Learning algorithms and probability distributions in feed-forward and feed-back networks

Article *in* Proceedings of the National Academy of Sciences · January 1988

DOI: 10.1073/pnas.84.23.8429 · Source: PubMed

1 author:

John J Hopfield
Princeton University
233 PUBLICATIONS **74,525** CITATIONS

Proc. Natl. Acad. Sci. USA
Vol. 84, pp. 8429–8433, December 1987
Biophysics

Learning algorithms and probability distributions in feed-forward and feed-back networks

J. J. HOPFIELD

Divisions of Chemistry and Biology, California Institute of Technology, Pasadena, CA 91125, and AT&T Bell Laboratories, Murray Hill, NJ 07974

Contributed by J. J. Hopfield, August 17, 1987

ABSTRACT **Learning algorithms have been used both on feed-forward deterministic networks and on feed-back statistical networks to capture input–output relations and do pattern classification. These learning algorithms are examined for a class of problems characterized by noisy or statistical data, in which the networks learn the relation between input data and probability distributions of answers. In simple but nontrivial networks the two learning rules are closely related. Under some circumstances the learning problem for the statistical networks can be solved without Monte Carlo procedures. The usual arbitrary learning goals of feed-forward networks can be given useful probabilistic meaning.**

Learning algorithms enable model "neural networks" to acquire capabilities in tasks such as pattern recognition or continuous input–output control. *Feed-forward* networks of analog units having sigmoid input–output response have been studied extensively (1–4). These networks are multilayer perceptrons with the two-state threshold units of the original perceptron (5–8) replaced by analog units having a sigmoid response. Another kind of network (9, 10) is based on *symmetrical connections*, an energy function (11), two-state units, and a random process to generate a statistical equilibrium probability of being in various states. Its connection to the physics of a coupled set of two-level units in equilibrium with a thermal bath (like a magnetic system of Ising spins with abitrary exchange) led it to be termed a Boltzmann network.

These networks appear rather different. One is deterministic, the other statistical; one is discrete, the other continuous; one has a one-way flow of information (feed-forward) in operation, the other a two-way flow of information (symmetrical connections). The learning algorithms therefore appear quite different, so much so that comparisons of the computational effort needed to learn a given task for these two kinds of networks have sometimes been made. This paper shows that variants of each of these two classes of networks, adapted to emphasize the meaning of the actual procedure employed, often have very closely related learning algorithms and properties. This view finds useful meaning, in terms of probabilities, for a parameter that has appeared arbitrary in analog perceptron learning algorithms. Some three-layer statistical networks can be solved by gradient descent without the necessity of statistical averaging on a computer.

Task

Consider a set of instances α of a problem. For each instance, input data consist of analog input values I_k^α ($k = 1, \ldots, n$). We are also given a set of propositions ϕ ($\phi = 1, \ldots, m$). We consider problems for which the input data do *not* exactly define the situation. Thus, given the input data for instance α, there is a probability $Q_{+,\phi}^\alpha$, that proposition ϕ is true and a probability $Q_{-,\phi}^\alpha$ that the proposition is false. The object of the network learning is to capture the I^α–$Q_{\pm,\phi}^\alpha$ relationship, which is all the information that is known about the implication of the input instance α. This information can subsequently be used in a variety of modes, of which the simplest would be to choose an action based on maximum likelihood by using these probabilities.

A computational probabilistic approach to a task is exemplified in hidden Markov approaches to speech-to-text conversion (12). The ensemble of speech utterances is described in terms of word models using a Markov description of the possible sound patterns associated with a given word. When a particular utterance is heard, the probability that each word model might generate that sound is evaluated. Sequences of such probabilities can then be used for word selection (13). The problem is intrinsically probabilistic because individual words often cannot be unambiguously understood in a context-free and speaker-independent fashion and because the analysis done may intrinsically ignore evidence necessary to distinguish accurately between similar sounds. A feed-forward network for doing such a task should generate probabilities of the occurrence of words as its outputs.

Both the deterministic and the stochastic networks to be discussed will be given the same task—namely, to capture the probability of the truth of a set of propositions based on a given set of instances by using a learning algorithm. E. Baum and F. Wilczek (personal communication) have considered the utility of learning a probability distribution with an analog perceptron. Anderson and Abrahams (14) have discussed more elaborate uses of probabilities in deterministic networks.

Analog Perceptron

Consider a multilayer, feed-forward analog perceptron. Although what is described in this section can be extended to systems having a large number of layers, we will for simplicity restrict consideration to a system having three layers of analog units and two layers of connections (Fig. 1*a*). The outputs of the first layer are forced by the input data. When input case α is present, the input data are the output of these units k and are given by I_k^α.

The input–output relation of the second and third units are given by output = g(input):

$$g(u) = \tanh\beta u. \qquad [1]$$

The connections T_{bk} connect the outputs of the input units k to the input of the second-layer unit b. Thus the input to unit b for instance α is

$$u_b^\alpha \equiv \sum_k T_{bk} I_k^\alpha \qquad [2]$$

and the output of middle-layer unit b is then $V_b^\alpha = g(\beta u_b^\alpha)$. The third layer of units corresponds to the propositions ϕ

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

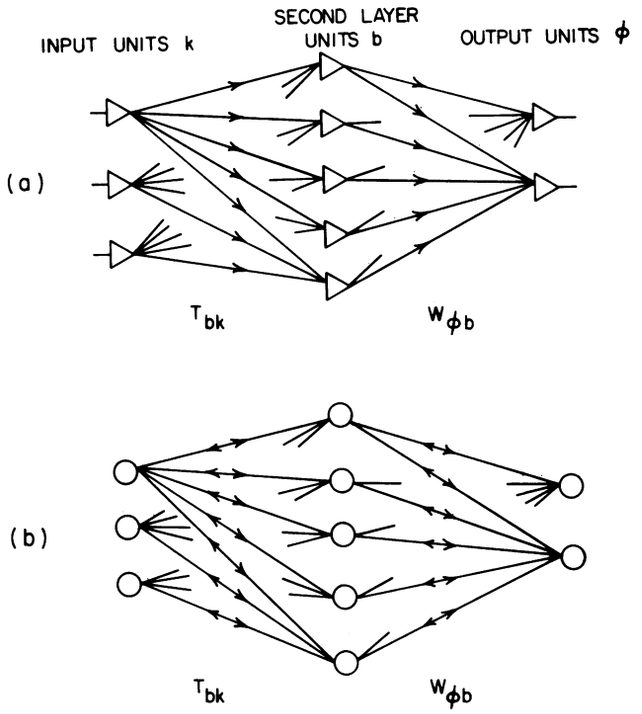

FIG. 1. (*a*) The three-layer analog perceptron. The connections are feed-forward. The second- and third-layer units have sigmoid response. The outputs of the input layer are I_k^α. (*b*) The three-layer Boltzmann network. The connections are of equal strength in the feed-back and feed-forward directions (symmetric). The second- and third-layer units have two-state thermal equilibria. The outputs of the input layer (which is always clamped) are I_k^α.

and is connected to the second layer by connection weights $W_{\phi b}$. The input to a third-layer unit for case α is

$$u_\phi^\alpha \equiv \sum_b W_{\phi b} V_b^\alpha \qquad [3]$$

and unit ϕ has an output $X_\phi^\alpha = g(u_\phi^\alpha)$.

The output X_ϕ^α will be interpreted as follows. Define

$$P_{\pm,\phi}^\alpha \equiv (1 \pm X_\phi^\alpha)/2 = (e^{\pm \beta u_\phi^\alpha})/(e^{\beta u_\phi^\alpha} + e^{-\beta u_\phi^\alpha}). \qquad [4]$$

$P_{+,\phi}^\alpha$ will be assigned a meaning—namely, the probability prediction of the network that proposition ϕ is true, given input case α. $P_{-,\phi}^\alpha$ is the probability that ϕ is false.

A learning algorithm can be constructed with the help of a convex positive function that is minimized if $P_{\pm,\phi}^\alpha = Q_{\pm,\phi}^\alpha$ for all α. The entropy of Q with repect to P is a logarithmic measure often used for comparing probability functions (15):

$$f = \sum_\alpha A^\alpha \sum_\phi \sum_{+,-} Q_{\pm,\phi}^\alpha \ln(Q_{\pm,\phi}^\alpha / P_{\pm,\phi}^\alpha) \geq 0. \qquad [5]$$

A^α is a positive "significance weight" assigned to case α. There is no need that all A^α be the same. This kind of probability measure was used by Ackley *et al.* (10) in developing the Boltzmann learning algorithm and by Baum and Wilczek (personal communication). Since we want to compare analog multilayer perceptron learning with Boltzman machine learning, this is the appropriate optimizing measure to use for the analog perceptron case. It is not the measure that has customarily been used in analog perceptrons, where an arbitrary quadratic criterion of "best" is in general use.

The shape of f can be studied by taking its derivatives with respect to the synaptic weights. This gradient is also of interest for learning by gradient descent on f in the space of connection weights T_{bk} and $W_{\phi b}$. By direct differentiation,

$$\frac{\partial f}{\partial W_{\phi b}} = -\beta \sum_\alpha A^\alpha [(Q_{+,\phi}^\alpha - Q_{-,\phi}^\alpha) - \tanh \beta u_\phi^\alpha] V_b^\alpha, \qquad [6]$$

where $V_b^\alpha = \tanh\left(\beta \sum_k T_{bk} I_k^\alpha\right)$. Similarly,

$$\frac{\partial f}{\partial T_{bk}} = -\beta^2 \sum_\alpha A^\alpha [(Q_{+,\phi}^\alpha - Q_{-,\phi}^\alpha) - \tanh \beta u_k^\alpha] W_{\phi b} I_k^\alpha g'(u_b^\alpha), \qquad [7]$$

where g' is the derivative of g with respect to its argument. When learning by gradient descent is used, data may not be explicitly available about Q^α for each particular case. However, the average over a large training set will implicitly contain the information. Thus, Eqs. **6** and **7** can still be used to capture the real probability distribution when information about Q^α is limited to values of 1 or 0 for particular instances.

The usual analog perceptron approach to the problem of making a decision about the truth or falsity of the various propositions lacks any notion of probability. If the proposition is nominally true for instance α, it would try to train the output unit to an arbitrary target value like $0.8 = S_\phi^\alpha$. Similarly, if the instance is false, the target output would be $S_\phi^\alpha = -0.8$. The cost function usually minimized is

$$G = \frac{1}{2} \sum_\alpha A^\alpha \sum_\phi (S_\phi^\alpha - X_\phi^\alpha). \qquad [8]$$

Direct differentiation yields

$$\frac{\partial G}{\partial W_{\phi b}} = -\beta \sum_\alpha A^\alpha (S_\phi^\alpha - g(\beta u_\phi^\alpha)) V_b^\alpha g'(\beta u_\phi^\alpha). \qquad [9]$$

Eqs. **6** and **9** differ by the factor of $g'(\beta u_\phi^\alpha)$ and the occurrence of the arbitrary S_ϕ^α in place of the meaningful $(Q_{+,\phi}^\alpha - Q_{-,\phi}^\alpha)$. Some differences in meaning will be illustrated in the *Appendix*. The same relation exists between Eq. **7** and $\partial G/\partial T_{bk}$.

Statistical Network

The statistical network (9) considered has two-state units in a thermal statistical ("Boltzmann") environment, with characteristic "temperature" $1/\beta$. The arrangement of the units is as shown in Fig. 1*b*, with the weights now representing two-way symmetric connections between the model units. The topology of connections is identical to that in Fig. 1*a*.

We attack the same problem posed in the previous section, that of capturing the correct probability distribution of output states for all particular input cases α. The input units are always fixed for any paticular case α, and no thermal averaging is ever done on the input units. This ensemble averaging is thus subtly different from that which was chosen by Ackley *et al.* (10). It corresponds more naturally to the way the network is to be queried, namely by fixing the input units and observing the statistics of the output units. The input units then need not be taken as binary but may instead be continuous. They will be held to values I_κ^α. The energy function of the system is then specific to α and is given by

$$E = -\sum_{k,b} \sum I_k^\alpha V_b T_{bk} - \sum_{b,\phi} \sum W_{\phi b} V_b \mu_\phi. \qquad [10]$$

$V_b = \pm 1$ are the variables representing the two states of the second-layer units, while $\mu_\phi = \pm 1$ is the variable represent-

ing the state of the unit corresponding to proposition ϕ. The value of $P^{\alpha}_{+,\phi}$ given by the network is equated to the probability that, for case α, the variable μ_ϕ has the value 1 (averaged over all configurations of the V and μ units). The appropriate function to minimize in the comparison of P and Q is now

$$F = \sum_{\alpha} A^{\alpha} \sum_{\phi} \sum_{+,-} Q^{\alpha}_{\pm,\phi} \ln(Q^{\alpha}_{\pm,\phi}/P^{\alpha}_{\pm,\phi}). \quad [11]$$

The subscript $\pm$ on $Q_\pm$ refers to the cases true and false, and $P_{\pm,\phi}$ are the thermal ensemble average probabilities for the variables μ_ϕ.

Consider first the case of a single output proposition, so that there is no sum on ϕ. Cases of few propositions have been used in many problems, ranging from the assessment of symmetries in spatial patterns (16) to the discrimination between pipes and rocks in sonar (17). Define

$$Z^{\alpha}_{\pm} = \sum_{\substack{\text{all } V \\ \text{configurations}}} \exp\left(+\beta \sum_{k,b} I^{\alpha}_k V_b T_{bk} \pm \beta \sum_b W_{\phi b} V_b\right) \quad [12]$$

(i.e., the partition function for the intermediate-layer "spins" for a fixed final $\mu_\phi = \pm 1$). Then

$$Z^{\alpha} = Z^{\alpha}_{+} + Z^{\alpha}_{-} \quad [13a]$$

$$P^{\alpha}_{\pm,\phi} = Z^{\alpha}_{\pm}/Z^{\alpha} \quad [13b]$$

$$Z^{\alpha}_{\pm} = \prod_b 2 \cosh\left(\beta \sum_k T_{bk} I^{\alpha}_k \pm W_{\phi b}\right) \quad [13c]$$

$$\frac{\partial F}{\partial W_{\phi b}} = -\sum_{\alpha} A^{\alpha} \sum_{+,-} Q^{\alpha}_{\pm,\phi} \left[\frac{\partial(\ln Z^{\alpha}_{\pm})}{\partial W_{\phi b}} - \frac{\partial(\ln Z^{\alpha})}{\partial W_{\phi b}}\right] \quad [13d]$$

$$= -\beta \sum_{\alpha} A^{\alpha} \sum_{+,-} Q^{\alpha}_{\pm,\phi} (\langle \pm V_b \rangle_{\pm} - \langle \mu_\phi V_b \rangle)$$

$$\equiv -\beta \sum_{\alpha} A^{\alpha} (\langle \mu_\phi V_b \rangle_{\text{clamped}} - \langle \mu_\phi V_b \rangle_{\text{free}})$$

A similar expression can be written for $\partial F/\partial T_{bk}$. Exactly as in the Boltzmann machine case, such derivatives can be written as a sum over two ensembles. In the present case, the "free" ensemble has μ_ϕ and V_b both taking on values of ± 1, while for the "clamped" average, μ is assigned a fixed value of + or − with probability $Q^{\alpha}_{\pm,\phi}$ and the V_b averaged over all configurations. The ensemble average in the free case is different from that of Ackley *et al.* (10) in that in our free average the input units are *fixed*, whereas in their free average the input units are also free. The ensemble of Ackley *et al* is particularly appropriate if one wishes also to be able to do reverse inference, inferring "inputs" from "outputs," or to do general pattern completion. When only forward inference is desired, the ensemble defined here is more appropriate.

Using Eq. **13c**, we can also write out these derivatives, obtaining

$$\frac{\partial F}{\partial W_{\phi b}} = -2\beta \sum_{\alpha} A^{\alpha} \sum_{\pm} \left(\pm Q^{\alpha}_{\pm,\phi} \mp \frac{Z^{\alpha}_{\pm}}{Z^{\alpha}_{+} + Z^{\alpha}_{-}}\right) \left[\tanh\left(\beta \sum_k T_{bk} I^{\alpha}_k \pm \beta\, W_{\phi a}\right)\right] \quad [14]$$

and

$$\frac{\partial F}{\partial T_{bk}} = -2\beta \sum_{\alpha} A^{\alpha} \sum_{\pm} \left(Q^{\alpha}_{\pm,\phi} - \frac{Z^{\alpha}_{\pm}}{Z^{\alpha}_{+} + Z^{\alpha}_{-}}\right) \left[I^{\alpha}_k \tanh\left(\beta \sum_k T_{bk} I^{\alpha}_k \pm \beta\, W_{\phi b}\right)\right].$$

All the statistical averaging that normally occurs in Boltzmann (symmetric) network learning has been done explicitly. The amount of computational labor involved in optimizing the weights by gradient descent on F using Eq. **14** is now essentially the same as that which would be required for a feed-forward analog network having the same structure.

Equivalence in the Mean-Field Approximation

The partition functions involved in F correspond to a simple spin system. In circumstances where one spin interacts with many other spins, a mean-field model often gives a good description of the statistical mechanics. When there is only a single ϕ variable, many spins b interact with an external field I and a single spin μ_ϕ. The total field or input acting on the spin μ_ϕ must be only of order $\approx 2/\beta$ if proposition ϕ has a probability between 5% and 95% of being true. If there are many intermediate layer spins b, then a typical $\beta W_{\phi b}$ should be small compared to 1. This will certainly be the case if the information about the probability distribution is delocalized over many (>10) of the intermediate layer units, even if the probabilities approach 1 or 0. In such a case, and in the spirit of the mean-field approximation, the hyperbolic tangents can be expended in powers of W. In lowest order,

$$\frac{\partial F}{\partial W_{\phi b}} = -2\beta \sum_{\alpha} A^{\alpha} \left[\left(Q^{\alpha}_{+,\phi} - Q^{\alpha}_{-,\phi}\right) - \frac{Z^{\alpha}_{+} - Z^{\alpha}_{-}}{Z^{\alpha}_{+} + Z^{\alpha}_{-}}\right] \tanh\left(\beta \sum_k T_{bk} I^{\alpha}_k\right). \quad [15]$$

In the effective field approximation and to lowest order in W, the mean field on spin ϕ will be given by

$$u^{\alpha}_{\phi} = \sum_b W_{\phi b} \langle V^{\alpha}_b \rangle, \quad [16]$$

where

$$\langle V^{\alpha}_b \rangle = \tanh\left(\beta \sum_k T_{bk} I^{\alpha}_k\right) \quad [17]$$

and

$$\frac{Z^{\alpha}_{+} - Z^{\alpha}_{-}}{Z^{\alpha}_{+} + Z^{\alpha}_{-}} \approx \tanh \beta u^{\alpha}_{\phi}. \quad [18]$$

With these substitutions and to zero-order in W, Eq. **15** is the same as Eq. **6** for the analog perceptron case except for a trivial scale factor of 2.

A similar result can be obtained for $\partial F/\partial T_{bk}$. In this case, the term of zero-order in W vanishes, so an expansion must be made in the right-hand square bracket of Eq. **14** to include the term first-order in W. The expression for $\partial F/\partial T_{bk}$ is then the same as Eq. **7** except for the same scale factor of 2. Thus the terrain in "weight space" of the Boltzmann machine is essentially the same as that of the analog perceptron and the gradient-descent learning algorithms equivalent. This line of argument is valid when the number of propositions is very small.

Discussion

The case of many output units is more complex and can be simplified in many directions. One rigorous conclusion is that for the case of m output units (propositions), Eq. **14** can be generalized by defining 2^m partial partition functions instead of 2. If m is small, this may still lead to rapid learning by gradient descent compared to a Monte Carlo approach.

(The same is true if only 2^m patterns dominate the outputs, even though there are more output units.)

Analog approximations of Boltzmann networks in the effective-field approximation are useful under broader circumstances. In any large connectivity and symmetric statistical network that is input-dominated and has a single stable solution in the mean-field approximation, the statistical averaging of the statistical networks has little effect except to smooth the mean response of the units, which is more easily done by using analog units and the same connections. The conditions necessary to produce an equivalence between the generalized Δ-rule learning and learning in statistical networks emphasize a circumstance in which a large amount of information from the previous layer is brought to bear on a small set of propositions. The ability to make an expansion in powers of matrix elements will follow in most networks that funnel-down information from the first large layer of hidden units.

A network with feed-back structure can capture some aspects of problems not available to feed-forward networks. For multiple propositions, there is in general a probability distribution $Q^{\alpha}(\mu_1, \mu_2, \mu_3, \ldots)$ for variables $\mu_1, \mu_2, \mu_3, \ldots$, whereas this paper has dealt a matching criterion in which only the single-variable probability distributions are of interest. A symmetric statistical network can in principle generate higher-order correlations and represent more complex probability information. For such cases, the Boltzmann learning algorithm and the feed-forward learning systems will not be equivalent.

Appendix

A simple "diagnosis" model illustrates the significance of probability knowledge and experts when relatively few cases are available. Suppose a diagnosis "appendicitis" or "no appendicitis" is to be made on the basis of quantitative measurements of eight variables. In a noise- and variation-free world, these variables would have the values shown below.

Variable	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
Appendicitis	1	1	1	1	0	0	0	0
No appendicitis	−1	−1	−1	−1	0	0	0	0

However, the person-to-person variation of each of these parameters could be considerable and the values are also influenced by other illnesses, so that noise will be present in all data. The data available for a particular (appendicitis) case are then $1 + N_1, 1 + N_2, 1 + N_3, 1 + N_4, N_5, N_6, N_7$, and N_8. The noise values $N_1, \ldots, N_8$ were chosen uniformly between $N_{\max}$ and $-N_{\max}$. This diagnosis problem is appropriate for a two-layer analog perceptron with eight input units, a single output unit, eight adjustable weights, and one threshold value. [An analysis of an actual medical diagnosis problem using an analog perceptron has been made by Le Cun (personal communication).]

Suppose the existence of a perfect expert who can evaluate, for any set I_i^{α} the probability $Q_{\ddagger}^{\alpha}$ that the patient has appendicitis. There is also an oracle, who knows whether each patient α actually has appendicitis. Three learning protocols were compared: (*a*) The generalized Δ learning rule, with the target of ±0.8, using exact information from the oracle; (*b*) same as *a*, but with a yes/no diagnosis from the expert ("yes" if $Q_{\ddagger}^{\alpha} \geq 0.5$); (*c*) Eq. **6**, with probability advice Q from the expert for each case.

The network learned on a set of 20 cases, half of which were (according to the oracle) cases of appendicitis. The performance of each network was then evaluated on 200 new cases, using the diagnosis "appendicitis" if the output of the proposition unit was > 0 for case α. The experiment was tried many times to collect statistics. When the noise with $N_{\max}$ was such that the expert made 8.4% errors, protocols *a–c* yielded 12.6%, 10.5%, and 8.4% errors, respectively. The network trained with the oracle disagreed with the expert 9.4% of the time and that trained in protocol *b* differed 6.4%, whereas that trained on the basis of probability information disagreed with the expert only 1.4% of the time. The disagreement between the network and the perfect expert is a more striking measure of how well the problem has been solved because it is first-order in the difference between a particular network and the ideal one, while the number of excess errors increases only quadratically with such differences.

In this problem, all three learning protocols should achieve the same performance with large training sets. However, it is common to work with problems for which the data are not complete or exhaustive, and appropriate generalization by the network is still desired. In such a case, probability information from experts or other sources can be used to good advantage in learning protocol *c*. Even in a yes/no evaluation, a true expert is to be preferred to an oracle.

The arbitrary parameter S in Eqs. **8** and **9** can be qualitatively interpreted in this problem. In case *a* the pattern of the learning is already apparent early in the gradient descent, where all the values of $g'(u)$ are nearly equal. In this region, the gradient descent of Eq. **9** is equivalent to that of the special case of Eq. **6** in which each instance is assigned the same probability, 0.1, of being a misdiagnosis. This assignment is not optimal if cases in the training set are recognizably of varying uncertainty or if the probability of error is not 0.1. This anlaysis is not valid later in the learning, since $g'(u)$ will then take on different values. This may affect both the rate of learning and, in multilayer perceptrons, the quality of the minimum that is found.

I acknowledge helpful conversations with D. W. Tank, E. Baum, and S. Solla. This work was supported by contract N00014-K-0377 from the Office of Naval Research.

1. Werbos, P. (1974) Dissertation (Harvard University, Cambridge, MA)
2. Parker, D. B. (1985) *Learning-Logic* (Massachusetts Institute of Technology, Cambridge, MA), MIT Tech. Rep. TR-47.
3. Rummelhart, D. E., Hinton, G. E. & Williams, R. J. (1986) in *Parallel Distributed Processing*, eds. McClelland, J. L. & Rumelhart, D. E. (MIT Press, Cambridge, MA), Vol. 1, pp. 318–364.
4. Smolensky, P. (1986) in *Parallel Distributed Processing*, eds. McClelland, J. L. & Rumelhart, D. E. (MIT Press, Cambridge, MA), Vol. 2, pp. 194–281.
5. Rosenblatt, F. (1962) *Principles of Neurodynamics* (Spartan, Hasbrook Heights, NJ).
6. Gamba, A. L., Gamberini, G., Palmieri, G. & Sanna, R. (1961) *Nuovo Cimento* Supp. 2 **20**, 221–231.
7. Minsky, M. & Papert, S. (1964) *Perceptrons* (MIT Press, Cambridge, MA), pp. 205–228ff.
8. Widrow, G. & Hoff, M. E. (1960) *Western Electronic Show and Convention Record* (Institute of Radio Engineers, New York), Part 4, 96–104.
9. Hinton, G. E. and Sejnowski, T. J. (1983) *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Washington, DC), pp. 448–453.
10. Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. (1985) *Cognit. Sci.* **9**, 147–169.
11. Hopfield, J. J. (1982) *Proc. Natl. Acad. Sci. USA* **79**, 2554–2558.
12. Levinson, S. E., Rabiner, L. R. & Sondhi, M. M. (1983) *Bell Syst. Tech. J.* **62**, 1035–1074.
13. Bahl, L. R., Jelinek, R. & Mercer, R. L. (1983) *IEEE Trans. Patt. Anal. Mach. Int.* **5**, 179–190.

14. Anderson, C. H. & Abrahams, E. (1987) *Proceedings of the IEEE International Conference on Neural Networks* (IEEE, San Diego), in press.
15. Pinsker, M. S. (1964) *Information and Information Stability of Random Process* (Holden-Day, San Francisco), p. 19.
16. Sejnowski, T. J., Kienher, P. K. & Hinton, G. E. (1986) *Physica* **22D**, 260–275.
17. Gorman, P. & Sejnowski, T. J. (1987) *Workshop on Neural Network Devices and Applications* (Jet Propulsion Laboratory, Palo Alto, CA), Document D-4406, pp. 224–237.