

## 1.32 Neural Computation Theories of Learning<sup>☆</sup>

**Samat Moldakarimov and Terrence J Sejnowski**, University of California – San Diego, La Jolla, CA, United States; and Salk Institute for Biological Studies, La Jolla, CA, United States

© 2017 Elsevier Ltd. All rights reserved.

1.32.1	Introduction	579
1.32.2	Hebbian Learning	580
1.32.3	Unsupervised Learning	581
1.32.4	Supervised Learning	581
1.32.5	Reinforcement Learning	583
1.32.6	Spike Timing–Dependent Plasticity	584
1.32.7	Plasticity of Intrinsic Excitability	586
1.32.8	Homeostatic Plasticity	586
1.32.9	Complexity of Learning	587
1.32.10	Conclusions	588
References		588

### 1.32.1 Introduction

The anatomical discoveries in the 19th century and the physiological studies in the 20th century showed that the brain was made of networks of neurons connected together through synapses (Kandel et al., 2012). These discoveries led to a theory that learning could be the consequence of changes in the strengths of the synapses (Hebb, 1949). The Hebb's rule for synaptic plasticity states that:

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, where high-frequency stimulation (HFS) of a presynaptic neuron causes long-term potentiation (LTP) in the synapses connecting it to the postsynaptic neurons (Bliss and Lomo, 1973). LTP takes place only if the postsynaptic cell is also active and sufficiently depolarized (Kelso et al., 1986). In many brain areas, 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.

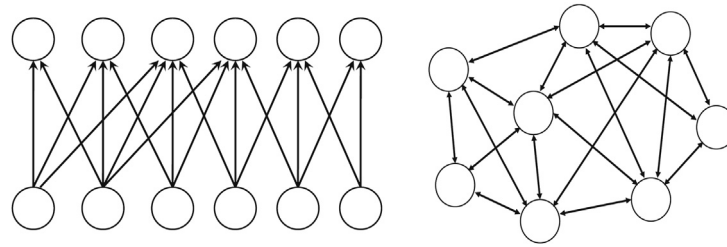
Hebb's rule has served as the starting point for studying the learning capabilities of artificial neural networks (ANN) and for the theoretical analysis and computational modeling of biological neural systems (Hertz et al., 1991). The architecture of an ANN determines its behavior and learning capabilities. 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. Two general classes of network models have feedforward and recurrent architectures.

The simplest feedforward network has one layer of input units and one layer of output units (Fig. 1, left). All connections are unidirectional and project from the input units to the output units. Perceptron is an example of such a simple feedforward network (Rosenblatt, 1958). It can learn to classify patterns from examples. It turned out that 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 (Rumelhart et al., 1986). Multilayer feedforward networks can solve some difficult problems (Rumelhart and McClelland, 1986) and underlie current rapid development of the field of deep learning in machine learning (LeCun et al., 2015).

In contrast to strictly feedforward network models, recurrent networks also have feedback connections among units in the network (Fig. 1, right). A simple recurrent network can have a uniform architecture such as all-to-all connectivity combined with symmetrical weights between units as in Hopfield network (Hopfield, 1982), or it can be a network with specific connections designed to model a particular biological system (Sporns, 2010).

<sup>☆</sup>*Change History:* November 2016. S Moldakarimov and TJ Sejnowski updated the text and references to this entire chapter.

This is an update of S.B. Moldakarimov and T.J. Sejnowski, 1.34 – Neural Computation Theories of Learning, In *Learning and Memory: A Comprehensive Reference*, edited by John H. Byrne, Academic Press, Oxford, 2008, Pages 667–679.



**Figure 1** Network architectures. Left: feedforward network. Right: recurrent network. *Open circles* represent neuronal units and *arrowhead lines* represent synaptic connections.

Modeling learning processes in neural networks implies that strengths of connections are adjusted according to some dynamical equations. Other parameters that may change include spiking thresholds of units, various time constants, and the other dynamical variables of units and connections. Thus, a learning rule is a set of equations that governs changes in the parameters of the network. We will discuss some major categories of learning in neural network models: Hebbian, unsupervised, supervised, and reinforcement. Hebbian and unsupervised learning rules require no teaching signal. Supervised learning rules requires a teaching signal, which provides detailed information on the desired values of the output units of the network, and parameters are adjusted based on discrepancy between the actual output and the desired one. Reinforcement learning is also an error-correcting learning that involves a single scalar feedback signal about the overall performance of the network, and thus it requires less-detailed information than supervised learning.

For simple problems, it is possible to come up with a neural network model that includes only one type of learning rule, but in practice, a real-world problem may require complex models that involve a combination of several different learning rules. Thus, 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.

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.

### 1.32.2 Hebbian Learning

Implementations of the Hebb's rule can take different forms (Sejnowski and Tesauro, 1988). A simple associative Hebbian learning is based on coactivation of presynaptic and postsynaptic neurons (Dayan and Abbott, 2001). The dynamics of Hebbian learning is governed by the following 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 the parameter  $\alpha$  represents the rate of learning.

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

Cortical neurons respond selectively to particular features of stimuli (Hubel and Wiesel, 1959). 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 some 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 input selectivity will emerge. To develop input selectivity, some dependencies among weights are needed, so that changes at one connection will influence the others. There are many different ways to introduce weight dependencies. One approach is to introduce weight normalization (Miller and Mackay, 1994). A different approach, based on competition among input patterns, called BCM rule (after 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. 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—retinotopic location, ocular dominance, orientation preference—are considered simultaneously. To deal with such problems a more abstract class of models was developed by Kohonen (1982). The Kohonen self-organizing map algorithm is usually applied to two-layer networks with feedforward connections from an input layer to an output layer. The input layer is  $N$  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 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 who 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 reduces as learning proceeds. This allows the network to organize a map rapidly and then refine it more slowly with subsequent learning. Models based on 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.

### 1.32.3 Unsupervised Learning

The goal of unsupervised learning is to discover the statistical structure in unlabeled input data. A popular 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  $M$ -dimensional representation ( $M < N$ ) 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. The standard algorithm for computing the principal components of a data set requires finding the eigenvectors of a matrix.

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 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 out of many speakers. The human brain can, to some extent, solve this auditory source separation problem using directional tuning from two ears and knowledge of the speaker, such as dominant frequency bands, 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 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, ICA finds a nonorthogonal linear coordinate system such that the resulting signals are as statistically independent from each other as possible using higher order correlations.

One approach to BSS, called infomax, 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 proved to be highly effective in the blind separation and deconvolution of hundreds of non-Gaussian sources (Bell and Sejnowski, 1995). Several other ICA algorithms have been developed (Hyvärinen and Oja, 2000).

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. EEG signals are generated by underlying components of brain activity, such as 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 filters for representing images, audio recordings, and other kinds of data. Standard linear transformations used in image and auditory processing, such as the Fourier transforms, cosine transforms, etc., may not be optimal but it would be useful to find the most efficient linear transformation based on the statistics of the data, to optimally compress the data.

### 1.32.4 Supervised Learning

Consider the problem of learning to retrieve an output pattern given an input pattern. To remember the patterns the associative 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 weight normalization algorithm as in case of unsupervised learning algorithms (Miller and Mackay, 1994).

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 using information of the actual performance of the network is

by supervised learning. Supervised learning requires a teaching feedback signal, which provides detailed information of the desired outputs of the network, and adjust the connections based on discrepancy between the actual outputs and the desired ones.

The perceptron uses a supervised learning rule to learn to classify input patterns (Rosenblatt, 1958). It consists of a network with many input units, one layer of weights and one output unit that can classify inputs patterns into two groups. The associative Hebbian learning solves the problem, but a perceptron using 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 rule 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,

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

This 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 a network output and the desired one. If input patterns are linearly separable, then this learning rule is guaranteed to find a set of weights that allow pattern classification.

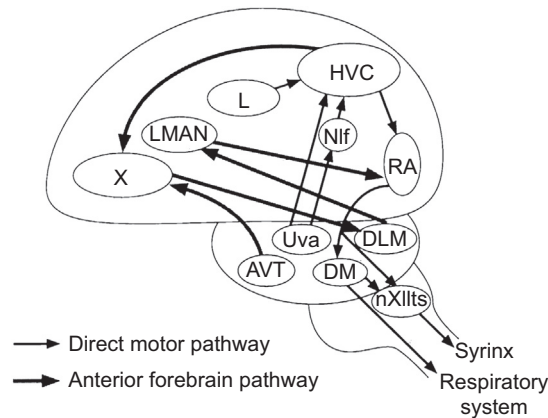
The associative Hebbian learning rule adjusts synaptic weights based on correlations between presynaptic and postsynaptic neurons. This approach is inefficient when the goal of the network is to perform a specific function, rather than simply represent data. To perform a specific task the network should receive some information about the task. Here is an example of how Hebbian plasticity can be incorporated into a supervised learning algorithm: A two-layer network 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. However, 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 perceptron cannot learn higher order problems, but adding additional layers to the feedforward network provides more representational power. New learning algorithms are needed to train a multilayer perceptron. The back-propagation learning algorithm was developed to train multilayer networks (Rumelhart et al., 1986). The learning rule relies on passing the 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 (LeCun et al., 2015).

An example of a multilayer network that was trained using 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. Syringeal and respiratory motoneurons responsible for song production are driven by precisely executed sequences of neural activity in the premotor nucleus robustus archistriatalis (RA) of songbirds (Fig. 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 songbird model was to study the effect of HVC sparseness on the learning speed of the network. They studied multilayer feedforward network with HVC layer that provides input to a “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 burst 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



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

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.

### 1.32.5 Reinforcement Learning

Learning about stimuli or actions based on rewards and punishments is called reinforcement learning (Sutton and Barto, 1998; Dayan and Abbott, 2001). 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 the correct output pattern for every input pattern.

A computational model of birdsong learning based on reinforcement learning has been developed (Doya and Sejnowski, 2000). 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 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. 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 LMAN.

The learning algorithm consisted in making small random changes in the HVC to RA synapses, and keeping the new weights only if overall performance for the entire song, consisting of multiple syllables 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.

When there is a long sequence of steps leading to a reward, it is not an easy problem to decide which step was responsible for success. This is called the temporal credit assignment problem, which slows down learning. 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( $\lambda$ ), in which there is an exponentially decaying feedback of the error in time, so that previous estimates for previous states are also corrected. The timescale of the exponential decay is governed by the  $\lambda$  parameter.



A highly successful application of TD( $\lambda$ ) is TD-Gammon, a network model designed 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. The TD( $\lambda$ ) algorithm computes the reward prediction error at each time step,  $t$ , which is then back-propagated from the output 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 selected is then 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. However, 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.

The power of the TD( $\lambda$ ) algorithm was demonstrated recently by AlphaGo, a neural network system that learned to play Go (Silver et al., 2016). This ancient Chinese game is played on 19-by-19 board and is the most complex board game played by millions. Deep learning, a neural network architecture with many hidden layers in a feedforward hierarchy, learned to compute the value (strength of a board position) and policy (strength of a move from a board position) functions of in Go games from self-play (LeCun et al., 2015). In March 2016, AlphaGo played by Lee Sedol, a 9 dan Go champion, winning four out of five games, an achievement that received worldwide attention. This shows that when reinforcement learning, a relatively weak form of supervised learning, is coupled with powerful representations in cortical hierarchies, performance on complex learning tasks is vastly enhanced. Deep learning coupled with recurrent neural networks that include working memory have led to dramatic advances in natural language processing such as language translation and picture captioning (LeCun et al., 2015).

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.

Stochastic vesicle release was assumed to be the main 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 timescale, the stochastic creation and destruction of synapses. Thus, identifying specific sources of randomness is essential for connecting mathematical models and neurobiology.

### 1.32.6 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. Although Hebb’s formulation implicitly recognized the idea of causality (Hebb, 1949; Sejnowski, 1999; see the Hebb’s rule above), the relative spike timing was not appreciated by generations of modelers because rate coding was generally accepted as the primary form of information processing, and HFS protocols were used to induce plasticity at synapses (Bliss and Lomo, 1973).

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 asymmetrical 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 (Feldman, 2009).

A natural application for STDP is temporal sequence learning. If neurons are activated in 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 spike timing-dependent learning algorithm has been used to train a network to link together 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 based on previous experience. Asymmetrical 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 STDP learning rule. The only essential difference is timescale, 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. Since, this form of 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 STDP learning algorithm to form a direction map, which improved with training.

Timing is important in auditory processing and there are number of perceptual tasks, such as sound localization, that 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. But 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 (Kempster 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 ITD 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 time when presynaptic neuron fired a spike, and  $\Delta t$  is a fixed time interval, then the learning rule resembles the conventional STDP learning rule. This shows that the same algorithm can be implemented in different ways with different timescales.

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 (Shon et al., 2004). The network consisted of a single chain of recurrently connected excitatory neurons. Each neuron initially received symmetrical excitatory and inhibitory inputs of the same magnitude. 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 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, while the ones from the right side were weakened. Because neurons on the left side fired (on average) a few milliseconds before a considered neuron, while neurons on the right side fired (on average) a few milliseconds after; and as a result, the synaptic strengths of connections from the left side were increased, while 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 neurons were exposed alternately to leftward- and rightward-moving stimuli in 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 observed in the visual cortex of kittens.

### 1.32.7 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). 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 HFS (100 Hz) of the afferent fibers (Bliss et al., 1973). This second component has been called 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 NMDAR for its induction. These two forms of plasticity may share common induction pathways. In addition to studies in the hippocampus, a study of deep cerebellar nuclei neurons reported that 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 in 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.

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 an optimal representation. A Hodgkin–Huxley-type model of a neuron that can adjust its membrane conductance 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 matching by adjusting the membrane conductance. 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 (Swinehart and Abbott, 2005) in the [Supervised Learning](#) section.

### 1.32.8 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's rule, if a presynaptic neuron participates in firing of a postsynaptic neuron, it leads to strengthening of the synapses between the neurons. This makes it more likely that next time when 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 there must be an additional mechanism that stabilizes 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. Prolonged activity blockade of cultured neocortical pyramidal neurons lowered the threshold for spike generation, and neurons fired at a higher frequency for any given level of current injection (Desai et al., 1999). These changes occurred through selective modifications in the magnitude of voltage-dependent currents: Sodium currents increase, 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 maybe one of them.



All theoretical models implementing Hebbian learning rules have to deal with the instability problem. For example, the BCM learning rule deals with unconstrained growth of synaptic weights by 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.

In the next section, we present some other examples of learning algorithms that involve homeostatic plasticity as a critical element of learning process.

### 1.32.9 Complexity of Learning

Most of the models discussed earlier are based on a single mechanism for plasticity, e.g., STDP or reinforcement learning. However, many real-world problems cannot be solved using a single learning rule, but require combinations of several learning rules working together. These learning rules could be applied to adjust synaptic weights at different locations, or the same set of synapses could undergo many different types of plasticity.

There could be multiple timescales for synaptic changes and various mechanisms with different timescales might be simultaneously involved. The next study illustrates this multiplicity of plasticity timescales. 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 show exponential decay, but experimental data show that forgetting (memory degradation) follows a power law. A model based on a cascade of plasticity timescales 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 be resistant to plasticity at all. The metaplastic transitions effectively introduced multiple timescales 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 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. 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 repetitive 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. Inhibitory connections were adjusted to maintain activity in the excitatory neurons at a target homeostatic level. Another slow plasticity was used to determine the target average activities. Thus, the model included three types of synaptic plasticity with different timescales: 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, which could be exploited in speech perception.

A standard approach in models of self-organized map formation is the application of Hebbian plasticity augmented with some mechanisms of weight normalization (Kohonen, 1982). 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. In an alternative approach to weight normalization (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, visualization, and storage of high-dimensional data. As aforementioned, 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 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 using on Restricted Boltzmann Machine (RBM) learning algorithm based on a contrastive version of Hebbian learning. First, the input layer of the multilayer network was used as the visible layer of RBM and 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.

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

### 1.32.10 Conclusions

We have discussed learning rules used in neural network models, and we described some tasks that can be solved by neural networks with modifiable parameters. Neural networks is a broad field that continues to grow, and we selected only a few studies to illustrate general principles. Here we would like to emphasize one more time the complexity of mechanisms that may underlie learning in neural networks. Synaptic plasticity was once presumed to be the primary neural mechanism of learning (Hebb, 1949); therefore, early modeling efforts focused mainly on traditional synaptic plasticity, such as LTP and LTD. Moreover, until recently most studies of synaptic plasticity focused mainly on the excitatory connections; however, there is growing evidence that inhibitory connections could be plastic to similar degrees (Gaiarsa et al., 2002; Feldman, 2009). In addition, activity-driven synaptic changes should be balanced by various mechanisms of homeostatic plasticity, including changes both at synapses and of intrinsic properties of the neurons (Turrigiano, 2011).

We focused on models based on feedforward and recurrent network architectures. However, feedback projections are ubiquitous in the brain (Salin and Bullier, 1995; Bullier, 2003; Sillito et al., 2006). Although little is known about functional roles of feedback connections in the brain, recent studies suggest that feedback connections may contribute critically in cortical plasticity (Li et al., 2004; Ramalingam et al., 2013; Moldakarimov et al., 2014).

Most experimental studies of learning have studied mechanisms of plasticity in reduced preparations. Recently the focus has shifted to relating the neural changes 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 the whole-system behavior. Even after locating particular sites where changes may occur, it is still not clear to what degree those changes are directly related to learning. Building a computational model that integrates different plasticity mechanisms allows to evaluate contributions of the plasticity mechanisms into learning. Often, an observed mechanism for plasticity may be secondary, or compensatory to the primary mechanisms of learning (Lisberger and Sejnowski, 1992).

*See also:* 1.13 Birdsong Learning. 1.18 Bat Navigation. 1.25 Localization, Diversity and Behavioral Expression of Associative Engrams in *Drosophila*. 1.28 Operant Behavior in Model Systems. 1.31 Computational Models of Hippocampal Functions. 3.13 Spatial Memory. 4.02 A Comparative Analysis of the Molecular Mechanisms Contributing to Implicit and Explicit Memory Storage in *Aplysia* and in the Hippocampus.

## References

- Aizenman, C.D., Linden, D.J., 2000. Rapid, synaptically driven increases in the intrinsic excitability of cerebellar deep nuclear neurons. *Nat. Neurosci.* 3, 109–111.
- Bell, A.J., Sejnowski, T.J., 1995. An information maximization approach to blind separation and blind deconvolution. *Neural Comput.* 7, 1129–1159.
- Bi, G.-Q., 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.L., Cooper, L.N., 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., 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, T.V., Lomo, T., Gardner-Medwin, A.R., 1973. Synaptic plasticity in the hippocampal formation. In: Ansell, G., Bradley, P.B. (Eds.), *Macromolecules and Behavior*. MacMillan, London, UK, pp. 192–203.
- Blum, K.I., Abbott, L.F., 1996. A model of spatial map formation in the hippocampus of the rat. *Neural Comput.* 8, 85–93.
- Bullier, J., 2003. What is fed back? In: van Hemmen, J.L., Sejnowski, T.J. (Eds.), *23 Problems in Systems Neuroscience*. Oxford UP, New York, pp. 103–132.
- Daoudal, G., Debanne, D., 2003. Long-term plasticity of intrinsic excitability: learning rules and mechanisms. *Learn. Mem.* 10, 456–465.
- Dayan, P., Abbott, L.F., 2001. *Theoretical Neuroscience*. MIT Press.
- Desai, N.S., Rutherford, L.C., Turrigiano, G.G., 1999. Plasticity in the intrinsic excitability of cortical pyramidal neurons. *Nat. Neurosci.* 2, 515–520.
- Doya, K., Sejnowski, T.J., 2000. A computational model of avian song learning. In: Gazzaniga, M.S. (Ed.), *The New Cognitive Neurosciences*, second ed. MIT Press, Cambridge, MA, p. 469.
- Feldman, D.E., 2009. Synaptic mechanisms for plasticity in neocortex. *Annu. Rev. Neurosci.* 32, 33–55.
- Ferster, D., Miller, K.D., 2000. Neural mechanisms of orientation selectivity in the visual cortex. *Annu. Rev. Neurosci.* 23, 441–471.
- Fiete, I.R., Hahnloser, R.H.R., Fee, M.S., Seung, H.S., 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., Johnston, D., 2004. LTP is accompanied by an enhanced local excitability of pyramidal neuron dendrites. *Nat. Neurosci.* 7, 126–135.
- Fusi, S., Drew, P.J., Abbott, L.F., 2005. Cascade models of synaptically stored memories. *Neuron* 45, 599–611.
- Gaiarsa, J.L., Caillard, O., Ben-Ari, Y., 2002. Long-term plasticity at GABAergic and glycinergic synapses: mechanisms and functional significance. *Trends Neurosci.* 25, 564–570.

- Gittis, A.H., du Lac, S., 2006. Intrinsic and synaptic plasticity in the vestibular system. *Curr. Opin. Neurobiol.* 16, 385–390.
- Hebb, D.O., 1949. *Organization of Behavior: A Neuropsychological Theory*. John Wiley and Sons, New York.
- Hertz, J., Krogh, A., Palmer, R.G., 1991. *Introduction to the Theory of Neural Computation*. Westview Press.
- Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 504–507.
- Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* 79, 2554–2558.
- Hubel, D.H., Wiesel, T.N., 1959. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* 148, 574–591.
- Hyvärinen, A., Oja, E., 2000. Independent component analysis: algorithms and applications. *Neural Netw.* 13 (4–5), 411–430.
- Jung, T.-P., Makeig, S., McKeown, M.J., Bell, A.J., Lee, T.-W., Sejnowski, T.J., 2001. Imaging brain dynamics using independent component analysis. *Proc. IEEE* 89, 1107–1122.
- Kandel, E.R., Schwartz, J.H., Jessell, T.M., Siegelbaum, S.A., Hudspeth, A.J., 2012. *Principles of Neural Science*. McGraw-Hill.
- Kelso, S.R., Ganong, A.H., Brown, T.H., 1986. Hebbian synapses in hippocampus. *Proc. Natl. Acad. Sci. U.S.A.* 83, 5326–5330.
- Kempster, R., Leibold, C., Wagner, H., van Hemmen, J.L., 2001. Formation of temporal-feature maps by axonal propagation of synaptic learning. *Proc. Natl. Acad. Sci. U.S.A.* 98, 4166–4171.
- Klopf, A.H., 1982. *The Hedonistic Neuron: A Theory of Memory, Learning and Intelligence*. Hemisphere, Washington, DC.
- Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 43, 59–69.
- LeCun, Y., Bengio, Y., Hinton, G.E., 2015. Deep learning. *Nature* 521, 436–444.
- Lee, T.-W., 1998. *Independent Component Analysis. Theory and Applications*. Kluwer Academic Publisher.
- Li, W., Piëch, V., Gilbert, C.D., 2004. Perceptual learning and top-down influences in primary visual cortex. *Nat. Neurosci.* 7, 651–657.
- Lisberger, S.G., Sejnowski, T.J., 1992. Motor learning in a recurrent network model based on the vestibulo-ocular reflex. *Nature* 360, 159–161.
- Markram, H., Lübke, J., Frotscher, M., Sakmann, B., 1997. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 275, 213–215.
- Miller, K.D., Mackay, D.J.C., 1994. The role of constraints on Hebbian learning. *Neural Comput.* 6, 100–109.
- Moldakarimov, S.B., McClelland, J.L., Ermentrout, G.B., 2006. A homeostatic rule for inhibitory synapses promotes temporal sharpening and cortical reorganization. *Proc. Natl. Acad. Sci. U.S.A.* 103, 16526–16531.
- Moldakarimov, S., Bazhenov, M., Sejnowski, T.J., 2014. Top-down inputs enhance orientation selectivity in neurons of the primary visual cortex during perceptual learning. *PLoS Comput. Biol.* 10, e1003770.
- Rao, R.P., Sejnowski, T.J., 2003. Self-organizing neural systems based on predictive learning. *Philos. Trans. R. Soc.* 361, 1149–1175.
- Ramalingam, N., McManus, J.N., Li, W., Gilbert, C.D., 2013. Top-down modulation of lateral interactions in visual cortex. *J. Neurosci.* 33, 1773–1789.
- Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65, 386–408.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature* 323, 533–536.
- Rumelhart, D.E., McClelland, J.L., 1986. *Parallel Distributed Processing*. In: *Foundations*. A Bradford Book, vol. 1. The MIT Press, Cambridge.
- Salin, P.A., Bullier, J., 1995. Corticocortical connections in the visual system: structure and function. *Physiol. Rev.* 75, 107–154.
- Sejnowski, T.J., 1999. *The book of Hebb*. *Neuron* 24, 773–776.
- Sejnowski, T.J., Tesauro, G., 1988. The Hebb rule for synaptic plasticity: algorithms and implementations. In: Byrne, J., Berry, W.O. (Eds.), *Neural Models of Plasticity*. Academic Press, New York, pp. 94–103.
- Seung, H.S., 2003. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron* 40, 1063–1073.
- Shon, A.P., Rao, R.P.N., Sejnowski, T.J., 2004. Motion detection and prediction through spike-timing dependent plasticity. *Network: Comput. Neural. Syst.* 15, 179–198.
- Sillito, A.M., Cudeiro, J., Jones, H.E., 2006. Always returning: feedback and sensory processing in visual cortex and thalamus. *Trends Neurosci.* 29, 307–316.
- Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D., 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529, 484–489.
- Sporns, O., 2010. *Networks of the Brain*. MIT Press.
- Stemmler, M., Koch, C., 1999. How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. *Nat. Neurosci.* 2, 521–527.
- Sullivan, T.J., de Sa, V.R., 2006. Homeostatic synaptic scaling in self-organizing maps. *Neural Netw.* 19, 734–743.
- Sutton, R.S., Barto, A.G., 1998. *Reinforcement Learning. An Introduction*. A Bradford Book. The MIT Press, Cambridge.
- Swindale, N.V., 1996. The development of topography in the visual cortex: a review of models. *Network* 7, 161–247.
- Swinehart, C.D., Abbott, L.F., 2005. Supervised learning through neuronal responses modulation. *Neural Comput.* 17, 609–631.
- Tesauro, G., 1995. *Temporal Difference Learning and TD-Gammon*, vol. 38. *Communications of the ACM*, pp. 58–68.
- Tesauro, G., Sejnowski, T.J., 1989. A parallel network that learns to play backgammon. *J. Artif. Intell.* 39, 357–390.
- Turrigiano, G.G., Nelson, S.B., 2000. Hebb and homeostasis in neural plasticity. *Curr. Opin. Neurobiol.* 10, 358–364.
- Turrigiano, G., 2011. Too many cooks? Intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement. *Annu. Rev. Neurosci.* 34, 89–103.
- Whitlock, J.R., Heynen, A.J., Shuler, M.G., Bear, M.F., 2006. Learning induces long-term potentiation in the hippocampus. *Science* 313, 1093–1097.