned previously, we are concerned with experience-induced changes in synaptic weights rather than with the values of the weights themselves. An advantage of this approach is that we can consider small shifts in synaptic weights as a perturbation on the basic neuronal response characterized by firing rate tuning curves (Abbott, 1994). This allows us to compute the effects of training-induced LTP without making large numbers of assumptions. Our calculations are quantitatively accurate only when the amount of LTP is fairly small, although we expect our results to be qualitatively correct even for larger changes in synaptic weights. Because the mechanisms we consider are collective, even small changes in synaptic weights can produce behaviorally relevant effects. We associate the synaptic weights we are discussing with individual synapses but a more abstract interpretation in terms of ‘effective’ synaptic strengths is also possible.

Suppose that during the training period the sensory input takes a specific sequence of values given by $\vec{X}(t)$. This could represent the presentation of a temporal sequence of sensory stimuli with specific attributes, for example the movement of a limb along a specific trajectory for proprioceptive coding. The training sequence may include repetitions. If the average-firing rate tuning curves f_i have been measured we can predict how each neuron will fire (on average) during the training period,

$$r_i(t) = f_i(\vec{X}(t)) . \quad (2.3)$$

To compute the synaptic modifications caused by training we use a standard model of long-term potentiation in which the increase of synaptic strength is proportional to the product of the average firing rates of the pre- and postsynaptic neurons (Hebb, 1949; Sejnowski, 1977; Linsker, 1986; Abbott, 1990 & 1994; Levy *et al.*, 1990; Miller, 1990, 1992 & 1994; Bienenstock *et al.*, 1992). However, we include a temporal asymmetry with respect to the pre- and postsynaptic order of firing that plays an essential role in our model. NMDA-mediated LTP induction (Bliss & Collingridge, 1993; Malenka & Nicoll, 1993), as measured in hippocampal slices, requires presynaptic activity in conjunction with either simultaneous or somewhat later, but not earlier, postsynaptic activity (Levy and Steward, 1983; Gustafsson *et al.*, 1987). If postsynaptic activity precedes presynaptic activity, synapses may even be depressed (Levy and Steward, 1983; Debanne *et al.*, 1994). Our model is not sensitive to the exact details of this temporal asymmetry as long as presynaptic firing occurring before postsynaptic firing produces more LTP than the reverse time order. To incorporate this temporal asymmetry we include a factor $H(t')$ that represents the efficiency of LTP induction when presynaptic activity precedes postsynaptic activity by a time t' . If LTP occurs during presentation of the sensory sequence $\vec{X}(t)$, the strength of the synapse from neuron j to neuron i is enhanced in the model by

$$\Delta W_{ij} = \int dt dt' H(t') f_i(\vec{X}(t+t')) f_j(\vec{X}(t)) . \quad (2.4)$$

The t' integral in this equation sums over the all possible time differences between pre- and postsynaptic firing while the t integral sums over the time course of the training period.

Equation (2.4) allows the synaptic strengths to grow without bound. This is clearly unreasonable so some form of constraint should be imposed (Miller & MacKay, 1994). However, we will assume throughout that the total amount of synaptic potentiation is small and then the constraint does not play an important role (see also MacKay & Miller, 1990).

Results

Experience-Based Prediction

We will assume that, before training, the activity of the neuronal array accurately represents the value of the sensory input. Thus, initially the sensory input vector and the population vector are identical, $\vec{p} = \vec{x}$. Indeed, if the tuning curves of the encoding neurons are evenly distributed over the range of coded values, it can be shown that the population vector given by equation (2.1) will agree with the value of the sensory input (Georgopoulos *et al.*, 1988). However, modifications in the strength of synapses between coding neurons produced by training experience can change the population vector characterizing the ensemble response to an input and destroy this equality. At first sight, this might appear to introduce unwanted inaccuracy in the representation of the sensory input, but we will see that the shifts in ensemble-coded quantities caused by experience-based synaptic modification can be useful.

How do the synaptic weight changes arising from training alter the pattern of neuronal firing produced by exposure to an arbitrary sensory input after training has concluded? We assume that potentiation of a synapse increases the firing rate of the postsynaptic neuron by an amount given by the product of the synaptic weight change times the firing rate of the corresponding presynaptic neuron. The total effect for a given postsynaptic neuron is obtained by summing over all of its presynaptic neurons. Thus, after training, the average firing rate of neuron i when the input takes the value \vec{x} is, to linear order in ΔW ,

$$r_i = f_i(\vec{x}) + \sum_j \Delta W_{ij} f_j(\vec{x}) \quad (3.1)$$

rather than $r_i = f_i(\vec{x})$ as it was before training. We now compute how these training-induced changes in the firing rates shift the value of the population vector representing the ensemble firing. We do this by inserting the rates given by equation (3.1) into equation (2.1). To linear order in ΔW we find that, after training, when the sensory input takes the value \vec{x} the responses of the network neurons code for a population vector

$$\vec{p} = \vec{x} + \frac{\sum_{ij} (\vec{s}_i - \vec{x}) \Delta W_{ij} f_j(\vec{x})}{\sum_i f_i(\vec{x})} \quad (3.2)$$

that is different from the sensory input vector \vec{x} . This result is derived in Appendix A. A similar result using the least-squares method of decoding is also given in Appendix A.

What is the significance of the fact that, after training, the coded population vector is no longer equal to the sensory input? Before presenting the results of detailed calculations we will answer this question qualitatively. During training, neurons responding to the training inputs are sequentially activated. Because of the temporal asymmetry of NMDA-mediated LTP induction, synapses from presynaptic cells activated at one point in the training sequence to postsynaptic cells activated simultaneously or somewhat later will be potentiated. Now consider what happens after the training period is over. Suppose that one of the sensory inputs from the training sequence is presented to the network. Through the potentiated synapses, neurons activated by this input will preferentially excite other neurons that are tuned to respond to subsequent sensory inputs in the training sequence. This shifts the ‘center of gravity’ of neuronal firing forward along the training sequence. The result is

that the coded population vector will not be equal to the sensory input vector but rather to a value somewhat ahead of it in the training sequence. Thus, the population vector representing ensemble activity no longer encodes the present value of the sensory input but instead provides a prediction of its future value based on the training experience.

The population vector is also shifted for sensory inputs that are not in the training sequence but are similar to training inputs. As we will see from the more detailed computations, this shift is toward the training set (see also Muller *et al.*, 1991). As a result, the network can interpolate to provide predictions for sensory inputs not in the training set.

The ability of LTP to produce an ensemble-coded prediction on the basis of training experience is seen in figure 1. We consider a two-dimensional sensory input vector so that our results can be plotted and visualized easily. This vector could correspond, for example, to the position or movement direction of a limb. In figure 1, we show an area in the two-dimensional space of sensory inputs. Initially, the response of the coding array to sensory inputs anywhere within this area resulted in a population vector that was equal to the sensory input vector, $\vec{p} = \vec{x}$. We then simulated a particular training sequence. In these plots, a training sequence is represented by a path through the two-dimensional space of sensory inputs. In the case of arm position coding, these paths would represent arm movement trajectories. The solid lines in figure 1 indicate the particular training sequences used. These training sequences induced changes in synaptic efficacy as discussed in the Methods section.

After training, when the network was exposed to a sensory input vector \vec{x} and its response population vector was computed, the result was no longer $\vec{p} = \vec{x}$ but rather, for most \vec{x} values within the area shown in the figures, the population vector shifted relative to \vec{x} . In figure 1, the arrows indicate these shifts, that is, the arrows are equal to the difference $\vec{p} - \vec{x}$. This difference is defined for every value of \vec{x} but we have plotted the arrows on a grid of \vec{x} values. At every point on the grid, the tail of the arrow represents a particular sensory input to the network and the head of the arrow marks the corresponding value of the population vector decoded from the network activity evoked by this input. Examination of the arrows indicates that for sensory inputs lying on the training sequence path, the population vector lies ahead of the sensory input vector, forward along the path. Beside the training path, the population vector is displaced both forward and toward the training sequence relative to the actual sensory input.

Figure 1 implies that a network that initially encoded a sensory input will no longer code for its value after training, but instead will provide a prediction of a future value on the basis of the training experience. This is seen in figure 2. A network was trained using the circular sequence of figure 1b. Then, the same sequence was used as input to the network and the resulting population vector was decoded. In figure 2, the dashed line shows the horizontal component of the sensory input vector as a function of time. The solid line indicates the corresponding component of the population vector. The output of the network, as decoded by the population vector, clearly provides a prediction of the actual sequence. Figure 2b shows a prediction that is more advanced in time, though less accurate, than the one in figure 2a. The network used in figure 2b had more highly overlapping firing tuning curves than those used in figure 2a and this resulted in a longer prediction time (see below).

The arrows in figure 1 collectively provide a map of the space of sensory input values that predicts how sequences of inputs are likely to flow on the basis of the training sequence. It is possible to build up such maps from multiple paths. Figure 1c shows the case of two superimposed paths. Clearly information about both paths is present in the map, although

some confusion can arise at the junction. If many training sequence paths are superimposed, information about individual sequences gets lost but, nevertheless, the map contains information about how the sequences collectively flowed through the space of sensory input values. In our study of hippocampal place cell coding we showed how a large number of training paths build up a map that can be used for navigation (Blum and Abbott, 1996). Here, we will concentrate primarily on single paths.

How large can the training-induced shift in the population vector be? This depends on the rate of LTP induction and degree of its temporal asymmetry, and on the width and degree of overlap of the neuronal tuning curves (see Appendix B). Because the shift along a learned path is approximately tangent to the path, the shifted population vector acts as a linear predictor of the future sensory input vector. Broader, more overlapping tuning curves produce large shifts and consequently longer prediction times. The resulting linear predictions can be significantly longer than the 100-200 ms time window for LTP. This somewhat surprising result is derived in Appendix B. The magnitude of the shift in the coded population vector is proportional to the product of three factors. One factor is the amount of synaptic potentiation that occurs in one LTP window time between two neurons firing at their maximum rates. The second factor is the average width of the firing rate tuning curves and the third is a measure of the amount of overlap between the firing rate tuning curves of the coding neurons. If we want to know how far into the future a neuronal array predicts, we need to divide the shift in the population vector by the speed at which sensory inputs are changing. For slowly evolving input data and broad, highly overlapping neuronal tuning curves, large prediction times are possible. Note, however, that the prediction we are discussing is a simple linear extrapolation so long prediction times may produce inaccurate results if the training sequence is too nonlinear over the range of extrapolation.

Calculations (see Appendix B) also show that the size of the shift in the population vector is relatively insensitive to the speed at which the training sequence is presented. This is because rapid sequences result in larger asymmetries even though their shorter periods of firing induce less LTP. As a result, a sequence can be learned at any reasonable presentation speed. The resulting synaptic weight changes store information primarily about the direction and not the magnitude of the ‘velocity’ of the training sequence path.

Learning Motor Sequences

The arrows in figure 1 suggest that the information stored in potentiated synapses and read out by the population vector could be used not only to predict a training sequence but to recreate it as well. To make these ideas concrete, consider coding of arm position (Kalaska *et al.*, 1983). In this case, the plots in figure 1 represent the position of the arm (hand, for example) during a two-dimensional movement. The training sequence in figure 1b corresponds to a circular arm movement. The arrows in figure 1b indicate that from any position the population vector indicates where the arm should go next to generate the learned movement. To test whether this prediction can be used to produce a learned sequence, we simulated a movement guided by the population vector. We started with an arbitrary position and generated a movement path by traveling at constant speed from a given position \vec{x} toward the position coded by the population vector evoked by that value of \vec{x} . The solid lines in figure 3 show the resulting movement path. Figures 3 a) and b) show that the learned movement is quite accurately recreated by this simple procedure. Figure 3c shows that when

two paths are superimposed confusion can arise where they cross, but otherwise, the path is followed accurately.

Figure 3 suggests a mechanism by which neuronal activity can guide the execution of a learned motor task. If temporally-asymmetric NMDA-mediated LTP occurs between neurons representing arm position, for example, training experience will cause the position represented by the network firing pattern to lead the actual position of the arm for positions along the path of the learned movement. If the encoded, shifted position evoked by proprioceptive input acts as a target for a motor circuit, the learned movement will automatically be generated. The motion is executed by a loop. Proprioceptive input from the arm generates activity in the coding network that represents not the current position of the arm but the next position in the learned motor sequence. Transfer of this information to the motor systems evokes movement to this new position thereby generating new proprioceptive input. This new input causes the trained network to provide the next target location and so on. The motor system produces the learned movement by chasing the proprioceptively generated target position. In this scheme, the recall of information about the learned sequence is driven by the motor response not by internal feedback within the network. The retrieval rate automatically matches the rate at which the task is being performed because it is the movement itself that triggers the next recall. As a result, once the task is learned it can be performed at any desired rate or in discrete partial sequences.

The ability of this mechanism to reproduce a learned sequence is based on the fact the NMDA-mediated LTP automatically produces a predictive representation of the training sequence. It is equally important that the arrows in figure 1 show a shift toward the training path for sensory inputs beside the training sequence. This indicates that the generated motor sequence is stable. If some perturbation knocks the arm off the learned trajectory, the target position that is generated by the perturbed arm position will direct the motor system back toward the learned path.

Careful examination of figure 3 reveals that there is a tendency for guided paths to ‘cut corners’ relative to the training paths. Figure 4 shows an interesting consequence of this during iterated learning of a movement. When we allow a guided movement to act as the training sequence for subsequent movements, there is a gradual straightening of the path of the movement. If the process is iterated enough times, straight-line motion will result. Straightening will not occur if LTP is deactivated or if some other constraint prevents the straight-line movement from being carried out. However, this process could give rise to more efficient movement trajectories consistent with physical constraints.

Discussion

LTP appears to have the correlation-dependent properties (Morris *et al.*, 1986; Baudry & Davis, 1991; Hawkins *et al.*, 1993) needed to cause information about a training experience to be stored in synaptic weights (Sejnowski, 1977; Byrne & Berry, 1989; Gluck & Rumelhart, 1990; Levy *et al.*, 1990). However, the mechanisms by which this information is read out to affect behavior have not been clearly established. We suggest that the shifts in ensemble-coded quantities that we have computed provide one such mechanism. Through these shifts, information stored in synaptic weights appears as modifications in firing rates that can be transferred to other neural networks (Salinas & Abbott, 1995). We have shown that the temporal properties of NMDA-mediated LTP naturally cause a population of neurons that represent a quantity to start predicting that quantity after training. Population decoding

techniques can reveal this phenomenon at the network level and its molecular basis can be traced to properties of the NMDA receptor (Hestrin *et al.*, 1990; Lester, *et al.*, 1990; Jahr & Stevens, 1990, Madison *et al.*, 1991). We find that training experience produces a shift in the position coded by neuronal ensemble activity that has clear predictive implications. The difference between the coded population vector and sensory input vector provides an experience-dependent map of sensory space that automatically develops during training and that can be used to generate experienced-based predictions and extrapolations. Furthermore, this information can be used to generate learned motor sequences. The figures we have constructed provide a graphic image of the information stored in training-modified synaptic weights.

Direct readout of the coded population vector after training on a specific sequence provides the location of the next point in that sequence. However, for some tasks it may be more advantageous to obtain the direction and distance from the present point to the next point in the sequence analogous to the arrows drawn in figures 1 and 3. To extract this information a neural circuit must effectively subtract the input vector \vec{x} from the population vector \vec{p} . Information about both of these quantities is available: \vec{x} is represented by the input to the coding array and \vec{p} is coded by its output. One method for extracting the difference $\vec{p} - \vec{x}$ (Blum & Abbott, 1996) is based on a mechanism used to determine a gaze-invariant measure of the position of an object in parietal cortex (Andersen *et al.*, 1988; Zipser & Andersen, 1988). If information about both \vec{p} and \vec{x} is conveyed to and simultaneously represented by a single network, it has been shown that downstream networks can extract the difference $\vec{p} - \vec{x}$ (Salinas & Abbott, 1995).

Figures 1c and 3c show that multiple sequences can be stored and recalled but they tend to interfere with each other. Additional information can be provided to resolve ambiguities at junction points but clearly large numbers of sequences cannot be superimposed without loss of information about individual sequences. In our study of hippocampal place-cell coding (Blum & Abbott, 1996) we found that when large numbers of paths were stored the result was a navigational map of the environment, not a record of specific paths. How then are large numbers of sequences stored? One possibility is the random reassignment of tuning curves as occurs for place cells when a rat enters a new environment. If a training sequence is stored and recalled as we have suggested, the reassignment of coding cells would effectively remove the shifts in the population vector that represent the learned sequence. This is because the coherent summation of synaptic weights that produces these shifts is destroyed by the reassignment. A new sequence can then be learned using the new assignment of coding cells. However, since the original pattern of synaptic weight changes has not been erased, this is a non-destructive process. Restoration of the original tuning curves of the coding neurons causes the first sequence to reappear. Through this mechanism a large number of different sequences can be stored simultaneously by shifting and shuffling the tuning curves of the coding neurons.

In our model, the shifts that store a particular sequence arise because network output is decoded using the pre-training firing rate tuning curves f even after the firing rates have been modified by training. Both before and after training, f is used to define \vec{s} in equation (2.1) and in equation (2.2). Suppose that at a later time, we decode using the firing rates given by equation (3.1) instead. The decoding methods we use work for a wide variety of tuning curve shapes so such a switch is possible even if the tuning curves are distorted from their original form. The result of such a modified decoding scheme would be a population

vector that is identical to the sensory input vector. Switching decoding schemes removes the experienced-based shifts in the population vector that reflect the training sequence. When translated into a more realistic framework, this provides a simple mechanism for erasing all traces of previously learned sequences without modifying any synapses within the coding array.

Within the nervous system, decoding of the type we use is unlikely to occur. Rather, information is transferred from one network to another in a distributed form. It has been shown that synaptic connections that accurately transfer encoded information from one network to another can arise from a correlation-based learning rule (Salinas & Abbott, 1995). Furthermore, there is a direct correspondence between the pattern of synaptic connections needed to transfer information and the decoding scheme that reveals the information being transferred. Suppose that the firing rates of the coding array were characterized by tuning curves f during the time that the synaptic connections from the coding array to a downstream network were established. The value encoded in the downstream array will then be the same as the value obtained from our decoding scheme using the tuning curves f (Salinas & Abbott, 1995).

An implicit assumption in our work is that synaptic connections from the coding array to downstream networks are not changed during the training experience that modifies synapses within the coding array. As a result, the information transferred to downstream networks after training is revealed by decoding the output of the coding array using the pre-training tuning curves, as we have done. However, synapses to downstream networks could be modified at a later time. Suppose that the same processes that originally established the synaptic weights from the coding array to a downstream network is activated sometime after training. The new synapses that develop between these two networks will no longer transfer the value decoded using the original tuning curves f . Instead, the relevant tuning curves will be those given by equation (3.1). Once the synapses to the downstream networks have been modified, the value transferred to them and revealed by the population vector will be identical to the sensory input. The previous training-induced shifts will have been completely eliminated. Thus, modifying and resetting synapses to downstream networks completely erases all information about previously learned sequences, leaving the network in a state suitable for learning new and different sequences.

The mechanisms for readout and erasure of stored sequences just outlined provide a rather novel picture of what a memory is. The sequence memories we have studied are relative; they are recalled through the interaction of a representation that was modified by training with another unmodified representation. The memorized information resides in differences between these two representations and memory is erased if both are modified so that these differences are eliminated. The pattern of synaptic weights within the coding array is, by itself, meaningless since it can be cancelled by a set of corresponding weights between the coding and readout networks. In this view, information is stored and recalled only through relationships between two sets of synaptic weights, in particular, through experience-induced differences between them.

Experimental Predictions

Our model makes some specific prediction that can be tested experimentally. First, we predict that the responses of encoding neurons will be shifted by training experience in a

distinct manner. Equations (2.4) and (3.1) provide this prediction which is shown, for one particular example, in figure 5. Figure 5 shows the effect of the training sequence used in figure 1a on the firing rates of 4 individual encoding neurons. The most important prediction is that neurons with tuning curves overlapping a training sequence will be elongated along the direction of the training sequence and shifted *backward* along the path of the training sequence. This effect is intimately related to the tendency of the population vector to *lead* the sequence. Experimental verification of this effect would be strong evidence for the presence of a predicting representation. The parameters chosen for figure 5 resulted in a small amount of elongation and a larger backward shift. However, this parameter choice is not unique so we would more generally predict two consequences of training: 1) An elongation of the receptive field along the path of the training sequence, especially in cases where the sequence is experienced in both directions during training. 2) An additional backward shift of the receptive field when the sequence is experienced unidirectionally during training. Finally, the receptive fields shown in figure 5 are larger after training than before. This is because our model uses only LTP so that firing rates are increased by training. If LTD or enhancement of inhibition also occurs during training, the growth of the receptive fields may not be as pronounced.

It may be possible to generate maps like those in figure 1 directly from experimental data. Equations (3.2) or (6.11) allow the arrows in these maps to be computed if the tuning curves are measured and if the training induced shifts in synaptic efficacy, represented by ΔW_{ij} , are known. Unfortunately, it is not possible to measure large numbers of synaptic weights directly, but it may be possible to infer them from firing correlation measurements (Georgopoulos *et al.*, 1994; Wilson & McNaughton, 1994). If the asymmetric, short-latency firing correlation matrix is used in place of ΔW_{ij} in equation (3.2) or (6.11), a map representing the effect of training on a neuronal representation of sensory data can be generated. Results like those in figure 1 would provide remarkable insight into the functional significance of changes in synaptic efficacy.

Generating Motor Sequences

In our model for the generation of learned motor sequences, proprioceptive feedback provides the input that drives further movement (Feldman, 1966; Ghez, *et al.* 1990). Proprioception is known to play a key role in movement generation in humans (Sittig *et al.*, 1987; Soechting & Flanders, 1989; Cordo, 1990; Cordo *et al.*, 1994), other mammals (Andersson & Grillner, 1983; Conway *et al.*, 1987; Saling *et al.*, 1992) and insects (Bassler, 1986; Wolf & Pearson, 1988). Attempts to generate movements in the absence of proprioceptive feedback have been studied in patients with an unusual neuropathy (Rothwell *et al.*, 1982; Sanes *et al.*, 1985; Cole & Sedgwick, 1992). Psychophysical results suggest that during movement the perception of a limb position may lead its actual location (Dassonville, 1994) as we would predict. During rapid movements, it is likely that limb velocity rather than limb position is the correct encoded quantity to consider (Sittig *et al.*, 1987; Schwartz, 1994). For very rapid movements, proprioceptive feedback about limb position may be too delayed to generate the population vector guiding movement rapidly enough. It has been suggested that the cerebellum may generate an internal copy of the proprioceptive feedback (Miall *et al.*, 1993) and it might be possible that such an internally generated proprioceptive signal could be used to drive recall of learned movements requiring very rapid feedback.

A number of models of arm movements have been developed (Flash, 1987; Kawato *et al.*, 1988; Bullock & Grossberg, 1988; Kuperstein, 1988; Uno *et al.*, 1989; Massone & Bizzi, 1989; Lukashin, 1990; Jeannerod, 1990; Gaudiano & Grossberg, 1991; Mussa-Ivaldi & Giszter, 1992; Burnod *et al.*, 1992; Houk *et al.*, 1990; Lukashin & Georgopoulos, 1993; Berthier *et al.*, 1993). Many of these are complementary to our model (although see Houk *et al.*, 1990; Lukashin & Georgopoulos, 1993; Berthier *et al.*, 1993) but some of the mechanisms discussed in these models can be used in our approach to transfer information from the proprioceptive array that provides the target location of a learned sequence to motor circuits (Salinas & Abbott, 1995).

Schemes for generating sequences have been constructed using mathematical neural network models (see Amit, 1989; Hertz *et al.*, 1991 for reviews, Minai & Levy, 1993). A key distinction between these models and ours concerns the rate at which the learned sequence is generated. In previous models nonlinear networks produce learned sequences autonomously through internal feedback. This means that a sequence is generated at a speed characteristic of neuronal dynamics. In most cases, this is far too fast to direct a motor system. Furthermore, it does not allow for any adjustments to delays or errors in the motor response. Thus even if the sequence could be generated slowly enough, the movement could only be performed at one speed and it would not be stable to perturbations. By using population decoding techniques, we have been able to include proprioceptive feedback from the moving limb into our model and we use it, not internal feedback, to drive the network. Since recall is response driven, the learned sequence is generated at a rate that matches the task. Furthermore, once it is learned, the task can be performed at any desired speed with or without pauses and stability of the motion is assured.

Acknowledgements: Supported by NSF-DMS9208206 and the McDonnell-Pew Centre for Cognitive Neuroscience at Oxford (LA) and NIH-NS07292 (KB). We thank Marco Idiart, John Lisman and Eve Marder for discussions.

Appendix A

Equation (3.2) is derived by substituting equation (3.1) for the firing rates after training into equation (2.1) defining the population vector giving

$$\vec{p} = \frac{\sum_i \left(f_i(\vec{x}) + \sum_j \Delta W_{ij} f_j(\vec{x}) \right) \vec{s}_i}{\sum_i \left(f_i(\vec{x}) + \sum_j \Delta W_{ij} f_j(\vec{x}) \right)}. \quad (5.1)$$

We assume that before training, when $\Delta W = 0$, the population vector is equal to the sensory input so

$$\frac{\sum_i f_i(\vec{x}) \vec{s}_i}{\sum_i f_i(\vec{x})} = \vec{x}. \quad (5.2)$$

Equation (3.2) follows from this and an expansion of the denominator of equation (5.1) to linear order in ΔW .

The analogous result for the shift in the population vector using the least-squares method is obtained by substituting equation (3.1) into equation (2.2) so that \vec{p} is determined by the

condition

$$\sum_i \left(f_i(\vec{x}) + \sum_j \Delta W_{ij} f_j(\vec{x}) - f_i(\vec{p}) \right)^2 = \text{minimum} . \quad (5.3)$$

It is convenient to express the population vector as

$$\vec{p} = \vec{x} + \Delta \vec{p} \quad (5.4)$$

and we compute to linear order in $\Delta \vec{p}$ writing

$$f_i(\vec{p}) \approx f_i(\vec{x}) + \vec{\nabla} f_i(\vec{x}) \cdot \Delta \vec{p} . \quad (5.5)$$

Performing the minimization with respect to $\Delta \vec{p}$ we find

$$\sum_i \left(\sum_j \Delta W_{ij} f_j(\vec{x}) - \vec{\nabla} f_i(\vec{x}) \cdot \Delta \vec{p} \right) \vec{\nabla} f_i(\vec{x}) = 0 . \quad (5.6)$$

Defining the matrix

$$\mathbf{Q} = \sum_i \vec{\nabla} f_i(\vec{x}) \vec{\nabla} f_i(\vec{x}) \quad (5.7)$$

the solution is

$$\Delta \vec{p} = \mathbf{Q}^{-1} \sum_{ij} \vec{\nabla} f_i(\vec{x}) \Delta W_{ij} f_j(\vec{x}) . \quad (5.8)$$

If the tuning curves f are uniformly arranged the inverse matrix just introduces a numerical constant and

$$\Delta \vec{p} \propto \sum_{ij} \vec{\nabla} f_i(\vec{x}) \Delta W_{ij} f_j(\vec{x}) . \quad (5.9)$$

Appendix B

Equations (2.4) and (3.2) provide the results needed to generate the figures shown. We will discuss how these equations are evaluated for the case when \vec{x} is two-dimensional as in the figures. Other dimensions involve trivial changes in the formulas. The tuning curves are taken to be Gaussian,

$$f_i(x) = R_{max} \exp \left(-\frac{|\vec{x} - \vec{s}_i|^2}{2\sigma^2} \right) , \quad (6.1)$$

and the vector \vec{x} is allowed to extend over an infinite range. We assume that the maximal response values \vec{s}_i are spread uniformly and that the firing rate curves are highly overlapping. This allows us to replace sums over coding neurons with integrals over their maximal response vectors,

$$\sum_i \rightarrow \rho \int d\vec{s} \quad (6.2)$$

where ρ is equal to the number of neurons with \vec{s}_i vectors lying within one unit area in the space of \vec{x} values. We have checked the accuracy of this approximation and found it to be very good even for a modest number of neurons (Salinas & Abbott, unpublished). For simplicity,

when computing the effects of LTP we assume all-to-all coupling between coding neurons. Using these approximations, equation (3.2) becomes

$$\vec{p} = \vec{x} + \frac{\vec{A}}{B} \quad (6.3)$$

where

$$\vec{A} = \rho^2 R_{max} \int d\vec{s}_1 d\vec{s}_2 (\vec{s}_1 - \vec{x}) \Delta W(\vec{s}_1 - \vec{s}_2) \exp\left(-\frac{|\vec{x} - \vec{s}_2|^2}{2\sigma^2}\right) \quad (6.4)$$

and

$$B = \rho R_{max} \int d\vec{s} \exp\left(-\frac{|\vec{x} - \vec{s}|^2}{2\sigma^2}\right) \quad (6.5)$$

with

$$\Delta W(\vec{s}_1 - \vec{s}_2) = R_{max}^2 \int dt dt' H(t') \exp\left(-\frac{|\vec{X}(t+t') - \vec{s}_1|^2 + |\vec{X}(t) - \vec{s}_2|^2}{2\sigma^2}\right). \quad (6.6)$$

Performing the Gaussian integrals gives

$$\vec{p} = \vec{x} + \pi\sigma^2 \rho R_{max}^2 \int dt dt' H(t') (\vec{X}(t+t') - \vec{x}) \exp\left(-\frac{|\vec{X}(t) - \vec{x}|^2}{4\sigma^2}\right). \quad (6.7)$$

We have used this formula to check our numerical work but in the results shown we have made use of a further approximation within this integral,

$$\vec{X}(t+t') \approx \vec{X}(t) + t' \vec{X}'(t), \quad (6.8)$$

where $\vec{X}' = d\vec{X}/dt$. With this approximation we find

$$\vec{p} = \vec{x} + \pi h R_{max}^2 \rho \sigma^2 \int dt [\vec{X}(t) - \vec{x} + \tau \vec{X}'(t)] \exp\left(-\frac{|\vec{X}(t) - \vec{x}|^2}{4\sigma^2}\right). \quad (6.9)$$

Here h is the time integral of H and τ is average LTP window time given by

$$\tau = \frac{1}{h} \int dt' t' H(t'). \quad (6.10)$$

The term $\vec{X} - \vec{x}$ in the integral of equation (6.9) pulls the population vector toward values in the training sequence while the term proportional to \vec{X}' moves it forward along the training sequence providing future prediction. The figures were generated by numerically integrating equation (6.9) using $\pi h R_{max}^2 \rho \sigma^2 = 2$, $\tau = 200ms$ and $\sigma = 0.7$. The magnitude of the velocity $|\vec{X}'|$ was 5/s except for the circular path where it was 12.5/s.

For the case of the least-square method, we obtain similar results. We evaluate equation (5.8) for the training-induced shift in the population vector exactly as above using Gaussian tuning curves and replacing sums by integrals. The results are

$$\Delta \vec{p} = \frac{2h}{\sqrt{\pi}} \sum_{ij} \vec{\nabla} f_i(\vec{x}) \Delta W_{ij} f_j(\vec{x}) \quad (6.11)$$

and

$$\Delta \vec{p} = \pi h R_{max}^2 \rho \sigma^2 \int dt \left\{ \vec{z}(t) + \tau \dot{\vec{x}}(t) - \frac{\tau \vec{z}(t) \dot{\vec{x}}(t) \cdot \vec{z}(t)}{2\sigma^2} \right\} \exp \left(-\frac{|\vec{z}(t)|^2}{2\sigma^2} \right) \quad (6.12)$$

where $\vec{z}(t) = \vec{X}(t) - \vec{x}$. This $\Delta \vec{p}$ is not identical to the difference $\vec{p} - \vec{x}$ from equation (6.9) but it produces very similar figures. If diagrams like those of figure 1 are generated using this result, the arrows beside the training sequence point slightly more in the direction toward the training path and slightly less along it. For points on the training path, the direction of the arrows is unchanged but they are slightly shorter for the same parameter values.

It is interesting to evaluate equation (6.9) for the case where sensory inputs in the training sequence change at a constant ‘velocity’, $\vec{X}(t) = \vec{V}t$. This provides a good approximation of the population vector for sensory inputs in a sufficiently small range near the middle of a long training sequence. In this case,

$$\vec{p} = \vec{x} + 2\pi^{3/2} h R_{max}^2 \rho \sigma^2 \left(\tau \sigma \hat{V} - \frac{\sigma \vec{x}_\perp}{|\vec{V}|} \right) \exp \left(-\frac{|\vec{x}_\perp|^2}{4\sigma^2} \right) \quad (6.13)$$

where \vec{x}_\perp is the component of \vec{x} perpendicular to the path of the training sequence, $\vec{x}_\perp = \vec{x} - (\vec{x} \cdot \hat{V}) \hat{V}$, and $\hat{V} = \vec{V}/|\vec{V}|$. The term proportional to \hat{V} moves the population vector forward along the training sequence. The fact that this term is proportional to \hat{V} means that it is independent of the magnitude of the training ‘velocity’ vector. The power to predict decreases exponentially with the square of the distance between the sensory input \vec{x} and the training sequence. In equation (6.13), $h R_{max}^2 \tau$ is the amount of synaptic potentiation that occurs in one LTP window time between two neurons firing at rates R_{max} . The factor $\rho \sigma^2$ is equal to the number of units with maximal response vectors \vec{s}_i lying within a one σ by one σ area. This is equal to the number of neurons with tuning curves that peak within the width of a typical tuning curve and is a measure of the degree of overlap between the coding neurons. The factor $\sigma/|\vec{V}|$ is approximately the length of time that any one neuron fires during the training sequence.

References

- Abbott, L.F. (1990) Learning in Neural Network Memories. *Network: Comp. Neural Sys.* 1:105-122.
- Abbott, L.F. (1994) Decoding Neuronal Firing and Modeling Neural Networks. *Quart. Rev. Biophys.* 27:291-331.
- Amit, D.J. (1989) *Modelling Brain Function*. Cambridge University Press, NY.
- Andersen, R.A., Essick, G.K., Siegel, R.M. (1985) Encoding of spatial location by posterior parietal neurons. *Science* 230:450-458.
- Andersson, O. and Grillner S. (1983) Peripheral control of the cat's step cycle. II. Entrainment of the central pattern generators for locomotion by sinusoidal hip movements during 'fictive locomotion'. *Acta Physiol. Scand.* 118:229-239.
- Bassler, U. (1986) Afferent control of walking movements in the stick insect *Cuniculina impigra*. II. Reflex reversal and the release of the swing phase in the restrained foreleg. *J. Comp. Physiol.* 150:351-362.
- Baudry, M. & Davis, J.L., eds. (1991) *Long-Term Potentiation* (MIT Press, Cambridge MA).
- Berthier, N.E., Singh, S.P, Barto, A. and Houk, J.C. (1993) Distributed representation of limb motor programs in arrays of adjustable pattern generators. *J. Cogn. Neurosci.* 5:56-78.
- Bienenstock, E.L., Cooper, L.N. and Munro, P.W. (1982) Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* 2:32-48.
- Bliss, T.V.P. and Collingridge, G.L. (1993) A synaptic model of memory: long-term potentiation in the hippocampus. *Nature* 361:31-39.
- Blum K.I. & Abbott, L.F. (1996). A model of spatial map formation in the hippocampus of the rat. *Neural Comp.* 8:85-93.
- Bullock, D. and Grossberg, S. (1988) Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation. *Psychology Rev.* 95:49-90.
- Burnod, Y., Grandguillaume, P., Otto, I., Ferraina, S. and Johnson, P.B. and Caminiti R. (1992) Visuomotor transformation underlying arm movements toward visual targets: A neural network model of cerebral cortical operations. *J. Neurosci.* 12:1435-1452.
- Byrne, J.H. & Berry, W.O., eds. (1989) *Neural Models of Plasticity* (Academic Press, San Diego).
- Caminiti, R., Johnson, P.B., Galli, C., Ferraina, S. and Burnod, Y. (1991) Making arm movements within different parts of space: The premotor and motor cortical representations of a coordinate system for reaching to visual targets. *J. Neurosci.* 11:1182-1197.
- Cole, J.D. and Sedgwick, E.M. (1992) The perceptions of force and movement in a man without large myelinated sensory afferents below the neck. *J. Physiol. Lond.* 449:503-515.
- Conway, B.A., Hultborn, H. and Kiehn, O. (1987) Proprioceptive input resets central locomotor rhythm in the spinal cat. *Exp. Brain Res.* 68:643-656.
- Cordo, P.J. (1990) Kinesthetic control of a multijoint movement sequence. *J. Neurophysiol.* 63:161-172.
- Cordo, P.J., Carlton, L., Bevan, L., Carlton, M., and Kerr, G.K. (1994) Proprioceptive coordination of movement sequences: role of velocity and position information. *J. Neurophysiol.* 71:1848-1861.
- Dassonville, P. (1994) Tactile localization and the internal representation of hand in space.

(submitted).

Debanne, D., Gähwiler, B.H. and Thompson, S.M. (1994) Asynchronous pre- and postsynaptic activity induces associative long-term depression in area CA1 of the rat hippocampus *in vitro*. *Proc. Natl. Acad. Sci. USA* 91:1148-1152.

Eichenbaum, H. (1993) Thinking about brain cell assemblies. *Science* 261:993-994.

Feldman, A.G. (1966) Functional tuning of the nervous system during control of movement or maintenance of a steady posture. III. Mechanographic analysis of the execution by man of the simplest motor tasks. *Biofizika* 11:76-775.

Flash, T. (1987) The control of hand equilibrium trajectories in multi-joint arm movements. *Biol. Cybern.* 52:257-274.

Fortier, P.A., Kalaska, J.F. and Smith, A.M. (1989) Cerebellar neuronal activity related to whole-arm reaching movements in the monkey. *J. Neurophysiol.* 62:198-211.

Gaudiano, P. and Grossberg, S. (1991) Vector associative maps: unsupervised real-time error based learning and control of movement trajectories. *Neural Networks* 4:147-183.

Georgopoulos, A.P., Kettner, R.E. and Schwartz, A. (1988) Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *Neurosci.* 8:2928-2937.

Georgopoulos, A.P., Schwartz, A. and Kettner, R.E. (1986) Neuronal population coding of movement direction. *Science* 233:1416-1419.

Georgopoulos, A.P., Taira, M. and Lukashin, A. (1993) Cognitive neurophysiology of the motor cortex. *Science* 260:47-52.

Ghez, C., Gordon, J., Ghilardi, M.F., Christakos, C.N. and Cooper, S.E. (1990) Roles of proprioceptive input in the programming of arm trajectories. *Cold Spring Harbor Symp. on Quant. Biol.* 55:837-847.

Gilbert, C.D. and Wiesel, T.N. (1990) The influence of contextual stimuli on the orientation selectivity of cells in primary visual cortex of the cat. *Vision Res.* 30:1689-1701.

Gluck, M.A. and Rumelhart, D.E. (1990) *Neuroscience and Connectionist Theory*. Lawrence Erlbaum, Hillsdale NJ.

Grossberg, S. (1988) Nonlinear neural networks: principles, mechanisms and architectures. *Neural Networks* 1:17-61.

Gustafsson, B., Wigstrom, H., Abraham, W.C. and Huang, Y.-Y. (1987) Long-term potentiation in the hippocampus using depolarizing current pulses as the conditioning stimulus to single volley synaptic potentials. *J. Neurosci.* 7:774-780.

Hawkins, R.D., Kandel, E.R. and Siegelbaum, S.A. (1993) Learning to modulate transmitter release: Themes and variations in synaptic plasticity. *Annu. Rev. Neurosci.* 16:625-665.

Hebb, D.O. (1949) *The Organization of Behavior: A Neuropsychological Theory*. J. Wiley, NY.

Hertz, J., Palmer, R.G. and Krogh, A. (1991) *Introduction to the Theory of Neural Computation*. Addison-Wesley, NY.

Hestrin, S., Sah, P. and Nicoll, R.A. (1990) Mechanisms generating the time course of dual component excitatory synaptic currents recorded in hippocampal slices. *Neuron* 5:247-253.

Hopfield, J.J. (1982) Neural networks and systems with emergent selective computational abilities. *Proc. Natl. Acad. Sci. USA* 79:2554-2558.

Houk, J.C., Keifer, J. and Barto, A.G. (1993) Distributed motor commands in the limb pre-motor network. *Trends Neurosci.* 16:27-33.

Jahr, C.E. and Stevens, C.F. (1990) A quantitative description of NMDA receptor channel

kinetic behavior. *J. Neurosci.* 10:1830-1837.

Jeannerod, M. ed. (1990) *Attention and Performance XIII. Motor Representation and Control* (Erlbaum, Hillsdale, NJ).

Kalaska, J.F., Caminiti, R. and Georgopoulos, A.P. (1983) Cortical mechanisms related to the direction of two-dimensional arm movements: relations in parietal area 5 and comparison with motor cortex. *Exp. Brain. Res.* 51:247-260.

Kawato, M., Isobe, M., Maeda, Y. and Suzuki, R. (1988) Coordinate transformations and learning control for visually guided voluntary movements with iteration: a Newton-like method in function space. *Bio. Cybern.* 59:161-177.

Knudsen, E. and Konishi, M. (1978) A neural map of auditory space in the owl. *Science* 200:795-797.

Knudsen, E., DuLac, S. and Esterly, S.D. (1987) Computational maps in the brain. *Annu. Rev. Neurosci.* 10:41-65.

Kohonen, T. (1984) *Self Organization and Associative Memory*. Springer Verlag, Berlin.

Kohonen, T. (1988) An introduction to neural computing. *Neural Networks* 1:3-16.

Konishi, M. (1987) Centrally synthesized maps of sensory space. *Trends Neurosci.* 9:163-168.

Konishi, M. (1991) Deciphering the brain's codes. *Neural Comp.* 3:1-18.

Kuperstein, M. (1988) Neural model of adaptive hand-eye coordination for single postures. *Science* 239:1208-1311.

Lester, R.A.J., Clements, J.D., Westbrook, G.L. and Jahr, C.E. (1990) Channel kinetics determine the time course of NMDA receptor-mediated synaptic currents. *Nature* **346**, 565-567.

Levy, W.B. (1989) A computational approach to hippocampal function. In *Computational Models of Learning in Simple Neural Systems* (R.D. Hawkins and G.H. Bower, Eds.) Academic Press, San Diego. pp. 243-305.

Levy, W.B., Colbert, C.M. and Desmond, N.L. (1990) Elemental adaptive processes of neurons and synapses: a statistical/computational perspective. In Gluck, M.A. and Rumelhart, D.E. eds., *Neuroscience and Connectionist Theory*. Lawrence Erlbaum, Hillsboro, N.Y. pp. 187-236.

Levy, W.B. and Steward, O. (1983) Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neurosci.* 8:791-797.

Linker, R. (1986) From basic network principles to neural architecture. *Proc. Natl. Acad. Sci. USA* 83:7508-7512, 8390-8394 & 8779-8783.

Lukashin, A.V. (1990) A learned neural network that simulates properties of the neural population vector. *Biol. Cybern.* 63:377-382.

Lukashin, A.V. and Georgopoulos, A.P. (1993) A dynamical neural network model for motor cortical activity during movement: population coding of movement trajectories. *Biol. Cybern.* 69:517-524.

Madison, D.V., Malenka, R.C. and Nicoll, R.A. (1991) Mechanisms underlying long-term potentiation of synaptic transmission. *Annu. Rev. Neurosci.* 12:379-397.

Malenka, R.C. and Nicoll, R.A. (1993) MBDA-receptor-dependent synaptic plasticity: Multiple forms and mechanisms. *Trends Neurosci.* 16:521-527.

Marr, D. (1971) Simple memory: A theory for archicortex. *Phil. Trans. Royal Soc. London* B262:21-81.

Massone, L. and Bizzi, E. (1989) A neural network model for limb trajectory formation. *Biol. Cybern.* 61:417-425.

- Maunsell, J.H.R. and Newsome, W.T. (1987) Visual processing in monkey extrastriate cortex. *Annu. Rev. Neurosci.* 10:363.
- MacKay, D.J.C. and Miller, K.D. (1990) Analysis of Linsker's application of Hebbian rules to linear networks. *Network* 1:257-299.
- Miall, R.C., Weir, D.J., Wolpert, D.M. and Stein, J.F. (1993) Is the cerebellum a Smith predictor? *J. Motor Behav.* 25:203-216.
- Miller, K.D. (1990) Correlation-based models of neural development. In Gluck, M.A. and Rumelhart, D.E., eds., *Neuroscience and Connectionist Theory*. Lawrence Erlbaum, Hillsdale NJ. pp. 267-354.
- Miller, K.D. (1992) Models of activity-dependent neural development. *Seminars in Neurosci.* 4:61-73.
- Miller, K.D. (1994) A model for the development of simple cell receptive fields and the ordered arrangement of orientation columns through activity-dependent competition between on- and off-center inputs. *J. Neurosci.* 14:409-441.
- Miller, K.D. and MacKay, D.J.C. (1994) The role of constraints in Hebbian learning. *Neural Comp.* 6:100-126.
- Minai, A.A. and Levy, W.B. (1993) Sequence learning in a single trial. *INNS World Congress on Neural Networks II*:505-508.
- Morris, R.G.M., Anderson, E., Lynch, G.S. and Baudry, M. (1986) Selective impairment of learning and blockade of long-term potentiation by an N-methyl-D-aspartate receptor antagonist, AP5. *Nature* 319: 774-776.
- Muller, R.U. & Kubie, J.L. (1989) The firing of hippocampal place cells predicts the future position of freely moving rats. *J. Neurosci.* **9**, 4101-4110.
- Muller, R.U., Kubie, J.L. & Saypoff, R. (1991) The hippocampus as a cognitive graph. *Hippocampus* **1**, 243-246.
- Mussa-Ivaldi, F.A. and Giszter, S.F. (1992) Vector-field approximation: a computational paradigm for motor control and learning. *Biol. Cybern.* 67:491-500.
- O'Keefe, J. & Dostrovsky, J. The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* **34**, 171-175 (1971).
- O'Keefe, J. A review of the hippocampal place cells. *Prog. Neurobiol.* **13**, 419-439 (1979).
- O'Keefe, J. & Nadel, L. *The Hippocampus as a Cognitive Map* (Clarendon, London, 1978).
- O'Neill, W.E. & Suga, N. (1982) Encoding of target range information and its representation in the auditory cortex of the mustache bat. *J. Neurosci.* 2:17-31.
- Orban, G.A. (1984) *Neuronal Operations in the Visual Cortex*. Springer, Berlin.
- Rolls, E.T. (1989) Functions of neuronal networks in the hippocampus and neocortex in memory. In Byrne, J.H. & Berry, W.O., eds. *Neural Models of Plasticity* (Academic Press, San Diego) pp. 240-265.
- Rothwell, J.C., Traub, M.M., Day, B.L., Obeso, J.A., Thomas, P.K. and Marsden, C.D. (1982) Manual motor performance in a deafferented man. *Brain* 105:515-542.
- Salinas, E. & Abbott, L.F. (1994) Vector reconstruction from firing rates. *J. Computational Neurosci.* 1:89-107.
- Salinas, E. & Abbott, L.F. (1995) Transfer of information between sensory and motor networks. *J. Neurosci.* 15:6461-6474.
- Saling, M., Sitarova, T., Vejsada, R. and Hnik, P. (1992) Reaching behavior in the rat: absence of forelimb peripheral input. *Physiol. Behav.* 51:185-191.
- Sanes, J.N., Mauritz, K.-H., Dalakas, M.C. and Evarts, E.V. (1985) Motor control in humans

with large-fiber sensory neuropathy. *Hum. Neurobiol.* 4:101.

Schwartz, A.B. (1994) Direct cortical representation of drawing. *Science* 265:540-542.

Sejnowski, T.J. (1977) Storing covariance with nonlinearly interacting neurons. *J. Math. Biol.* 4:303-321.

Sejnowski, T.J. (1988) Neural populations revealed. *Nature* 332:308.

Sittig, A.C., Denier van der Gon, J.J. and Gielen, C.C.A.M. (1987) The contribution of afferent information on position and velocity to the control of slow and fast human forearm movements. *Exp. Brain Res.* 67:33-40.

Soechting J.F. and Flanders, M. (1992) Moving in 3-dimensional space: frames of reference, vectors and coordinate systems. *Ann. Rev. Neurosci.* 15:167-191.

Steinmetz, M.A., Motter, B.C., Duffy, C.J. and Mountcastle, V.B. (1987) Functional properties of parietal visual neurons: Radial organization of directionalities with the visual field. *J. Neurosci.* 7:177-191.

Uno, Y., Kawato, M. and Suzuki, R. (1989) Formation and control of optimal trajectory in human multijoint arm movement. *Biol. Cybern.* 61:89-101.

Wilson, M.A. & McNaughton, B.L. Dynamics of the hippocampal ensemble code for space. *Science* **261**, 1055-1058 (1993).

Wilson, M.A. and McNaughton, B.L. 1994 Reactivation of hippocampal ensemble memories during sleep. *Science* 265:676-679.

Wolf, H. and Pearson, K.G. (1988) Proprioceptive input patterns elevator activity in the locust flight system. *J. Neurophysiol.* 59:1831-1853.

Young, M.P. and Yamane, S. (1992) Sparse population coding of faces in the inferotemporal cortex. *Science* 256:1327-1331.

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

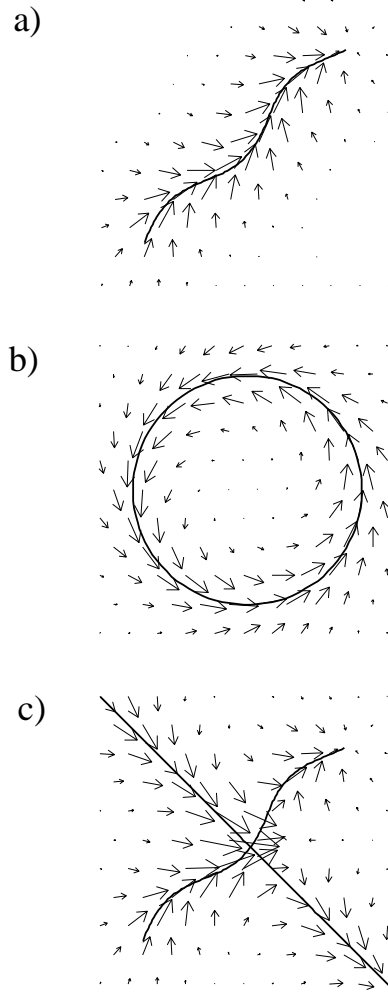


Fig. 1

Figure 1: Training-induced shifts in the encoded value of a sensory input. The sensory input vector is represented as a point in a two-dimension plot of size 10 by 10. Initially the ensemble-coded population vector and the sensory input vector are identical (not shown). The solid curves show the training sequences used to induce LTP between encoding neurons. The arrows show the shifts in the encoded population vector arising from this simulated training experience. Arrows are drawn for a grid of different sensory input values. a) The training sequence moved from lower left to upper right at uniform speed along the solid curve. b) The training sequence is represented by a counter-clockwise circle with constant speed. c) Two superimposed training sequences are shown. One is the same as a) while the other is a straight line with constant velocity moving from upper left to lower right.

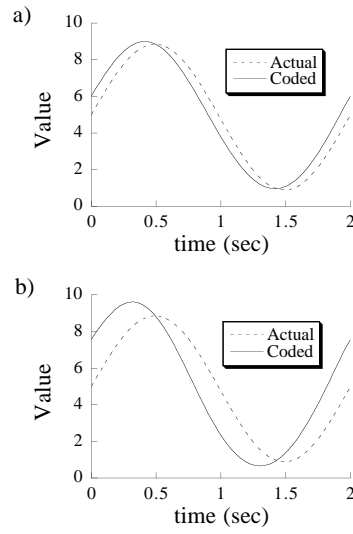


Figure 2: Comparison of actual and coded values after training. The value plotted is the horizontal component of the position in the plane of figure 1. The network was first trained using the circular sequence of figure 1b. The same training sequence was then used as a sensory input to the trained network and the resulting population vector was computed. The actual sensory input to the network is plotted as a dashed curve and the corresponding encoded value is indicated by a solid curve. In both figures the coded value leads the actual value. The difference between the upper and lower plots corresponds to a 5-fold increase in the overlap of the neuronal tuning curves.

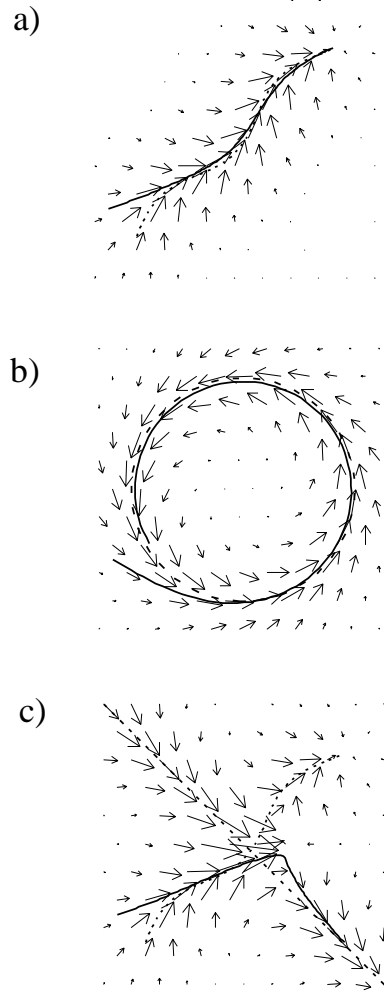


Fig. 3

Figure 3: Movements directed by the training-induced shifts. After training, the networks shown in figure 1 were used to guide movement. The dashed curves indicate the training sequences (identical to those shown in figure 1) while the solid curves show the path of a point whose motion was directed by the population vector arrows. In each case, the movement started at the lower left endpoint of the solid curves. At every time step, the sensory input corresponding to the instantaneous position of the moving point was fed into the network and the position represented by the ensemble firing response was computed. The point was then translated toward this encoded position at a constant speed. In a) and b), the curves generated closely match the training curves. In c) the presence of two training curves led to confusion at their junction and the guided path switched from one curve to the other.

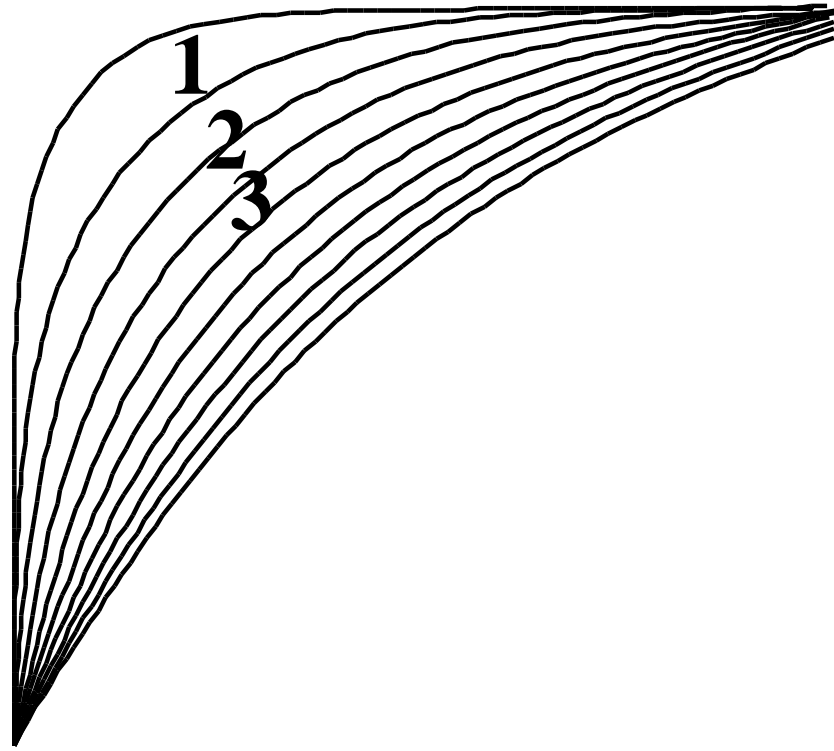


Fig. 4

Figure 4: Gradual straightening of guided movements. The solid curves indicate movements from the lower left to the upper right. The curve marked 1 was used as the training sequence and the next curve, 2, was generated from the resulting population vector as in figure 3. Curve 2 was then used to retrain the network and the resulting shifts generated curve 3. The sequence continued and, as a result, the generated paths gradually straightened. If the process is continued a straight line path will ultimately result (not shown).

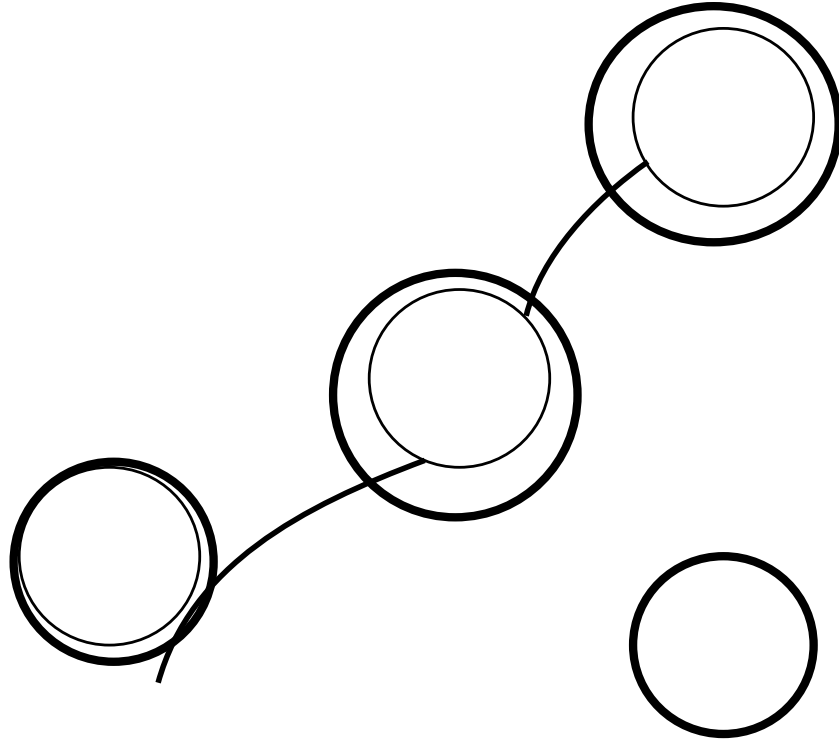


Fig. 5

Figure 5: Firing rates of individual encoding neurons before and after simulated training. Synaptic potentiation was evoked by a training sequence moving from lower left to upper right identical to that in figure 1a and indicated by the solid curve. As a result, neuronal tuning curves located along the training path were elongated and displaced *backward*. Tuning curves beside the path were distorted toward it. The tuning curves of four neurons are shown before and after training. The shaded disks show the initial Gaussian tuning curves. After training, potentiated synapses distort the tuning curves as indicated by the curves surrounding the disks. These curves and the edges of the disks indicate positions where the firing rate is $0.6R_{max}$.