The Ups and Downs of Hebb Synapses

GEOFFREY HINTON
University of Toronto

Abstract
Modelers have come up with many different learning rules for neural networks. When a teacher specifies the correct output, error-driven rules work better than pure Hebb rules in which the changes in synapse strength depend on the correlation between pre and postsynaptic activities. But for unsupervised learning, Hebb rules can be very effective if they are combined with suitable normalization or "unlearning" terms to prevent the synapses growing without bound. Hebb rules that use rates of change of activity instead of activity itself are useful for discovering perceptual invariants and may also provide a way of implementing error-driven learning.

It would be truly wonderful if randomly connected neural networks could turn themselves into useful computing devices by using some simple rule to modify the strengths of synapses. This was the hope that lay behind the original Hebb learning rule and it is the vision that has driven neural network modelers for half a century. Initially, researchers tried simulating various rules to see what would happen. After a decade or two of messing around, researchers realized that there was a much better way to explore the space of possible learning rules: First write down an objective function (a quantitative definition of how well the network is performing) and then use elementary calculus to derive a learning rule that will improve the objective function. For the last few decades, the big theoretical advances in learning rules for neural networks have been associated with new optimization methods and new ideas about what objective function should be optimized.

If we think of a neural network as a device for converting input vectors into output vectors, it is obvious that one sensible objective is to minimize some measure of the difference between the output the network actually produces and the output it ought to produce. This approach led to effective "error-driven" learning rules such as the Widrow-Hoff rule (Widrow & Hoff, 1960) and the perceptron convergence procedure (Rosenblatt, 1961) and it was later generalized to multilayer networks by using backpropagation of the errors to get training signals for intermediate "hidden" layers (Rumelhart, Hinton, & Williams, 1986). Within the neural network community, the "Hebbian" approach of using the product of pre and postsynaptic activities to drive learning was seen as inferior to error-driven methods that use the product of the presynaptic activity and the postsynaptic activity *derivative* – the rate at which the objective function changes as the postsynaptic activity is changed. Even when the task was merely to associate random input vectors with random output vectors, it was shown that an error-driven rule worked much better than a Hebbian rule.

Unfortunately, error-driven learning has some serious drawbacks. It requires a teacher to specify the right answer and it is hard to see how neurons could implement the backpropagation required by multilayer versions. It is possible to get the teaching signal from the data itself by trying to predict the next term in a temporal sequence (Elman, 1991) or by trying to reconstruct the input data at the output (Hinton, 1989) but it is also possible to use quite different objective functions for learning. Some of these alternative objective functions lead to learning rules that are far more Hebbian in flavour.

A common objective in processing high-dimensional data is to reduce the dimensionality without losing the ability to reconstruct the raw data from the reduced representation. If we measure the accuracy of the reconstruction by the squared error, the optimal strategy is to extract the principal components – the dominant directions of variation in the data. Oja (1982) showed how to extract the first principal component using Hebbian learning to maximize the squared output of a neuron combined with normalization of the synapse strengths to prevent them growing without bound. Sanger (1989) showed that lateral inhibition between neurons can be used to make them extract several different principal components.

Another objective that might have appealed to Hebb is to create a set of attractor states in a nonlinear network. Leading researchers (Marr, Palm, & Poggio, 1978) speculated that it would be very hard to analyze and manipulate the dynamical behaviour of networks of binary threshold neurons with recurrent interconnections, but in 1982, Hopfield pointed out that if the connections were symmetrical the network would settle down into states that were local minima of a simple "energy function." Moreover, new minima could be created by simple Hebbian learning. So the activity dynamics of a network with fixed weights could implement the retrieval of a memory from a corrupted or incomplete version of the memory, and Hebbian learning could be used to store new memories.

Hopfield networks introduced an extra level of complexity by using one objective function – the energy – to determine the fast dynamics of the neural activities and a quite different objective function – the proximity of the energy minima to the vectors that need to be stored – to determine the slow dynamics of the synapse strengths. Hinton and Sejnowski (1986) realized that Hopfield nets could be generalized by adding noise to the activity dynamics, so that instead of simply settling to a point attractor, the "Boltzmann machine" would wander around among its various possible activity states spending most of its time in low energy states but occasionally visiting higher energy ones. If the network is divided into a set of "visible" units that represent the sensory input and a set of hidden units whose states represent an interpretation of the sensory input, the stochastic dynamics can be interpreted as a way of sampling various possible interpretations of the sensory data. An interpretation that has energy *E* will get sampled with a probability proportional to *exp(-E)* that corresponds exactly to correct Bayesian inference if the probability of an interpretation is proportional to *exp(-E)*.

The fact that the stochastic dynamics of the activities perform Bayesian inference is appealing but it also justifies a simple Hebbian learning rule. With a sensory input vector clamped on the visible neurons, we run the stochastic activity dynamics for a while and then we increment the weight between any two neurons that are active at the same time. This is the purest form of Hebbian learning and it does not work because the weights just keep growing until all the neurons are turned on all the time. So we add another, anti-Hebbian term to the learning rule. We run the stochastic dynamics without clamping the visible neurons and we decrement the weights between neurons that are simultaneously active. This may look like an optimistic hack designed to fix up an obviously deficient learning rule (Hopfield, Feinstein, & Palmer, 1983) but it is actually exactly the right thing to do if want the learning to minimize a very sensible objective function. We can think of the neural network as having a model that specifies the probability of each possible sensory activity vector and the objective of learning is to make the probabilities in the model match the probabilities with which sensory activity vectors actually occur.

According to the model, the probability of a sensory vector is just the sum of the probabilities of all possible interpretations of that vector. The Hebbian part of the learning rule samples the interpretations in proportion to their probabilities and lowers the energies of the sampled interpretations by increasing the weights between active neurons. The anti-Hebbian part of the rule is required because the probability of a state of the network depends not only on the energy of that state but also on the energy of all the alternative states. So in addition to lowering the energy of a state, it is necessary to raise the energy of all the alternatives, and this is what the anti-Hebbian learning is doing. To summarize: Boltzmann machines can perform unsupervised learning of distributed representations using a simple, local learning rule that combines Hebbian and anti-Hebbian terms and they have a neat mathematical justification in which the learning rule amounts to following the gradient of a sensible objective function.

Boltzmann machines are slow because the stochastic dynamics need to run for a while before interpretations are sampled with the right probability, but it has recently been shown that a modification to the objective function allows the same learning rule to work even when the stochastic dynamics are only run for a couple of time steps (Hinton, 2002).

There is another objective function that has been influential in theoretical neuroscience. Sensory data is highly redundant, so it can be compressed by mapping it into a code in which the individual components of the code vectors are statistically independent (Barlow, 1961). This is obviously sensible for squeezing information through a limited channel such as the optic nerve, and it explains many properties of retinal ganglion cells. But it is also useful for discovering what caused the sensory data if we assume that causes tend to be independent of one another. At first glance, the objective of achieving independence seems very different from the objective of matching the probability distribution defined by a model to the probability distribution of the observed data, but the two are actually equivalent if we treat the observed data as the output of a stochastic generative model that converts configurations of statistically independent causes into sensory data. There is therefore a surprisingly close link between the old psychological idea of analysis-by-synthesis and Barlow's ideas about redundancy reduction. Over the last decade, researchers have discovered efficient ways of performing Independent Components Analysis (ICA) – learning a set of filters that produce statistically independent outputs when they are applied to structured datasets (Bell & Sejnowski, 1995). When ICA is applied to natural images, the filters resemble those found in visual cortex (Olshausen & Field, 1996) and they even arrange themselves into topographic maps if an extra term is added to encourage filters to be physically close to one another if they take on extreme values at the same time (Hyvarinen, Hoyer, & Inki, 2001). Unfortunately, the synaptic learning rule for ICA is not particularly simple because it requires the postsynaptic neuron to know about more than just the activity of the presynaptic neuron.

Minimizing the redundancy between the components of code vectors whilst ensuring that each component varies as the data vary has the effect of maximizing the information that the code vector conveys about the sensory input. An interesting alternative is to maximize the information that one code vector conveys about the next one. This objective function is subtly different from simply trying to predict the next sensory input vector. It encourages the network to extract from the sensory input a representation of those aspects that are predictable and to ignore those aspects that are random. An especially simple version of this approach assumes that some features of the sensory input do not change over short time periods. When observing a moving rigid object, for example, the retinal image changes but the shape of the object does not, so features that act as invariant shape descriptors can be learned by a simple synaptic learning rule that treats the rate of change in the output of a neuron as the error signal (Becker & Hinton, 1992; Foldiak, 1991; Stone, 1996; Wiskott & Sejnowski, 2002). To prevent the weights all becoming zero, it is necessary to add a term that encourages the output to change significantly over longer time periods.

Treating the rate of change of the output as an error derivative solves the problem of how the output of a neuron can communicate both a value and an error derivative with respect to that value. Moreover, it makes it feasible to perform backpropagation of error derivatives in a more realistic way. Suppose there is a backward connection from neuron j to neuron i that has learned to have a weight proportional to the weight of the forward connection. Using this backward connection, the output of neuron j can be provided as additional input to neuron i via a "differential" synapse that injects an amount of charge that is proportional to the rate of change of its presynaptic input. So long as neuron i has a smooth activation function and so long as the top-down effect is small, the additional top-down input to neuron i will cause a change in its output that corresponds to the backpropagated error derivative with respect to the total input. Starting with a representation of the derivative of the error with respect to the total input to unit j, we have computed the derivative of the error with respect to the total input to unit i in the previous layer. The learning rule for the forward connection uses the product of the presynaptic activity with the rate of change of the postsynaptic activity. The learning rule for the backward connection is the same except that the neuron that was postsynaptic is now presynaptic and vice versa. Curiously, recent research on real synapses suggests that modification rules involving temporal derivatives may not be nearly as farfetched as they seemed in the early days of backpropagation.

The theory of learning rules for neural networks has come a long way but it still has a long way to go. Some of the theoretical discoveries fit very nicely with Hebb's original suggestion and some do not. After half a century we can say a lot about the extra terms that are needed to make Hebb rules work and a lot about the objective functions that Hebb rules can optimize, but what really happens at a synapse and why are still a mystery.

Address correspondence to Geoffrey Hinton, Department of Computer Science, University of Toronto, 10 King's College Road, Toronto, Ontario, Canada M5S 3G5.

Résumé

Les modélisateurs ont trouvé de nombreuses règles d'apprentissage différentes pour les réseaux neuraux. Lorsqu'un professeur spécifie le résultat correct, les règles axées sur l'erreur fonctionnent mieux que les règles de Hebb pures où les changements dans la force de la synapse dépendent de la corrélation entre les activités pré et post synaptiques. Cependant, pour l'apprentissage sans supervision, les règles de Hebb peuvent être très efficaces si elles sont combinées à une normalisation convenable ou à des termes de « désapprentissage » pour empêcher que les synapses grandissent sans limite. Les règles de Hebb qui empruntent les taux de changement d'activité plutôt que l'activité en soi sont utiles pour découvrir les invariants perceptifs et peuvent aussi fournir une façon de mettre en œuvre l'apprentissage axé sur l'erreur.

References

Barlow, H. B. (1961). The coding of sensory messages. In W. H. Thorpe & O. L. Zangwill (Eds.), *Current problems in animal behaviour* (pp. 331-360). Cambridge, MA: Cambridge University Press.

Becker, S., & Hinton, G. E. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature, 355,* 161-163.

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation, 7,* 1129-1159.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14,* 179-211.

Foldiak, P. (1991). Learning invariance from transformation sequences. *Neural Computation, 3,* 194-200.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence, 40,* 185-234.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14,* 1771-1800.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations.* Cambridge, MA: MIT Press.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Processing. National Academy of Science USA, 79,* 2554-2558.

Hopfield, J. J., Feinstein, D. I., & Palmer, R. G. (1983). Unlearning has a stabilizing effect in collective memories. *Nature, 304,* 158-159.

Hyvarinen, A., Hoyer, P. O., & Inki, M. (2001). Topographic independent component analysis. *Neural Computation, 13,* 1527-1558.

Marr, D., Palm, G., & Poggio, T. (1978). Analysis of a cooperative stereo algorithm. *Biological Cybernetics, 28,* 223-229.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology, 15,* 267-273.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature, 381,* 607-609.

Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain machines.* Washington, DC: Spartan Books.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323,* 533-536.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks, 2,* 459-473.

Stone, J. V. (1996). Learning perceptually salient visual parameters using spatiotemporal smoothness constraints. *Neural Computation, 8,* 1463-1492.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. In *IRE WESCON Convention Record, Part 4,* 96-104.

Wiskott, L., & Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural Computation, 14,* 715-770.