

1.34 Neural Computation Theories of Learning

S. B. Moldakarimov, Salk Institute for Biological Studies, La Jolla, CA, USA

T. J. Sejnowski, Salk Institute for Biological Studies and University of California at San Diego, La Jolla, CA, USA

© 2008 Elsevier Ltd. All rights reserved.

1.34.1	Introduction	667
1.34.2	Hebbian Learning	668
1.34.3	Unsupervised Hebbian Learning	669
1.34.4	Supervised Learning	670
1.34.5	Reinforcement Learning	672
1.34.6	Spike-Timing Dependent Plasticity	673
1.34.7	Plasticity of Intrinsic Excitability	675
1.34.8	Homeostatic Plasticity	676
1.34.9	Complexity of Learning	676
1.34.10	Conclusions	678
References		678

1.34.1 Introduction

The anatomical discoveries in the nineteenth century and the physiological studies in the twentieth century showed that brains were networks of neurons connected through synapses. This led to the theory that learning could be the consequence of changes in the strengths of the synapses.

The best-known theory of learning based on synaptic plasticity is that proposed by Donald Hebb, who postulated that connection strengths between neurons are modified based on neural activities in the presynaptic and postsynaptic cells:

When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased. (Hebb, 1949)

This postulate was experimentally confirmed in the hippocampus with high-frequency stimulation of a presynaptic neuron that caused long-term potentiation (LTP) in the synapses connecting it to the postsynaptic neuron (Bliss and Lomo, 1973). LTP takes place only if the postsynaptic cell is also active and sufficiently depolarized (Kelso et al., 1986). This is due to the *N*-methyl-D-aspartate (NMDA) type of glutamate receptor, which opens when glutamate is bound to the receptor, and the postsynaptic cell is sufficiently depolarized at the same time (See Chapters 1.33, 1.35).

Hebb's postulate has served as the starting point for studying the learning capabilities of artificial neural networks (ANN) and for the theoretical analysis and computational modeling of biological neural systems. The architecture of an ANN determines its behavior and learning capabilities. The architecture of a network is defined by the connections among the artificial neural units and the function that each unit performs on its inputs (See Chapter 1.35). Two general classes are feedforward and recurrent architecture.

The simplest feedforward network has one layer of input units and one layer of output units (Figure 1, left). All connections are unidirectional and project from the input units to the output units. The perceptron is an example of a simple feedforward network (Rosenblatt, 1958). It can learn to classify patterns from examples. It turned out that the perceptron can only classify patterns that are linearly separable – that is, if the positive patterns can be separated from all negative patterns by a plane in the space of input patterns. More powerful multilayer feedforward networks can discriminate patterns that are not linearly separable. In a multilayer feedforward network, the 'hidden' layers of units between the input and output layers allow more flexibility in learning features. Multilayer feedforward networks have also been applied to solve some other difficult problems (Rumelhart and McClelland, 1986).

In contrast to strictly feedforward network models, recurrent networks also have feedback connections

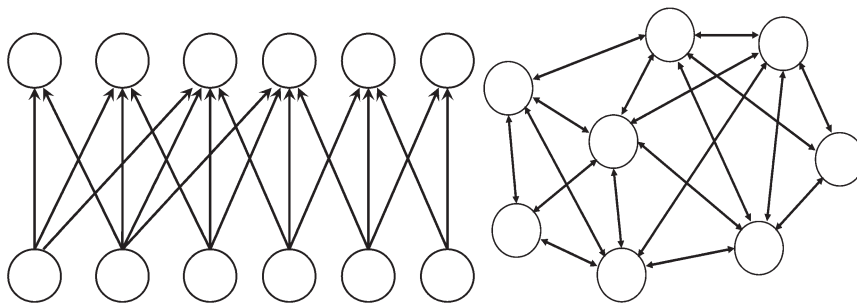


Figure 1 Network architectures. Left: Feedforward network. Right: Recurrent network. Open circles represent neuronal units, and arrowhead lines represent synaptic connections.

among units in the network (Figure 1, right). A simple recurrent network can have a uniform architecture such as all-to-all connectivity combined with symmetric weights between units, as in a Hopfield network (Hopfield, 1982), or it can be a network with specific connections designed to model a particular biological system.

Modeling learning processes in networks implies that the strengths of connections and other parameters are adjusted according to a learning rule (See Chapter 1.33). Other parameters that may change include the threshold of the unit, time constants, and other dynamical variables. A learning rule is a dynamical equation that governs changes in the parameters of the network. There are three main categories of learning rules: unsupervised, supervised, and reinforcement. Unsupervised learning rules are those that require no feedback from a teaching signal. Supervised learning rules require a teacher, who provides detailed information on the desired values of the output units of the network, and connections are adjusted based on discrepancies between the actual output and the desired one. Reinforcement learning is also error correcting but involves a single scalar signal about the overall performance of the network. Thus, reinforcement learning requires less-detailed information than supervised learning.

A learning algorithm specifies how and under what conditions a learning rule or a combination of learning rules should be applied to adjust the network parameters. For a simple task, it is possible to invent an algorithm that includes only one type of learning rule, but for more complex problems, an algorithm may involve a combination of several different learning rules.

In the following sections, we give an overview of basic learning rules and examples of learning

algorithms used in neural network models, and describe specific problems solved by neural networks with adjustable parameters.

1.34.2 Hebbian Learning

Implementations of Hebb's rule can take different forms (Sejnowski and Tesauro, 1988). Simple associative Hebbian learning is based on the coincidence of activities in presynaptic and postsynaptic neurons. The dynamics of Hebbian learning are governed by a differential equation:

$$\frac{dw_{ij}}{dt} = \alpha \cdot v_i \cdot u_j$$

where w_{ij} is the weight of a connection from an input unit j with activity u_j to an output unit i with activity v_i , and α is a learning rate.

The Hebbian learning rule has been used to model a wide variety of problems, including feature selectivity and cortical map development.

Cortical neurons respond selectively to particular feature stimuli, such as selectivity for ocular dominance and orientation in the visual cortex. To understand challenges of modeling the development of feature selectivity, consider a network with many input units and one output unit. We would like to explore under what conditions the output unit will respond well to few input units and less to the others. If we apply a stimulus to the input units and allow the connections to develop according to the Hebbian learning rule, then all connections will grow and eventually saturate, and no selectivity will emerge. To develop selectivity, some dependencies among weights are needed, so that changes at one connection will influence the others. There are many different ways to introduce dependencies. One

approach is to introduce weight normalization (Miller and Mackay, 1994). A different approach, based on competition among input patterns, called the BCM (Bienenstock, Cooper, and Munro) rule (Bienenstock et al., 1982), has been used to model the development of orientation selectivity and ocular dominance in neural networks.

Neuronal response selectivity varies across the cortex in regular patterns called cortical maps. Although some aspects of cortical map formation during development are activity independent, neuronal activity can modify the maps. Hebbian learning rules have also been applied to model the effects of cortical activity on map formations. For comprehensive overviews of neural network models that develop orientation selectivity maps and ocular dominance columns, see Swindale (1996) and Ferster and Miller (2000).

Models of cortical map formation can become extremely complex when multiple features, such as retinotopic location, ocular dominance, orientation preference, and others, are considered simultaneously. To deal with such problems, a more abstract class of models was developed by Kohonen (1982). The Kohonen algorithm is usually applied to two-layer networks with feedforward connections from an input layer to an output layer. The input layer is an N -dimensional vector layer. The output layer is normally a one- or two-dimensional array. There are no lateral connections in the output layer, but the algorithm can accomplish what models with lateral connections can achieve at less computational cost. The algorithm does this by a weight updating procedure that involves neighboring units. At every step, it chooses a 'winner' among output units whose weights are closest to the input pattern. Then it updates the weights of the winner and the nearby neighbors of the winner. The number of neighbors that participate in weight updating is controlled through a neighborhood function, which is dynamically changed during learning to ensure convergence. The neighborhood function starts out long range and is reduced as learning proceeds. This allows the network to organize a map rapidly and then refine it more slowly with subsequent learning.

Models based on the Kohonen algorithm perform dimensionality reduction, which facilitates data analysis, taking input vectors from a high-dimensional feature space and projecting them onto a low-dimensional representation.

1.34.3 Unsupervised Hebbian Learning

If the goal of learning is to discover the statistical structure in unlabeled input data, then the learning is said to be unsupervised. A common method for unsupervised learning is principal component analysis (PCA). Suppose the data are a set of N -dimensional input vectors. The task is to find an $M < N$ dimensional representation of N -dimensional input vectors that contains as much information as possible of the input data. This is an example of dimensionality reduction, which can significantly simplify subsequent data analysis.

A simple network that can extract the first principal component (the one with the maximal variance) is a network with N input units and one output unit. At each time step an N -dimensional input vector is applied to the input layer. If we allow the connections to be modified according to the Hebbian learning rule, then in the case of zero mean value of the input vector, the weights will form an N -dimensional vector, along which the variance will be the largest. This is the principal eigenvector or component. A network with N input and M output units, augmented with a generalized Hebbian learning rule, can learn first M components. The projections of the input data onto the components give us M -dimensional representation of the N -dimensional input data.

PCA is appropriate when the data obey Gaussian statistics, but images, audio recordings, and many types of scientific data often do not have Gaussian distributions. As an example of such a problem, consider a room where a number of people are talking simultaneously (cocktail party), and the task is to focus on one of the speakers. The human brain can, to some extent, solve this auditory source separation problem by using knowledge of the speaker, but this becomes a more difficult problem when the signals are arbitrary. The goal of blind source separation (BSS) is to recover source signals given only sensor signals that are linear mixtures of the independent source signals. Independent component analysis (ICA) is a method that solves the BSS problem for non-Gaussian signals. In contrast to correlation-based algorithms such as PCA and factor analysis, ICA finds a nonorthogonal linear coordinate system such that the resulting signals are as statistically independent from each other as possible.

One approach to BSS derives unsupervised learning rules based on information theory. The input is

assumed to be N mixtures of N independent sources, and the goal is to maximize the mutual information between the inputs and the outputs of a two-layer neural network. The resulting stochastic gradient learning rules are highly effective in the blind separation and deconvolution of hundreds of non-Gaussian sources (Bell and Sejnowski, 1995).

ICA is particularly effective at analyzing electroencephalograms (EEG) and functional magnetic resonance imaging (fMRI) data (Jung et al., 2001). Consider, for example, electrical recordings of brain activity at many different locations on the scalp. These EEG potentials are generated by underlying components of brain activity and various muscle and eye movements. This is similar to the cocktail-party problem: We would like to recover the original components of the brain activity, but we can only observe mixtures of the components. ICA can reveal interesting information of the brain activity by giving access to its independent components. ICA also gives useful insights into task-related human brain activity from fMRI recordings when the underlying temporal structure of the sources is unknown.

Another application of ICA is feature extraction (Lee, 1998). A fundamental problem in signal processing is to find suitable representations for images, audio recordings, and other kinds of data. Standard linear transformations used in image and auditory processing, such the Fourier transforms and cosine transforms, may not be optimal, and but it would be useful to find the most efficient linear transformation, based on the statistics of the data, to optimally compress the data.

1.34.4 Supervised Learning

Consider the problem of learning to retrieve an output pattern given an input pattern. To remember the patterns, the Hebbian rule can be applied to adjust weights between input and output units. As mentioned earlier, however, the associative Hebbian learning rule will lead to saturation with multiple repetitions, which reduces the capacity of the network. To resolve this problem, one can augment the Hebbian rule with a weight normalization algorithm as in the case of unsupervised learning algorithms.

Another disadvantage of using the associative Hebbian learning rule is that weight adjustments do not depend on the actual performance of the network. An effective way to adjust weights would be by using information of the actual performance of

the network. Supervised learning can do this. Supervised learning requires a teacher, who provides detailed information of the desired outputs of the network and adjusts the connections based on discrepancies between the actual outputs and the desired ones.

The perceptron uses a supervised learning rule to learn to classify input patterns (Rosenblatt, 1958). The perceptron is a two-layer network with one output unit that can classify input patterns into two categories. The Hebbian learning rule can be used to solve the task, but the perceptron with the Hebbian learning rule works well only if the number of input patterns is significantly less than the number of input units. An error-correcting supervised learning algorithm for weight adjustments is more effective for a large number of input patterns:

$$\frac{dw_{ij}}{dt} \propto u_j \cdot (R_i - v_i)$$

where w_{ij} is a weight of a connection from the input unit j with activity u_j to an output unit i with activity v_i , R_i is a target value of the output unit, and

$$v_i = \sum_j w_{ij} \cdot u_j$$

The perceptron learning rule uses the performance of the network to decide how much adjustment is needed and in which direction the weights should be changed to decrease the discrepancy between the actual network outputs and the desired ones. If input patterns are linearly separable, then the perceptron learning rule guarantees to find a set of weights that allow pattern classification.

A simple unsupervised Hebbian learning rule adjusts synaptic weights based on correlations between presynaptic and postsynaptic neurons. However, this approach is inefficient when the goal of the network is to perform a specific function, rather than simply represent data. To perform a specific task, the network should receive some information about the task.

An example of how Hebbian plasticity can be incorporated into a supervised learning framework is a two-layer network that was trained to perform a function approximation task (Swinehart and Abbott, 2005). The feedforward connections from input units to output units were modified according to an unsupervised Hebbian rule, and a supervised learning mechanism was used to adjust connections from a supervisor to the network. The supervisor is a network that assesses the performance of the training

network and, based on that information, modifies the gains of the input units using an error-correcting learning rule. The purpose of the supervised modulation was to enhance connections between the input and the output units to facilitate the synaptic plasticity needed to learn the task. Thus, Hebbian plasticity did not have direct access to the supervision, and the supervised modulations did not produce any permanent changes. Nonetheless, this network could learn to approximate different functions. In the initial phase the improvement in the network performance was mostly due to the gain modulation, and the synaptic adjustments were minimal. But later, the synaptic adjustments and the gain modulation were equally involved in shaping the performance. Once the network learned the task with the supervisor, it was possible to turn off the supervision, relying only on further Hebbian plasticity to refine the approximation.

The role of the supervisor in the model was to compute an error by comparing the actual and the desired output of the network and to use this error to direct the modification of network parameters such that the network performance improves. Conventionally, the major targets of this process were the synaptic weights. The novel feature of this supervised learning scheme was that supervision took place at the level of neuronal responsiveness rather than synaptic plasticity.

A simple two-layer perceptron cannot solve higher-order problems, but adding additional layers to the feedforward network provides more representational power. Then new learning algorithms are needed to train multilayer networks. The simple error-correcting learning rule was effective for training two-layer networks. With the rule, the connections from the input layer to the output one are adjusted based on discrepancies between the desired output and the actual output produced by the network. In a multilayer network, however, there are intermediate 'hidden' layers that also need to be trained. The back-propagation learning algorithm was developed to train multilayer networks (Rumelhart and McClelland, 1986). The learning rule relies on passing an error from the output layer back to the input layer. Multilayer networks trained with the back-propagation learning rule have been effective in solving many difficult problems.

An example of a multilayer network that was trained using a back-propagation algorithm is a model of song learning in songbirds (Fiete et al., 2004). Juvenile male songbirds learn their songs

from adult male tutors of the same species. Birdsong is a learned complex motor behavior driven by a discrete set of premotor brain nuclei with well-studied anatomy (See Chapter 1.17). Syringeal and respiratory motor neurons responsible for song production are driven by precisely executed sequences of neural activity in the premotor nucleus robustus archistriatalis (RA) of songbirds (Figure 2). Activity in RA is driven by excitatory feedforward inputs from the forebrain nucleus high vocal center (HVC), whose RA-projecting neural population displays temporally sparse, precise, and stereotyped sequential activity. Individual RA-projecting HVC neurons burst just once in an entire song motif and fire almost no spikes elsewhere in the motif. The temporal sparseness of HVC activity implies that these HVC–RA synapses are used in a special way during song; that is, each synapse is used only once during the motif. The goal of the work was to study the effect of HVC sparseness on the learning speed of the network. They studied multilayer feedforward network with an HVC layer that provides input to a

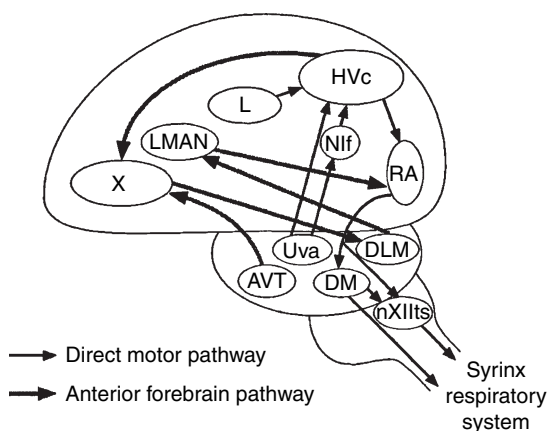


Figure 2 Schematic diagram of the major songbird brain nuclei involved in song control (See also Chapter 1.17). The thinner arrows show the direct motor pathway, and the thicker arrows show the anterior forebrain pathway. Abbreviations: Uva, nucleus uvaeformis of the thalamus; Nif, nucleus interface of neostriatum; L, field L (primary auditory area of the forebrain); HVC, higher vocal center; RA, robust nucleus of the archistriatum; DM, dorsomedial part of the nucleus intercollicularis; nXIIIts, tracheosyringeal part of the hypoglossal nucleus; AVT, ventral area of Tsai of the midbrain; X, area X of lobus parolfactorius; DLM, medial part of the dorsolateral nucleus of the thalamus; LMAN, lateral magnocellular nucleus of the anterior neostriatum. From Doya K and Sejnowski TJ (2000) A computational model of avian song learning. In: Gazzaniga MS (ed.) *The New Cognitive Neurosciences*, 2nd edn., p. 469. Cambridge, MA: MIT Press; used with permission.

'hidden' RA layer and RA projecting to an output layer of motor units. Song learning is thought to involve plasticity of synapses from HVC to RA because these synapses display extensive synaptic growth and redistribution during the critical period. So in the model, the weights from HVC layer to RA layer were modified. Because there is no evidence of plasticity in the synapses from RA to motor neurons, those connections in the model were kept fixed. For learning, the connections from HVC to RA were adjusted to minimize discrepancy between the desired outputs and the actual outputs produced by the network. They used the back-propagation gradient descent rule and varied the number of bursts in HVC neurons per motif. The network learned the motif for any number of bursts in HVC neurons, but the learning time for two bursts per motif nearly doubled compared to the one burst case and increased rapidly with the number of bursts. Based on these simulations, they concluded that the observed sparse coding in HVC minimized interference and the time needed for learning. It is important to note here that the back-propagation learning algorithm was not used to model the biological learning process itself, but rather to determine if the network architecture can solve the problem and what constraints the representation may have on the speed of learning.

1.34.5 Reinforcement Learning

Learning about stimuli or actions based solely on rewards and punishments is called reinforcement learning. Reinforcement learning is minimally supervised because animals are not told explicitly what actions to take in a particular situation. The reinforcement learning paradigm has attracted considerable interest because of the notion that the learner is able to learn from its own experience at attempting to perform a task without the aid of an intelligent 'teacher.' In contrast, in the more commonly employed paradigm of supervised learning, a detailed 'teacher signal' is required that explicitly tells the learner what the correct output pattern is for every input pattern.

A computational model of birdsong learning based on reinforcement learning has been proposed (Doya and Sejnowski, 2000). A young male songbird learns to sing by imitating the song of a tutor, which is usually the father or other adult males in the colony. If a young bird does not hear a tutor song during a

critical period, it will sing short, poorly structured songs. If a bird is deafened during the period when it practices vocalization, it develops highly abnormal songs. Thus, there are two phases in song learning – the sensory learning phase, when a young bird memorizes song templates, and the sensorimotor learning phase, in which the bird establishes the motor programs using auditory feedback. These two phases can be separated by several months in some species, implying that birds have remarkable capability for memorizing complex temporal sequences. Once a song is crystallized, its pattern is very stable. Even deafening the bird has little immediate effect.

The anterior forebrain pathway, which is not involved in song production, is necessary for song learning. In the previously discussed model (Fiete et al., 2004), it was assumed that HVC is a locus of pattern memorization during the first phase of learning, song acquisition, and RA is a motor command area (See Chapter 1.17). Therefore, the patterns stored in HVC serve as inputs to RA to produce motor commands. It was also assumed that evaluation of the similarity of the produced song to the memorized tutor song takes place in area X in the anterior forebrain. This assumption is supported by a finding that area X receives dopaminergic input. Depending on how closely the produced song is to the tutor's song, the connections from HVC to RA are modulated via the lateral magnocellular nucleus (LMAN).

The learning algorithm consisted of making small random changes in the HVC to RA synapses and keeping the new weights only if overall performance was improved. The network learned artificial song motifs and was even able to replicate realistic birdsongs within the number of trials that birds take to learn their songs.

Reinforcement learning has thus far had few practical successes in solving large-scale complex real-world problems. In the case of reinforcement learning with delay, the temporal credit assignment aspect of the problem has made learning very slow. However, a method called temporal difference (TD) learning has overcome some of these limitations (Sutton and Barto, 1998). The basic idea of TD learning is to compute the difference between temporally successive predictions. In other words, the goal of learning is to make the learner's current prediction for the current input pattern more closely match the prediction at the next time step. One of the most effective of these TD methods is an algorithm called TD(λ), in which there is an exponentially decaying feedback of the error in time, so that previous estimates for

previous states are also corrected. The time scale of the exponential decay is governed by the λ parameter.

Perhaps the most successful application of TD(λ) is TD-Gammon, which was designed for networks to learn to play backgammon (Tesauro, 1995). Backgammon is an ancient two-player game that is played on an effectively one-dimensional track. The players take turns rolling dice and moving their checkers in opposite directions along the track as allowed by the dice roll. The first player to move all his checkers all the way forward and off his end of the board is the winner.

At the heart of TD-Gammon is a neural network with a standard multilayer architecture. Its output is computed by a feedforward flow of activation from the input nodes, representing the game position, to the output node, which evaluates the strength of the position. Each of the connections in the network is parameterized by a real valued weight. Each of the nodes in the network outputs a real number equal to a weighted linear sum of inputs feeding into it, followed by a nonlinear sigmoid operation. At each time step, the TD(λ) algorithm is applied to the output, which is then back-propagated to change the network's weights.

During training, the neural network selects moves for both sides. At each time step during the course of a game, the neural network scores every possible legal move. The move that is then selected is the move with maximum expected outcome for the side making the move. In other words, the neural network learns by playing against itself. At the start of self-play, the network's weights are random, and hence its initial strategy is random. But after a few hundred thousand games, TD-Gammon played significantly better than any previous backgammon program, equivalent to an advanced level of play. In particular, it is not dependent on a human teacher, which would limit the level of play it can achieve (Tesauro and Sejnowski, 1989). After one million games, TD-Gammon was playing at a championship level.

One of the essential features of reinforcement learning is a trade-off between exploration and exploitation. The learning system should exploit a successful strategy to reach the goal of the task it learns, but it should also explore other strategies to find out if there is a better one. In models, exploration has been implemented by stochasticity. The source of such stochasticity in the brain remains unclear. A model implementing this trade-off between exploration and exploitation has been proposed (Seung,

2003). The model is based on the probabilistic nature of synaptic release by a presynaptic terminal when an action potential arrives at the terminal. The model combines this local synaptic release-failure event and a global reward signal received outside based on the output of the model. The main assumption is that synapses are hedonistic: they increase their probabilities of release or failure depending on which action immediately preceded reward. This concept of the hedonistic synapse is potentially relevant to any brain area in which a global reinforcement signal is received (Klopf, 1982).

This version of reinforcement learning was used to address the matching law phenomenon (Seung, 2003). When animals are presented with repeated choices between competing alternatives, they distribute their choices so that returns from two alternatives are approximately the same. A return is the total reward obtained from an alternative divided by the number of times it was chosen. Before trials, the alternatives are baited with unequal probabilities. The network had to learn a probabilistic strategy in which one alternative is favored over the other one. The network started from equal choices for both alternatives, but over time, it learned a preference that satisfied the matching law.

In the present model, stochastic vesicle release was assumed to be a source of stochasticity in the brain. However, there might be many other possible sources of noise, such as fluctuations in quantal size, irregular action potential firing, and on a slower time scale, the stochastic creation and destruction of synapses. Thus, identifying specific sources of randomness is essential for connecting mathematical models and neurobiology.

1.34.6 Spike-Timing Dependent Plasticity

The traditional coincidence version of the Hebbian learning rule implies simply that the correlation of activities of presynaptic and postsynaptic neurons drives learning. This approach has been implemented in many types of neural network models using average firing rate or average membrane potentials of neurons (See Chapter 1.35). Although Hebb's formulation implicitly recognized the idea of causality and relative spike timing (Hebb, 1949; Sejnowski, 1999), this was not appreciated by a generation of modelers because rate coding was generally accepted as the primary form of information processing, and high-frequency

stimulation protocols were used to induce plasticity at synapses. More recently, the relative timing of spikes has been shown to be critical for the direction and magnitude of synaptic plasticity in the cortex as well as the hippocampus (Markram et al., 1997; Bi and Poo, 1998). Potentiation of a synapse takes place if the presynaptic spike precedes the postsynaptic spike, and depression occurs when presynaptic spike follows the postsynaptic spike. This spike-timing dependent plasticity (STDP) is an asymmetric function of relative spike times in the presynaptic and postsynaptic neurons. The time window for the plasticity can be as short as 10 ms and as long as 100 ms, depending on the synapse.

A natural application for STDP is temporal sequence learning (See Chapters 1.34, 1.35). If neurons are activated in a sequential manner then, due to the asymmetry of the learning rule, synapses from previously activated neurons to following active neurons will be strengthened. For example, such a spike-timing dependent learning algorithm has been used to train a network to link sequential hippocampal place cells while a rat navigates a maze (Blum and Abbott, 1996). The goal was to predict the direction of a future motion on the basis of a previous experience. Asymmetric synaptic weights develop in the model because of the temporal asymmetry of LTP induction and because place fields are activated sequentially during locomotion. This learning algorithm closely resembles the STDP learning rule. The only essential difference is time scale, which in the model was 200 ms, longer than the STDP windows found in cortical or hippocampal neurons.

This model of a navigational map was based on three observations. First, NMDA-dependent LTP in hippocampal slices occurs only if presynaptic activity precedes postsynaptic activity by less than approximately 200 ms. Presynaptic activity following postsynaptic firing produces either no LTP or long-term depression (LTD). Second, place cells are broadly tuned and make synaptic connections with each other both within the CA3 region and between CA3 and CA1. Third, a spatial location can be determined by appropriately averaging the activity of an ensemble of hippocampal place cells. These three observations imply that when an animal travels through its environment, causing different sets of place cells to fire, information about both temporal and spatial aspects of its motion will be reflected in changes of the strengths of synapses between place cells. Because this LTP affects a subsequent place cell firing, it can shift the spatial location coded by the

place cell activity. These shifts suggest that an animal could navigate by heading from its present location toward the position coded by the place cell activity. To illustrate both how a spatial map arises and how it can be used to guide movement, these ideas were applied to navigation in the Morris maze. The network was trained using this spike-timing dependent learning algorithm to form a direction map, which improved with training.

Timing is important in auditory processing, and a number of perceptual tasks, such as sound localization, explicitly use temporal information. Sound localization is important to the survival of many species, in particular to those that hunt in the dark. Interaural time differences (ITD) are often used as a spatial cue. However, the question of how temporal information from both ears can be transmitted to a site of comparison, where neurons are tuned to ITDs, and how those ITD-tuned neurons can be organized in a map remains unclear. A network model based on STDP can successfully account for a fine precision of barn owl sound localization (Kempler et al., 2001). The model converts ITDs into a place code by combining axonal delay lines from both ears and STDP in synapses with distributed delays. The neurons are organized as a single-layer network for each frequency and receive inputs from both ears through axonal arbors. The axons have different time delays. After training, each neuron adjusts its connections to axons with the appropriate time delays in agreement with the neuron's spatial position. In this way, a map with neurons tuned to particular ITDs can be formed.

There is an interesting connection between STDP and TD learning at the computational level (Rao and Sejnowski, 2003). If, consistent with TD learning, synaptic weights between Hodgkin-Huxley type spiking neurons are updated based on the difference in the postsynaptic voltage at time $t + \Delta t$ and at time t , where t is the time when the presynaptic neuron fired a spike, and Δt is a fixed time interval, then the learning rule resembles the conventional STDP learning rule. Networks with this spike-dependent TD learning rule are able to learn and predict temporal sequences, as demonstrated by the development of direction selectivity in a recurrent cortical network. The network consisted of a single chain of recurrently connected excitatory neurons. Each neuron initially received symmetric excitatory and inhibitory inputs of the same magnitude. For training, the neurons in the network were exposed to 100 trials of retinotopic sensory inputs consisting of moving pulses of excitation in the rightward direction.

The effect of learning on the network was in developing a profound asymmetry in the pattern of excitatory connections from preceding and successor neurons. The synaptic conductances of excitatory connections from the left side were strengthened, whereas the ones from the right side were weakened. Because neurons on the left side fired (on average) a few milliseconds before a considered neuron, whereas neurons on the right side fired (on average) a few milliseconds after, as a result, the synaptic strengths of connections from the left side were increased, whereas the synaptic strengths for connections from the right side were decreased. As expected from the learned pattern of connections, the neuron responded vigorously to rightward motion but not to leftward motion.

To investigate the question of how selectivity for different directions of motion may emerge simultaneously, they also simulated a network comprising two parallel chains of neurons, with mutual inhibition between corresponding pairs of neurons along the two chains. As in the previous simulation, a given excitatory neuron received both excitation and inhibition from its predecessors and successors. To break the symmetry between the two chains, they provided a slight bias in the recurrent excitatory connections, so that neurons in one chain fired slightly earlier than neurons in the other chain for a given motion direction. To evaluate the consequences of spike-based TD learning in the two-chain network, the model neurons were exposed alternately to leftward- and rightward-moving stimuli for a total of 100 trials. As in the previous simulation, the excitatory and inhibitory connections to a neuron in one chain showed asymmetry after training, with stronger excitatory connections from the left neurons and stronger inhibitory connections from the right neurons. A corresponding neuron in the other chain exhibited the opposite pattern, and as expected from the learned patterns of connectivity, neurons in one chain were selective to rightward motion, and neurons in the other chain were selective to the leftward motion. This explanation was consistent with the development of directionally selective neurons in the visual cortex of kittens.

1.34.7 Plasticity of Intrinsic Excitability

Several lines of evidence argue for the presence of activity-dependent modification of intrinsic neuronal

excitability during development and learning (Daoudal and Debanne, 2003; *See* Chapter 4.40). In the dentate gyrus of the hippocampus, for example, in addition to homosynaptic LTP of excitatory synaptic transmission, the probability of discharge of the postsynaptic neurons to a fixed excitatory synaptic input is enhanced by high-frequency stimulation (HFS, 100 Hz) of the afferent fibers (Bliss et al., 1973). This second component has been called excitatory postsynaptic potential (EPSP)-to-spike potentiation (E-S potentiation) (Frick et al., 2004). Synaptic plasticity (LTP) and nonsynaptic E-S potentiation are complementary. As in LTP, E-S potentiation requires the activation of NMDA receptor (NMDAR) for its induction. These two forms of plasticity may share common induction pathways. In a recent study of deep cerebellar nuclei neurons, tetanization of inputs to these neurons produces a rapid and long-lasting increase in intrinsic excitability that depends on NMDAR activation (Aizenman and Linden, 2000). These studies suggest that plasticity of intrinsic excitability may be important in developmental plasticity and information storage.

Another form of plasticity in intrinsic excitability has been demonstrated in spontaneously firing vestibular nucleus neurons, which may be responsible for learning of the vestibuloocular reflex. Purkinje cells, which are inhibitory, contact a subset of the neuron in the vestibular nucleus, which receive direct vestibular input and project to the oculomotor nuclei. Brief periods of synaptic inhibition or membrane hyperpolarization produced a dramatic increase in both spontaneous firing rate and responses to intracellularly injected current (Gittis and du Lac, 2006). A similar change occurred after silencing the vestibular nerve. Neurons in the vestibular system fire at remarkably high rates in the intact animal, with resting rates on the order of 50–100 spikes/s and responses to head movements ranging up to 300 spikes/s. Loss of peripheral vestibular function silences the vestibular nerve, resulting in a significant loss of spontaneous firing in the neurons of the vestibular nucleus, which then returns to control values within about a week, even in the absence of vestibular nerve recovery. This plasticity of intrinsic excitability could potentially contribute either to adaptive changes in vestibular function during recovery from peripheral damage or to oculomotor learning in intact animals.

A similar phenomenon has been demonstrated in cultured neocortical pyramidal neurons (Desai et al., 1999). Prolonged activity blockade lowers the threshold for spike generation, and neurons fire at a higher

frequency for any given level of current injection. These changes occurred through selective modifications in the magnitude of voltage-dependent currents: sodium currents increase and persistent potassium currents decrease, whereas calcium currents and transient potassium currents are unaltered. Increase of neuronal excitability in response to reduced activity may contribute to the activity-dependent stabilization of firing rates. The stability in neuronal firing rates is maintained through many mechanisms, and regulation of neuronal excitability may be one of them.

Information about the outside world is transformed into spike trains in the nervous system. How do the neurons learn to represent the information, and do they change their behavior based on changing external stimuli? In the discussion of unsupervised learning and the ICA algorithm, it was shown that information theoretical approaches can be effective in solving real-world problems. A similar information theoretical approach can be implemented to search for an optimal representation. A Hodgkin–Huxley type model of a neuron that can adjust its membrane conductances to maximize information transfer has been proposed (Stemmler and Koch, 1999). The slope of the neuronal gain function should line up with the peak of the input to maximize information transfer. The learning rules they implemented in the model performed this matchup by adjusting the membrane conductances. The conductance modulations did not require calculation of mutual information but were based solely on local characteristics of the neuron. They showed that for different input distributions the model could successfully line up the gain function and the input distributions leading to maximization of information transfer. Thus, the ability of activity-dependent selective modification of the gain functions based on the active balance of inward and outward ion channels could serve a number of important functions, including fine-tuning of the output properties of neurons to match the properties of their inputs.

Plasticity of intrinsic excitability can also participate in regulating the conventional synaptic plasticity. For details, see the previously discussed model, which combines Hebbian and supervised learning (Swinehart and Abbott, 2005), in the section titled ‘Supervised learning.’

1.34.8 Homeostatic Plasticity

Correlation-based Hebbian plasticity is thought to be crucial for information storage because it produces associative changes in the strength of individual

synaptic connections. However, correlation-based learning in neural networks can be unstable. According to the Hebb rule, if a presynaptic neuron participates in firing of a postsynaptic neuron, it leads to strengthening the synapses between the neurons. This makes it more likely that next time the presynaptic neuron fires, it will cause firing in the postsynaptic neuron, which leads to further strengthening of the synapse. Simple associative Hebbian algorithm causes instability in the network by increasing the total activity of the network and losing selectivity among synapses. To keep the network stable and maintain the selectivity of the network, an additional mechanism must stabilize the properties of neuronal networks.

Homeostatic plasticity is a mechanism by which the neurons regulate the network's activity (Turrigiano and Nelson, 2000). There are many different ways neural activities could be regulated to keep them within a functional dynamical range. One mechanism that could maintain relatively constant activity levels is to increase the strength of all excitatory connections into a neuron in response to a prolonged drop in firing rates, and vice versa. This form of homeostatic plasticity is called synaptic scaling.

Regulating synaptic strength is not the only mechanism by which homeostatic activity can be maintained. Previously discussed plasticity of intrinsic excitability also contributes to the homeostatic regulation by controlling the firing rates of the neurons.

All theoretical models implementing associative Hebbian learning rule have to deal with the instability problem. For example, the BCM learning rule deals with unconstrained growth of synaptic weights by dynamically adjusting the threshold between potentiation and depression (Bienenstock et al., 1982). This algorithm is biologically plausible and reflects experimental findings indicating that calcium level is crucial for the direction of plasticity. The dynamical threshold modulation implemented in the BCM rule not only prevents the synapses from unconstrained growth but also maintains the activity level of the units at the appropriate value (See Chapters 1.33, 1.35).

In the next section we present some other examples of learning algorithms involving homeostatic plasticity as a critical element of learning.

1.34.9 Complexity of Learning

The learning paradigms discussed earlier were based on a single mechanism for plasticity (e.g., STDP versus homeostatic and synaptic versus intrinsic

neuronal). However, many difficult tasks cannot be solved using a single learning rule, but require combinations of several learning rules working together. Another essential element of modeling learning processes is the time scale of learning. There are multiple time scales for plasticity, from milliseconds to years, and depending on the demands of the task, different mechanisms for plasticity with different time scales may be involved.

Long-term memory is vulnerable to degradation from passive decay of the memory trace and ongoing formation of new memories. Memory based on synapses with two states shows exponential decay, but experimental data shows that forgetting (memory degradation) follows a power law. A cascade model was developed to address this problem (Fusi et al., 2005). In the model, synapses had two states, weak and strong, but in addition to transition between these two states, there were metaplastic transitions within each state. Based on the stage of metaplasticity, the synapses showed the range of behavior from being highly plastic to being resistant to any plasticity at all. The metaplastic transitions effectively introduced multiple time scales into the model.

The cascade model outperformed alternative models and exhibited a power law for the decay of memory as a function of time. The dependence of memory lifetime on the number of synapses in the model is also a power law function. Memory lifetimes diminish when the balance between excitation and inhibition is disturbed, but the effect is much less severe in the cascade model than in noncascade models.

The function of homeostatic plasticity is to maintain the activity of the cortex at a functional level. But are there any other computational or functional advantages of such plasticity? One study has shown that a combination of Hebbian and homeostatic plasticity can lead to temporal sharpening in response to multiple applications of transient sensory stimuli (Moldakarimov et al., 2006). The model included two types of homeostatic mechanisms, fast and slow. Relatively fast plasticity was responsible for maintaining the average activity of the units. To maintain activity in the excitatory neurons at a target homeostatic level, they implemented a learning rule, according to which inhibitory connections have been adjusted. The slow plasticity was used to determine the value of the target average activities. Thus, the model had three time scales for synaptic adjustments: Hebbian, fast homeostatic, and slow homeostatic mechanisms. Repeated presentations of

a transient signal taught the network to respond to the signal with a high amplitude and short duration, in agreement with experimental findings. This sharpening enhances the processing of transients and may also be relevant for speech perception.

A standard approach in models of self-organized map (SOM) formation is the application of Hebbian plasticity augmented with a mechanism of weight normalization. A conventional way to normalize weights is based on a sum of weights coming into each neuron: The soma collects information on every weight, sums them, and then decides on the amount of normalization. An alternative approach to weight normalization has been proposed (Sullivan and de Sa, 2006). The normalization algorithm did not need information from every synapse but rather was based on the average activities of the units and homeostatic plasticity. When Hebbian and homeostatic mechanisms were combined, the average activities of the units were better maintained compared to the standard Hebbian models.

Dimensionality reduction facilitates the classification, the visualization, and the storage of high-dimensional data. A simple and widely used method is PCA, which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. A new deep network model has been proposed to transform the high-dimensional data into a low-dimensional code (Hinton and Salakhutdinov, 2006). The adaptive multilayer network consisted of two subnetworks, an encoder and decoder. The encoder transformed high-dimensional data into a low-dimensional code. The code layer was then used as the input layer to the decoder network to reconstruct the original input pattern.

The two networks were trained together to minimize the discrepancy between the original data and its reconstruction. The required gradients were obtained using the chain rule to back-propagate error derivatives, first through the decoder network and then through the encoder network. In general, it is difficult to optimize the weights in a multilayer network with many hidden layers. Large initial weights typically lead to poor local minima; with small initial weights, the gradients in the early layers are tiny, making it impossible to train. But if the initial weights are close to a good solution, gradient descent back-propagation works well. A good initial network was obtained with unsupervised learning based on Restricted Boltzmann Machine (RBM) learning algorithm. First, the input layer of the

multilayer network was used as a visible layer of RBM, and the next layer served as a feature layer. After learning one layer of feature detectors, the weights were fixed and used for learning a second layer of feature detectors. This layer-by-layer learning was repeated many times. After pretraining multiple layers of feature detectors, the model was unfolded to produce the encoder and decoder networks that initially used the same weights. The global fine-tuning stage used back-propagation through the whole network to adjust the weights for optimal reconstruction.

They applied the algorithm to multiple tasks including handwritten digits visualization, grayscale images, and documents generalization. In all these tasks, the new algorithm outperformed different approaches based on PCA and other supervised algorithms.

1.34.10 Conclusions

We have discussed learning rules and learning algorithms designed for neural network models and described some problems that can be solved by neural networks with modifiable connections. Neural computation is a broad field that continues to grow; only a few selected studies have been used to illustrate general principles.

Although early modeling efforts focused mainly on traditional synaptic plasticity, such as LTP and LTD, relatively new homeostatic plasticity mechanisms are also being explored. Although synaptic plasticity was once presumed to be the primary neural mechanism of learning, recent models have incorporated changes of intrinsic properties of the neurons as well.

Most experimental studies of learning have studied the mechanisms of synaptic plasticity in reduced preparations. Recently the focus has shifted to relating the changes in the synapses with behavioral learning. For example, inhibitory avoidance learning in rats produced the same changes in hippocampal glutamate receptors as induction of LTP with HFS (Whitlock et al., 2006). Because the learning-induced synaptic potentiation occluded HFS-induced LTP, they concluded that inhibitory avoidance training induced LTP in hippocampus.

Theoretical approaches can integrate local mechanisms with whole system behavior. Even after locating particular sites where changes occur, it is still not clear to what degree those changes are

directly related to the learning. Building a computational model that integrates learning mechanisms allows one to evaluate the importance of different sites of plasticity. The observed plasticity for some sites may be secondary, or compensatory to the primary sites of learning (Lisberger and Sejnowski, 1992).

References

- Aizenman CD and Linden DJ (2000) Rapid, synaptically driven increases in the intrinsic excitability of cerebellar deep nuclear neurons. *Nature Neurosci.* 3: 109–111.
- Bell AJ and Sejnowski TJ (1995) An information maximization approach to blind separation and blind deconvolution. *Neural Comput.* 7: 1129–1159.
- Bi G-Q and Poo M-M (1998) Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18: 10464–10472.
- Bienenstock E, Cooper LN, and Munro PW (1982) Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* 2: 32–48.
- Bliss TV and Lomo T (1973) Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *J. Physiol.* 232: 331–356.
- Bliss TV, Lomo T, and Gardner-Medwin AR (1973) Synaptic plasticity in the hippocampal formation. In: Ansell G and Bradley PB (eds.) *Macromolecules and Behavior*, pp. 192–303. London: Macmillan.
- Blum KI and Abbott LF (1996) A model of spatial map formation in the hippocampus of the rat. *Neural Comput.* 8: 85–93.
- Daoudal G and Debanne D (2003) Long-term plasticity of intrinsic excitability: Learning rules and mechanisms. *Learn. Mem.* 10: 456–465.
- Desai NS, Rutherford LC, and Turrigiano GG (1999) Plasticity in the intrinsic excitability of cortical pyramidal neurons. *Nature Neurosci.* 2: 515–520.
- Doya K and Sejnowski TJ (2000) A computational model of avian song learning. In: Gazzaniga MS (ed.) *The New Cognitive Neurosciences*, 2nd edn., p. 469. Cambridge, MA: MIT Press.
- Ferster D and Miller KD (2000) Neural mechanisms of orientation selectivity in the visual cortex. *Annu. Rev. Neurosci.* 23: 441–471.
- Fiete IR, Hahnloser RHR, Fee MS, and Seung HS (2004) Temporal sparseness of the premotor drive is important for rapid learning in a neural network model of birdsong. *J. Neurophysiol.* 92: 2274–2282.
- Frick A, Magee J, and Johnston D (2004) LTP is accompanied by an enhanced local excitability of pyramidal neuron dendrites. *Nature Neurosci.* 7: 126–135.
- Fusi S, Drew PJ, and Abbott LF (2005) Cascade models of synaptically stored memories. *Neuron* 45: 599–611.
- Gittis AH and du Lac S (2006) Intrinsic and synaptic plasticity in the vestibular system. *Curr. Opin. Neurobiol.* 16: 385–390.
- Hebb DO (1949) *Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.
- Hinton GE and Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313: 504–507.

- Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* 79: 2554–2558.
- Jung T-P, Makeig S, McKeown MJ, Bell AJ, Lee T-W, and Sejnowski TJ (2001) Imaging brain dynamics using independent component analysis. *Proc. IEEE* 89: 1107–1122.
- Kelso SR, Ganong AH, and Brown TH (1986) Hebbian synapses in hippocampus. *Proc. Natl. Acad. Sci. USA* 83: 5326–5330.
- Kempster R, Leibold C, Wagner H, and van Hemmen JL (2001) Formation of temporal-feature maps by axonal propagation of synaptic learning. *Proc. Natl. Acad. Sci. USA* 98: 4166–4171.
- Klopf AH (1982) *The Hedonistic Neuron: A Theory of Memory, Learning and Intelligence*. Washington, DC: Hemisphere.
- Kohonen T (1982) Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 43: 59–69.
- Lee T-W (1998) *Independent Component Analysis. Theory and Applications*. Boston: Kluwer Academic Publishers.
- Lisberger SG and Sejnowski TJ (1992) Motor learning in a recurrent network model based on the vestibule-ocular reflex. *Nature* 360: 159–161.
- Markram H, Lubke J, Frotscher M, and Sakmann B (1997) Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 275: 213–215.
- Miller KD and Mackay DJC (1994) The role of constraints on Hebbian learning. *Neural Comput.* 6: 100–109.
- Moldakarimov SB, McClelland JL, and Ermentrout GB (2006) A homeostatic rule for inhibitory synapses promotes temporal sharpening and cortical reorganization. *Proc. Natl. Acad. Sci. USA* 103: 16526–16531.
- Rao RP and Sejnowski TJ (2003) Self-organizing neural systems based on predictive learning. *Philos. Trans. R. Soc. Lond.* 361: 1149–1175.
- Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65: 386–408.
- Rumelhart DE and McClelland JL (1986) *Parallel Distributed Processing, Vol. 1: Foundations*. Cambridge, MA: MIT Press.
- Sejnowski TJ (1999) The book of Hebb. *Neuron* 24: 773–776.
- Sejnowski TJ and Tesauro G (1988) The Hebb rule for synaptic plasticity: Algorithms and implementations. In: Byrne J and Berry WO (eds.) *Neural Models of Plasticity*, pp. 94–103. New York: Academic Press.
- Seung HS (2003) Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron* 40: 1063–1073.
- Stemmler M and Koch C (1999) How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. *Nature Neurosci.* 2: 521–527.
- Sullivan TJ and de Sa VR (2006) Homeostatic synaptic scaling in self-organizing maps. *Neural Networks* 19: 734–743.
- Sutton RS and Barto AG (1998) *Reinforcement Learning. An Introduction*. Cambridge, MA: MIT Press.
- Swindale NV (1996) The development of topography in the visual cortex: A review of models. *Network* 7: 161–247.
- Swinehart CD and Abbott LF (2005) Supervised learning through neuronal responses modulation. *Neural Comput.* 17: 609–631.
- Tesauro G (1995) Temporal difference learning and TD-Gammon. *Comm. ACM* 38: 58–68.
- Tesauro G and Sejnowski TJ (1989) A parallel network that learns to play backgammon. *J. Artif. Intell.* 39: 357–390.
- Turrigiano GG and Nelson SB (2000) Hebb and homeostasis in neural plasticity. *Curr. Opin. Neurobiol.* 10: 358–364.
- Whitlock JR, Heynen AJ, Shuler MG, and Bear MF (2006) Learning induces long-term potentiation in the hippocampus. *Science* 313: 1093–1097.