American Institute of Physics, Conference Proceedings 151: Neural Networks for Computing, Snowbird, Utah, 1986. Johns S. Denker, ed. AIP, 1986.

HIGHER-ORDER BOLTZMANN MACHINES

Terrence J. Sejnowski
Johns Hopkins University, Baltimore, MD 21218

ABSTRACT

The Boltzmann machine is a nonlinear network of stochastic binary processing units that interact pairwise through symmetric connection strengths. In a third-order Boltzmann machine, triples of units interact through symmetric conjunctive interactions. The Boltzmann learning algorithm is generalized to higher-order interactions. The rate of learning for internal representations in a higher-order Boltzmann machine should be much faster than for a second-order Boltzmann machine based on pairwise interactions.

INTRODUCTION

Thousands of hours of practice are required by humans to become experts in domains such as chess, mathematics and physics[1]. Learning in these domains requires the mastery of a large number of highly interrelated ideas, and a deep understanding requires generalization as well as memorization. There are two traditions in the literature on learning in neural network models. One class of models is based on the problem of content-addressable memory and emphasizes a fast, one-shot form of learning. The second class of models uses slow, incremental learning, which requires many repetitions of examples. It is difficult in humans to study fast and slow learning in isolation. In some amnesics, however, the long-term retention of facts is severely impaired, but the slow acquisition of skills, including cognitive skills, is spared[2]. Thus, it is possible that separate memory mechanisms are used to implement fast learning and slow learning.

Long practice is required to become an expert, but expert performance is swift and difficult to analyze; with more practice there is faster performance[1]. Why is slow learning so slow? One possibility is that the expert develops internal representations that allow fast parallel searches for solutions to problems in the task domain, in contrast to a novice who must apply knowledge piecemeal. An internal representation is a mental model of the task domain; that is, internal degrees of freedom between the sensory inputs and motor outputs that efficiently encode the variables relevant to the solution of the problem. This approach can be made more precise by specifying neural network models and showing how they incorporate internal representations.

LEARNING IN NETWORK MODELS

Network models of fast learning include linear correlation-matrix models[3,4,5,6] and the more recent nonlinear autoassociative models[7,8,9,10]. These models use the Hebb learning rule to store information that can be retrieved by the completion of partially specified input patterns. New

patterns are stored by imposing the pattern on the network and altering the connection strengths between the pairs of units that are above threshold. The information that is stored therefore concerns the correlations, or second-order relationships between the components of the pattern. The internal model is built from correlations.

Network models of slow learning include the perceptron[11] and adaline[12]. These networks can classify input patterns given only examples of inputs and desired outputs. The connection strengths are changed incrementally during the training and the network gradually converges to a set of weights that solves the problem if such as set of weights exists. Unfortunately, there are many difficult problems that cannot be solved with these networks, such as the prediction of parity[13]. The perceptron and adaline are limited because they have only one layer of modifiable connection strengths and can only implement linear discriminant functions. Higher-order problems like parity cannot be solved by storing the desired patterns using the class of content-addressable algorithms based on the Hebb learning rule. These models are limited because the metric of similarity is based on Hamming distance and only correlations can be used to access patterns.

The first network model to demonstrably learn to solve higher-order problems was the Boltzmann machine, which overcame the limitations of previous network models by introducing hidden units[14,15,16]. Hidden units are added to the network to mediate between the input and output units; they provide the extra internal degrees of freedom needed to form internal representations. The Boltzmann learning algorithm incrementally modifies internal connections in the network to build higher-order pattern detectors. The hidden units can be recruited to form internal representations for any problem; however, the learning may require an extremely large number of training examples and can be excessively slow. One way to speed up the learning is to use hidden units that have higher-order interactions with other units.

THIRD-ORDER BOLTZMANN MACHINES

Consider a Boltzmann machine with a cubic global energy function:

$$E = -\frac{1}{3}\sum_i\sum_j\sum_k w_{ijk}\, s_i\, s_j\, s_k$$

where s_i is the state of the i th binary unit and w_{ijk} is a weight between triples of units. This type of interaction generalizes the pairwise interactions in Hopfield networks[10] and Boltzmann machines, which contribute a quadratic term to the energy. Fig. 1 shows an interpretation of the cubic term as conjunctive synapses. Each unit in the network updates its binary state asynchronously with probability

$$p_i = \frac{1}{1 + e^{-\Delta E_i / T}}$$

where T is a parameter analagous to the temperature and the total input to the i th unit is given by

$$\Delta E_i = \sum_j \sum_k w_{ijk} s_j s_k$$

If w_{ijk} is symmetric on all pairs of indices

$$w_{ijk} = w_{jik} = w_{ikj} = w_{kji}$$

then the energy of the network is nonincreasing. It can be shown that in equilibrium the probabilities of global states P_α follow a Boltzmann distribution

$$\frac{P_\alpha}{P_\beta} = e^{-\frac{E_\alpha}{E_\beta}}$$

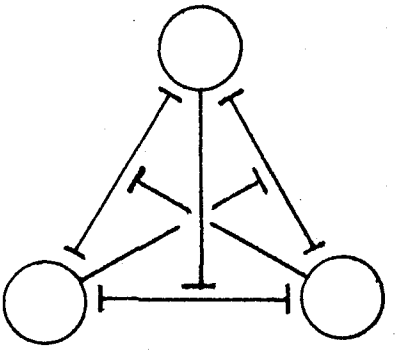

Fig. 1. Third-order interactions between three units. In the diagram the lines between units represent reciprocal interactions that are activated only when the third unit is in the *on* state. The third unit acts presynaptically to conjunctively control the pairwise interactions.

There are two forms of the Boltzmann learning algorithm, one for networks with inputs and outputs treated identically, and a second for networks where the input units are always clamped[15]. The former learning algorithm will be generalized for third-order interactions. The learning metric on weight space remains the same:

$$G = \sum_{\alpha} P_{\alpha} \log \frac{P_{\alpha}}{P'_{\alpha}}$$

where P_{α} is the probability of a global state with both the inputs and outputs clamped, and P'_{α} is the probability of a global state when the network is allowed to run freely. It can be shown that the gradient of G is given by

$$\frac{\partial G}{\partial w_{ijk}} = -\frac{1}{T}(p_{ijk} - p'_{ijk})$$

where p_{ijk} is the ensemble average probability of three units all being in the *on* state when the input and output units are clamped, and p'_{ijk} is the corresponding probability when the network is running freely. To minimize G, it is sufficient to measure the time averaged triple co-occurence probabilities when the network is in equilibrium under the two conditions and to change each weight according to

$$\Delta w_{ijk} = \epsilon (p_{ijk} - p'_{ijk})$$

where ϵ scales the size of each weight change.

HIGHER-ORDER BOLTZMANN MACHINES

Define the energy of a k-th order Boltzmann machine as

$$E = -\frac{1}{k} \sum_{\gamma_1} \sum_{\gamma_2} \cdots \sum_{\gamma_k} w_{\gamma_1 \gamma_2 \cdots \gamma_k} s_{\gamma_1} s_{\gamma_2} \cdots s_{\gamma_k}$$

where $w_{\gamma_1 \gamma_2 \cdots \gamma_k}$ is a k-dimensional weight matrix symmetric on all pairs of indices. The G matrix can be minimized by gradient descent:

$$\Delta w_{\gamma_1 \gamma_2 \cdots \gamma_k} = \epsilon (p_{\gamma_1 \gamma_2 \cdots \gamma_k} - p'_{\gamma_1 \gamma_2 \cdots \gamma_k})$$

where $p_{\gamma_1 \gamma_2 \cdots \gamma_k}$ is the probability of the k-tuple co-occurence of the $(s_{\gamma_1}, s_{\gamma_2}, \cdots s_{\gamma_k})$ when the inputs and outputs are clamped, and $p'_{\gamma_1 \gamma_2 \cdots \gamma_k}$ is the corresponding probability when the network is freely running.

In general, the energy for a Boltzmann machine is the sum over all orders of interaction and the learning algorithm is a linear combination of terms from each order. This is a Markov random field with polynomial interactions[17].

DISCUSSION

Conjunctive synapses such as those studied here can be used to model multiplicative relationships[18]. In a third-order Boltzmann machine the conjunctive interactions must be symmetric between all three pairs of units in a triple. This configuration has been used to implement shape recognition using mappings from a retinal-based frame of reference to object-based frames of reference[19,20]. In principle, these mappings could be learned by a sufficiently large number of hidden units with only pairwise interactions, but in practice the number of units and time required would be prohibitive. Learning this mappings using third order interactions occurs much more quickly.

Higher-order interactions have recently been introduced into content-addressable networks with fast learning[21,22]. The storage capacity of these networks is much larger than networks with only pairwise connections, but the number of connections is also much larger. Another advantage of higher-order interactions is the possibility of storing higher-order predicates[13]. However, these networks remain limited in their ability to generalize because they can only memorize the stored patterns; without hidden units they cannot generate new internal representations.

One of the serious problems with all higher-order schemes is the proliferation of connections, which tend to be the most expensive part of an implementation. A network of n units would require $O(n^k)$ connections to implement all interactions of order k. For example, consider the problem of learning mirror symmetries[16]. Random-dot patterns are generated with a mirror symmetry along one of several axes in an $N \times N$ grid. The task is to learn to classify new patterns given only examples of correctly classified mirror-symmetric patterns. A Boltzmann machine with pairwise interactions and 12 hidden units between the input and output layer can learn to classify patterns in about 50,000 trials. Using third-order interactions between the input and output layer would require $O(N^4)$ connections, most of which would be superfluous since only $O(N^2)$ of these connections carry any information relevant to the solution of the problem. Thus, learning may be faster but the price in connections may be prohibitive.

Whether a higher-order Boltzmann machine is of practical value depends on the tradeoff between the increased number of connections and the decreased learning time. At present it is not known how learning in Boltzmann machines scales with the size and difficulty of a problem, but it should be possible to simulate higher-order Boltzmann machines for small problems and compare them with conventional second-order Boltzmann machines. Other incremental learning algorithms, such as backpropagation[23] can also be generalized to higher-order units.

REFERENCES

1. D. A. Norman, Learning and Memory (W. H. Freeman, New York, 1982).

2. L. R. Squire & N. J. Cohen, In: J. L. McGaugh, N. W. Weinberger & G. Lynch (Eds.) Neurobiology of Learning and Memory (Guilford Press: New York, 1984)
3. K. Steinbuch, Kybernetik 1, 36-45 (1961)
4. J. A. Anderson, Math. Biosci. 14, 197-220 (1972)
5. T. Kohonen, IEEE Trans. C-21, 353 (1972)
6. H. C. Longuet-Higgins, Nature 217, 104 (1968)
7. J. A. Anderson & M. C. Mozer, In: G. E. Hinton & J. A. Anderson (Eds.), Parallel Models of Associative Memory (Erlbaum Associates, Hillsdale, N. J., 1981).
8. T. Kohonen, Self-Organization and Associative Memory (Springer-Verlag, New York, 1984).
9. T. J. Sejnowski, In: G. E. Hinton & J. A. Anderson (Eds.), Parallel Models of Associative Memory (Erlbaum Associates, Hillsdale, N. J., 1981).
10. J. J. Hopfield, Proceedings of the National Academy of Sciences USA 79, 2554-2558 (1982).
11. Rosenblatt, F., Principles of Neurodynamics, (Spartan, New York, 1959).
12. B. Widrow, In: M. C. Yovits, G. T. Jacobi & G. D. Goldstein (Eds.), Self-Organizing Systems 1962 (Spartan Books, Washington, D. C., 1962)
13. M. Minsky, & S. Papert, Perceptrons, (MIT Press, Cambridge, 1969).
14. G. E. Hinton, & T. J. Sejnowski, Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition, Washington, D. C., 448-453. (1983).
15. D. H. Ackley, G. E. Hinton, & T. J. Sejnowski, Cognitive Science 9, 147-169 (1985).
16. T. J. Sejnowski, P. K. Kienker & G. E. Hinton, Physica D (in press).
17. S. Geman, & D. Geman, IEEE Transactions on Pattern Analysis and Machine Intelligence 6, 721-741 (1984).
18. J. A. Feldman & D. H. Ballard, Cognitive Sci. 9, 205-254 (1983)
19. G. E. Hinton, Proc. 7th Int. Joint Conf. Artif. Intel. 1088-1096 (Kauffman, Los Altos, CA, 1981).
20. G. E. Hinton & K. J. Lang, Proc. 9th International Joint Conf. Artif. Intel. 252-259 (Kauffman, Los Altos, CA, 1985).
21. D. Psaltis, Proc. Snowbird Meeting on Neural Networks for Computing (1986)
22. Y. C. Lee, G. Doolen, H. H. Chen, G. Z. Sun, T. Maxwell, H. Y. Lee & L. Giles, Physica D (in press).
23. D. A. Rumelhart & G. E. Hinton, In: D. Rumelhart & J. McClelland (Eds.), Parallel Distributed Processing (MIT Press, Cambridge, 1986).