A statistical mechanical study of Boltzmann machines

1987 J. Phys. A: Math. Gen. 20 2133

(http://iopscience.iop.org/0305-4470/20/8/027)

J. Phys. A: Math. Gen. **20** (1987) 2133-2145. Printed in the UK

A statistical mechanical study of Boltzmann machines

D G Bounds

Royal Signals and Radar Establishment, St Andrews Road, Malvern, Worcs WR14 3PS, UK

Received 1 August 1986

Abstract. Hinton and Sejnowski have described recently a novel statistical mechanical system which they named the Boltzmann machine. The interesting property of Boltzmann machines is that they can learn to recognise the structure in a set of patterns simply by being shown an example subset of patterns. In this paper some numerical simulations of Boltzmann machines are reported. It is found that the annealing schedule proposed by Ackley, Hinton and Sejnowski is adequate to obtain a Boltzmann distribution of states, on which the key part of the algorithm depends, but it is clear that the algorithm will require massive computations for large networks. It is also found that there is a window of annealing temperatures at which learning is possible, and the sensitivity of the learning rate to temperature can be understood in terms of the density of states at low energies. Direct calculations of the partition function in small instances of Boltzmann machines are used to characterise the number of states which are thermally accessible for particular annealing schedules. Finally, since Boltzmann machines bear some resemblance to models of disordered magnetic systems, a comparison is made with results for the Sherrington-Kirkpatrick spin-glass model. Both systems support multiple metastable states (i.e. stable with respect to single spin flips), but, in contrast to the SK spin glass, Boltzmann machines exhibit a random distribution of low-energy states in terms of Hamming distance.

1. Introduction

There is currently considerable interest in the computational abilities of networks of simple processing units. Such networks loosely resemble neural networks in that they have high connectivity, a highly non-linear response at the 'neuron' and each neuron's output is determined by whether or not a weighted sum of the inputs from other neurons exceeds some threshold. Networks have been proposed which show fascinating emergent collective properties including the ability to memorise patterns and to recall perfectly stored patterns, given noisy or incomplete cues. Apart from the possible relevance of this behaviour to information processing in living systems, these models are promising candidates for pattern recognition devices and for fault-tolerant, content-addressable memories (CAM) with some capacity for error correction. A key difference between neural network models and conventional computer memories is that information is not stored locally at an address but in a distributed fashion throughout the whole network as a stable state of the dynamical system. Statistical mechanics is therefore a natural tool for the study of these systems, and the more simple models have already shown a rich variety of physical behaviour (Amit *et al* 1985a, b, Bruce *et al* 1986, Gardner 1986, Wallace 1985), with some similarities to that found in disordered magnetic systems.

A network consists of a set of N units, $S=\{\sigma_i;\ i=1,\ldots,N\}$, where in some models the units are two-state units ($\sigma_i=0$ or 1, or $\sigma_i=\pm 1$) and in others they can

0305-4470/87/082133+13$02.50

take continuous values in the range (0, 1). The units are joined by a set of scalar links $W = \{w_{ij};\ i, j = 1, \ldots, N\}$. For symmetric links, $w_{ji} = w_{ij}$, this is sufficient to define a Hamiltonian:

$$E(W, S) = -\sum_{i=1}^{N} \sum_{j=i}^{N} w_{ij}\sigma_i\sigma_j. \tag{1}$$

A pattern to be stored defines the state of a set, V, of 'visible' units, $V \subseteq S$, and a learning algorithm subsequently adjusts the connections W so that stored patterns correspond to low-energy states of the network.

In the simplest models, such as the one now associated with Hopfield (1982), a pattern fixes the states of all units, i.e. $V = S$. There is then no internal coding of patterns (beyond the fact that they correspond to energy minima) and input patterns which are close in Hamming distance produce the same output pattern. Although such networks exhibit the storage behaviour described above, they are limited by their inability to map dissimilar input patterns onto the same output where necessary. This may be achieved by networks containing a set of 'hidden' units, $H = S - V$, whose states are not fixed directly by the input patterns. A network with hidden units is shown in figure 1. Because the hidden units must generate some internal coding of the input patterns (since the input and output units in figure 1 are not connected to each other directly), they may act as feature detectors. Networks containing hidden units therefore show additional interesting behaviour beyond that possible in networks without hidden units. One simple example where hidden units are necessary is computation of the exclusive OR function (XOR) which has the truth table shown in table 1. In this problem patterns with the least overlap are required to give the same output. The XOR problem can be solved with appropriate connection strengths to a hidden unit whose state depends on whether or not the states of both input units are the same (Rumelhart *et al* 1986). In general, hidden units are necessary when the state of an output unit depends on the states of two or more input units: hidden units capture non-pairwise additive correlations in the input patterns.

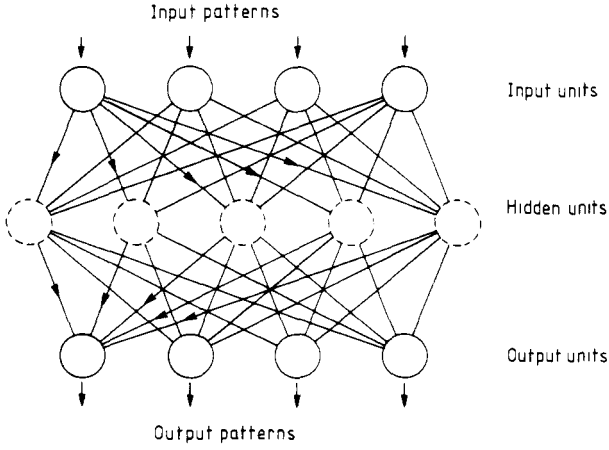

Figure 1. A network with hidden units. In this example the visible set V consists of the input and output units. Because the input and output units can only communicate via the hidden units, the network must learn connection strengths which result in an internal coding of the input patterns using the states of hidden units.

Table 1. Truth table for the exclusive OR function. In a network model which computes this function, each element of the input and output patterns would be represented by the state of a visible unit.

Input pattern	Output pattern
F F	F
F T	T
T F	T
T T	F

There have been several numerical and theoretical studies of the Hopfield model (Amit *et al* 1985a, b, Wallace 1985, Bruce *et al* 1986, Gardner 1986) and many of its properties have been elucidated. However, apart from the work of the original authors, we are not aware of any studies of the archetypal hidden unit model: the Boltzmann machine (Hinton *et al* 1983, Ackley *et al* 1985). In view of the novel behaviour which this system has demonstrated, at least for small model problems, a statistical mechanical study is timely. Here we investigate how the behaviour of Boltzmann machines depends upon various parameters in order to see how the algorithm might scale to large systems. Direct calculations of the partition function and the density of states give insights into the number of states which must be thermally accessible if the machine is to learn successfully, and into the sensitivity of the learning rate to temperature. The relationship between the Boltzmann machines and spin glasses is also discussed.

The general Boltzmann machine algorithm is described in § 2, and in § 3 a model problem, the v-h-v encoder (Ackley *et al* 1985), is outlined. Section 4 describes the calculations. The main body of the results is in § 5, and the conclusions are summarised in § 6.

2. The Boltzmann machine algorithm

Boltzmann machines are so called because they are stochastic systems in which the relative probability of two network states α and β depends on their energy difference through the Boltzmann relation:

$$P_\alpha / P_\beta = \exp[-(E_\alpha - E_\beta)/T] \tag{2}$$

where the temperature T is in inverse energy units. For a comprehensive description of Boltzmann machines, readers should consult the elegant paper of Ackley *et al* (1985). Here is a brief outline of the salient features.

In the original formulation $\sigma_i = 0$ or 1 and the (symmetric) links take even integer values. Dynamics are provided by a variant of the Metropolis algorithm. The visible units of a Boltzmann machine are split into an input set and an output set. At each stage of the algorithm a Boltzmann machine is running in one of three modes, depending upon which subset of V is held fixed. In the training mode both inputs and outputs are held fixed and the hidden units are allowed to change state. In the free-running mode neither inputs nor outputs are fixed—all units are allowed to flip. In the 'testing for completion' mode some inputs are fixed and the machine, if it has learned successfully, produces appropriate outputs. The free-running mode is necessary because data collected in this mode are required by the feedback mechanism which alters the link values.

There are several ways to arrive at the Boltzmann machine algorithm. Since the energy function can take only a limited number of values, it is useful to consider how many patterns could be stored in a network of given size. Suppose there are v visible units and h hidden units. For two-state units the possible number of patterns among the visible units is 2^v. However, the energy expression (1) only has $N(N+1)/2 = (v+h)(v+h+1)/2$ possible values for a given set of link values W. Furthermore, only a small number of these will be low-lying minima which would ensure the stability of stored patterns. Hence, for any set of link values, if patterns are to be associated with energy minima then it is only possible to distinguish between all possible 2^v states if $(v+h)(v+h+1) \gg 2^v$, which requires exponentially large h. While this would be feasible for small v, it would be impossible for any useful application. Because a perfect model is not possible, except for small v, the problem is to obtain an optimum model (i.e. an optimum set of w_{ij}) given some non-exponential number of hidden units. Hinton and Sejnowski (1983, see also Ackley *et al* 1985) use the information gain (Kullback 1959), G, to provide an optimality condition for W. G is defined as

$$G = \sum_{\alpha} P_\alpha \ln[P_\alpha / P^{\mathrm{f}}_\alpha] \tag{3}$$

where P_α is the probability of the network being in state α of the *visible* units when the machine is in the training mode and P^f_α is the corresponding probability when the machine is running freely without any pattern clamped on. G is zero if and only if the probability distributions P and P^{f} are equal, in which case the machine is modelling the set of input patterns perfectly. Otherwise $G > 0$ and the best model is that which minimises G. Since a direct evaluation of G is clearly not possible except for miniscule problems, an expression for the partial derivatives $\partial G/\partial w_{ij}$ is required. For highly interconnected non-linear networks this is not usually available (Ackley *et al* 1985), but a simple relation does exist for a system where the Boltzmann relation (equation (2)) holds. The simplification arises because for a Boltzmann distribution the log probabilities of states is a linear function of their energies. A derivation is given by Ackley *et al* (1985). The final expression is

$$\partial G/\partial w_{ij} = -(1/T)(p_{ij} - p^{\mathrm{f}}_{ij}) \tag{4}$$

where

$$p_{ij} \stackrel{\mathrm{def}}{=} \sum_{\alpha} \sum_{\beta} P_{\alpha\beta} \sigma_i^{\alpha\beta} \sigma_j^{\alpha\beta} \tag{5}$$

and

$$p^{\mathrm{f}}_{ij} \stackrel{\mathrm{def}}{=} \sum_{\lambda} \sum_{\mu} P_{\lambda\mu} \sigma_i^{\lambda\mu} \sigma_j^{\lambda\mu}. \tag{6}$$

The outer summations in (5) and (6) are over visible unit states and the inner sums are over hidden unit states. $\sigma_i^{\alpha\beta}$ is the state of unit i when the network is in a global state defined by the visible set being in state V_α and the hidden set being in state H_β. The derivatives necessary to minimise G are therefore available, provided that $\{p_{ij}\}$ and $\{p^{\mathrm{f}}_{ij}\}$ are calculated when the machine has reached thermal equilibrium. Thermal equilibrium is attained starting from any initial state by carrying out a series of Metropolis-type calculations at decreasing temperatures, i.e. by the simulated annealing technique described by Kirkpatrick *et al* (1983). Since $\sigma_i = 0$ or 1, p_{ij} is just the average probability that both units are in the 1 state when a pattern is clamping the visible

units, and p_{ij}^{f} is the same quantity when the machine is in the free-running mode. In their simulations, Ackley *et al* (1985) incremented or decremented the w_{ij} by a fixed amount depending on the sign of $(p_{ij} - p_{ij}^{f})$, instead of adopting a steepest descent method as suggested by equation (4). The same procedure has been used in the present calculations.

3. The v–h–v encoder problem

The encoder problem was proposed as a simple abstraction of the task of communicating information between components of a parallel network (Ackley *et al* 1985). It may also be viewed as a pattern recognition problem. The visible units are split into two groups, V_1 and V_2, each with $v = v/2$ units. All units in V_1 are connected to each other, as are all units in V_2, but there are no direct connections between V_1 and V_2. Instead, the visible groups communicate via a set of h hidden units. The hidden units are not connected to each other, but each is connected to all the visible units. The network is therefore the same as figure 1 except for the interconnections within V_1 and V_2. The problem is to evolve a set of w_{ij} which allows the visible groups to communicate their states to each other. Most of the calculations reported here are for a 4-2-4 encoder network.

In the 4-2-4 encoder problem, there are 2^4 possible states for each visible group. However, in the version studied by Ackley *et al* (1985), though all possible states have some probability of occurring in the training set, the statistics are dominated by nominated patterns where only one unit in V_1 and the corresponding unit in V_2 are in the $\sigma = 1$ state. The training patterns which occur most frequently are the four vector pairs: $V_1 = V_2 = (1, 0, 0, 0)$ or $(0, 1, 0, 0)$ or $(0, 0, 1, 0)$ or $(0, 0, 0, 1)$. Less frequently, the patterns are 'noisy' versions such as $V_1 = (1, 1, 0, 0)$, $V_2 = (0, 0, 1, 0)$. Because there are only two hidden units, the system can only communicate 2^2 states. The machine must therefore recognise that the four vector pairs above occur most often, and then develop a set of weights which encode these patterns so that they correspond to the four possible hidden unit patterns $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$. Since both the number of the noisy vectors and which particular noisy vectors occur varies from learning cycle to learning cycle during the iterative training process, this is not a trivial problem.

4. Simulation methods

Full details of the original experiments are given in Ackley *et al* (1985). However, it is worth summarising one iteration or 'learning cycle'.

(i) One pattern fixes the states of the visible units.

(ii) The hidden units are flipped according to a variant of the Metropolis Monte Carlo algorithm at a temperature T_{max}.

(iii) The system is annealed to T_{min} by repeating (ii) at successively lower temperatures.

(iv) Monte Carlo flips of the hidden units are continued at T_{min} and statistics are gathered about how often pairs of units are on simultaneously. (This accumulates the inner sum in equation (5).)

(v) Steps (i)-(iv) are repeated for a set of training patterns. (This accumulates the outer sum in equation (5).)

(vi) All units are now free to flip and the system is annealed to T_{min} as in steps (ii) and (iii).

(vii) Monte Carlo flips of all units are continued at T_{min} and statistics are gathered for $\{p_{ij}^f\}$ (equation (6)) for the same length of time as for $\{p_{ij}\}$.

(viii) The connections $\{w_{ij}\}$ are updated on the basis of $\text{sgn}(p_{ij}-p_{ij}^f)$.

It is evident that this is a computationally expensive process, and it is not surprising that convergence depends on various parameters which enter the Boltzmann machine algorithm. These include how often noisy vectors occur in the training sequence, the temperatures used and the length of time allowed for equilibration in the annealing schedule, the length of time over which statistics for $\{p_{ij}\}$ and $\{p_{ij}^f\}$ are collected (which affects the size of uphill steps in G space (Ackley *et al* 1985)) and the choice of weight update. Some of these factors have been discussed by Ackley *et al.* Here we focus on the effects of temperature and annealing time because they are the crucial parameters in the inner optimisation loop of the algorithm, and we have found that they play a major role in determining how the system evolves.

In 250 simulations of the 4-2-4 encoder, Ackley *et al* (1985) found that the median time to find a stable solution was 110 learning cycles and the longest time was 1810 cycles. Initially we used the same parameter values in order to try to reproduce the behaviour found there. In the present calculations we have considered a stable solution to have been found on cycle N if the same solution is found on cycles $N+1$, $N+2, \ldots, N+9$. In our experience it was rare for a machine to drift from a solution after this criterion was satisfied. In 150 simulations we found a median learning time of $N=149$ cycles. The shortest learning time was 23 cycles, and in 32 instances a stable solution was not found in the cut-off time of 400 cycles. In view of the broad distribution of convergence times, the present results are not significantly different from those of Ackley *et al.* Starting from a network with all $w_{ij}=0$, Ackley *et al* identified three stages of learning in terms of the way in which the weights changed with time. We observed identical behaviour. The sizes of the weights obtained when learning is complete are also similar to those found by Ackley *et al.* We have thus verified that our program behaves in a similar way to that of the original work.

5. Results

5.1. Equilibration and the annealing time

In the original work (Ackley *et al* 1985), when a settling to equilibrium was required all the unclamped units were randomised to 0 or 1 with equal probability, and then the network was allowed to run for two units of time at a temperature of $T=20$, followed by two at $T=15$, two at $T=12$ and four at $T=10$. It is not obvious why these temperatures were chosen or whether learning in Boltzmann machines is sensitive to temperature. Furthermore, it was assumed that this schedule allows the system to reach equilibrium. The total annealing schedule, ten units of time, allows each unclamped unit to have ten opportunities to flip states on average. However, in Monte Carlo simulations of solids and liquids, from which the Metropolis method is taken, calculations are usually carried out on hundreds of units (= atoms in that case) and it is necessary to allow at least 1000 trial flips per unit before the system is likely to have reached equilibrium. It is therefore not obvious that Boltzmann machines reach equilibrium under the chosen schedule. This is important, since unless $\{p_{ij}\}$ and $\{p_{ij}^f\}$

are collected when the network is at equilibrium, the state probabilities do not obey the Boltzmann distribution and equation (4), which is the whole basis of the method, no longer holds.

One way to test whether equilibrium is reached is to extend the number of time steps over which statistics for $\langle E \rangle$ are collected at the end of annealing and calculate $\langle E \rangle$ as a function of time, since equilibrium implies $d\langle E \rangle / dt = 0$. However, since in the 4-2-4 encoder there are only 2^{10} possible states for a given set of link strengths, the exact value of $\langle E \rangle$ can be calculated without approximation:

$$\langle E \rangle = (1/Z) \sum_{i=1}^{1024} E_i \exp(-E_i/T) \tag{7}$$

where Z is the partition function:

$$Z = \sum_{i=1}^{1024} \exp(-E_i/T). \tag{8}$$

$\mathrm{Var}^2(E) = \langle E \rangle^2 - \langle E \rangle^2$ can also be obtained from equation (7) and

$$\langle E^2 \rangle = (1/Z) \sum_{i=1}^{1024} E_i^2 \exp(-E_i/T). \tag{9}$$

Table 2 shows a typical comparison of $\langle E \rangle$ obtained by averaging after simulated annealing using the schedule above, together with the exact sum-over-states results from equations (7)-(9). The results are for the free-running mode where all ten units are free to flip, and the average was taken over the length of time used to collect $\{p_{ij}^{f}\}$ in the simulations of Ackley *et al* (ten units of time) at the lowest annealing temperature, $T_{\min}$. The weights used were a typical set obtained from a converged learning run.

Table 2. Energy results for a typical Boltzmann machine.

	BM algorithm	Exact SOS
$\langle E \rangle$	22.0	18.9
var(E)	14.0	14.5

The annealing schedule is long enough for the system to reach equilibrium, but only because the fluctuations are so large in this small system. Useful applications would probably require Boltzmann machines with hundreds or thousands of units. For large systems the ratio $\mathrm{var}(E)/\langle E \rangle$ will be smaller, and it will then be necessary to allow much longer times for equilibration, as is the case in Monte Carlo calculations on solids and liquids. There is already empirical evidence that larger networks require longer annealing schedules. In experiments on a 40-10-40 encoder, Ackley *et al* found that 'To achieve good performance on the completion tests (i.e. convergence of the learning algorithm), it was necessary to use a very gentle annealing schedule during testing. The schedule spent twice as long at each temperature and went down to half the final temperature of the schedule used during learning'. This is actually a larger increase in real time than a factor of two, since the unit of time in the algorithm is proportional to the number of free units.

5.2. The effect of temperature on the learning rate

Temperature influences the learning rate by altering the number of energy states which are thermally accessible at any stage of the learning algorithm. The partition function is one measure of the number of states readily accessible at temperature T. At low T, the limiting value of Z is the ground-state degeneracy of the system (if energies are defined relative to the ground-state energy), while at high T, Z approaches the total number of states in the system.

The partition function may be used to investigate several aspects of Boltzmann machine behaviour. A Boltzmann machine with all $w_{ij} = 0$ has not learned to solve the encoder problem, and, because all states have the same energy, $Z = 1024$ for all temperatures. However, a solution to the 4-2-4 problem must require at least four low-energy states at the lowest temperature of the annealing schedule, T_{min}. Given that the number of states which must be thermally accessible at T_{min} when a Boltzmann machine has evolved connections which solve the encoder problem is somewhere between 4 and 1024, what number is typically observed?

Table 3 shows the mean values of the partition function at T_{min} and at the highest annealing temperature, T_{max}. In each case Z was calculated from the final set of weights, i.e. when the convergence criterion was satisfied or after the cut-off time of 400 iterations. Averages were obtained over converged and unconverged machines separately. For the original annealing schedule (Ackley *et al* 1985), shown in the column headed 'normal T', there are no significant differences between converged and unconverged machines. Differences in the connection strengths, which determine whether or not the network has formed a satisfactory model of its environment, are not reflected in the underlying energy surface. Approximately eight states are available at T_{min} and approximately 35 states make a significant contribution to Z at T_{max}.

Table 3. A summary of partition function data and the effect of annealing temperature on convergence. Z_{min} is the mean value of Z at T_{min} and Z_{max} is the mean value at T_{max}.

	Low T	Normal T	High T
Annealing schedule	2 at 4	2 at 20	2 at 100
(number of timesteps	4 at 3	2 at 15	2 at 75
at temperature T)	4 at 2	2 at 12	2 at 60
		4 at 10	4 at 50
Number of runs	50	150	20
Fraction of runs that converged in 400 cycles	0.56	0.79	0.10
Median learning time (cycles)	350	149	>400
Z_{min}			
converged	3.6 ± 1.30	7.75 ± 1.85	15.75 ± 1.39
unconverged	5.04 ± 1.99	7.61 ± 1.73	11.99 ± 1.79
all machines	4.24 ± 1.69	7.72 ± 1.82	12.37 ± 1.89
Z_{max}			
converged	12.54 ± 4.84	35.98 ± 6.00	61.75 ± 0.62
unconverged	15.05 ± 5.51	32.51 ± 4.79	48.73 ± 4.99
all machines	13.64 ± 5.16	35.24 ± 5.82	50.03 ± 5.43

At low temperatures one might expect learning to be difficult because not enough states are thermally accessible. Convergence should also be poor at high annealing temperatures because too many states are available and the algorithm will fail to settle into a good solution. Fifty low-temperature Boltzmann machines were run with the annealing schedule (2 at $T=4$, 4 at $T=3$, 4 at $T=2$) and 20 high-temperature runs were carried out with the schedule (2 at $T=100$, 2 at $T=75$, 2 at $T=60$, 4 at $T=50$). The results are summarised in table 3. In both cases convergence is much worse. There is therefore some window of temperature at which learning in Boltzmann machines is possible. In much larger systems it might be profitable to look for phase transitions separating these three regions, but quantitative results would be meaningless in the small system studied here.

More detailed information about the energy surfaces can be obtained from the density of states, $\rho(E)$. The partition function is a crude measure of $\rho(E)$ at low energies. The mean, $\bar{\rho}(E)$, over all machines run with the normal annealing schedule is plotted in figure 2 in terms of the reduced quantity $\varepsilon = E/T_{\min}$. The distribution is rather broad, but a large number of states become important at energies just above the highest annealing temperature ($\varepsilon = 2$). This fact, combined with the partition function data for high T runs where convergence is poor, suggests that the 'normal' annealing schedule is close to the high-temperature end of the learning window.

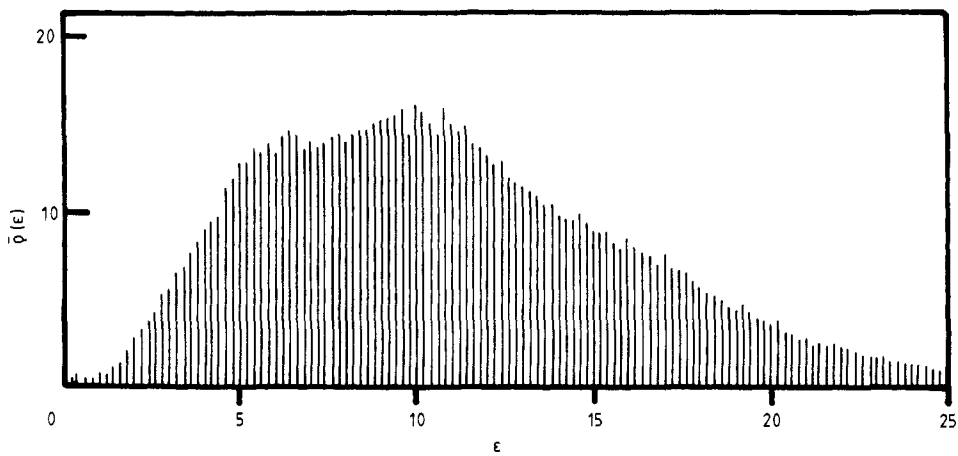

Figure 2. The mean density of states $\bar{\rho}(\varepsilon)$ over all 150 machines run with the normal T annealing schedule.

Ackley *et al* observed three stages of learning in the encoder problem. The first phase (starting from all $w_{ij}=0$) involves the development of negative weights throughout the network. In the second phase, the hidden units develop positive weights to some of the units in the visible groups, and the links between a hidden unit and equivalent units in V_1 and V_2 are roughly symmetric. By the end of the second stage most of the hidden unit codes are being used but some codes are utilised for more than one pattern. During the third stage, these final conflicts are resolved and a good solution is found. It is interesting to ask how the energy surface evolves during the learning process. As a lowest-order look at this, we have followed $Z_{\min}$ as a function of time (in learning cycles). Because of the small system size and the relatively large step size in the weight update, there are quite large fluctuations in $Z_{\min}$ from cycle to

cycle, especially when the weights are small at short times. However, some clear trends do emerge. $Z_{\min}$ drops rapidly over the first few cycles. For the normal T runs the latest cycle on which $Z_{\min} > 30$ is approximately 30 cycles. $Z_{\min}(t)$ then oscillates in the range 15–25 for, typically, another 30 cycles before dropping to final values consistent with the means in table 3. This behaviour is reminiscent of the description by Ackley *et al* of the three stages of learning and it is tempting to relate these stages to particular Z regimes. If, however, there are quantitative correlations, they are hidden by the large fluctuations.

5.3. Low-energy states and the relationship to spin glasses

The Boltzmann machine energy function, equation (1), is similar to the Hamiltonian of an Ising model in the absence of an external field:

$$H = -\sum_{ij} J_{ij}\sigma_i\sigma_j \qquad \sigma = \pm 1. \tag{10}$$

For a fully connected network, and if the J_{ij} are independent random variables with a Gaussian distribution, equation (10) is the Sherrington-Kirkpatrick (SK) spin-glass Hamiltonian (Sherrington and Kirkpatrick 1975, Kirkpatrick and Sherrington 1978). The SK spin glass is known to support such interesting phenomena as an ultrametric space amongst the overlaps between pure states (Mezard *et al* 1984). The overlap between pure states 1 and 2 is

$$q^{12} = (1/N)\sum_i \sigma_i^{(1)}\sigma_i^{(2)} \tag{11}$$

where $\sigma^{(1)}$ and $\sigma^{(2)}$ are two sets of spins with the same set of interactions.

However, there are also some important differences between the SK Hamiltonian and the Boltzmann machine Hamiltonian. In particular, the units of a Boltzmann machine need not be fully interconnected—they are not in the encoder problem. Furthermore, the Boltzmann machine links w_{ij} are not random variables; they must reflect correlations in the input patterns. To what extent do trained Boltzmann machines resemble spin glasses? Do they support an ultrametric topology? The latter question may have practical implications since ultrametric spaces are desirable for pattern classification (Jardine and Sibson 1971). We note in passing that spin-glass techniques have proved fruitful in understanding a simpler neural network model associated with Hopfield (1982). The existence and nature of phase transitions in the Hopfield model, which relate in that case to the pattern storage capacity of the network, have been elucidated by Amit *et al* (1985a, b), Wallace (1985), Bruce *et al* (1986) and Gardner (1986).

Whether or not the low-energy states of the Hopfield model have an ultrametric structure is an open question. No evidence for ultrametricity in the higher-energy minima, those which are usually obtained by the descent method used in Hopfield's storage prescription, has been found in numerical simulations of a 512 node model (Wallace 1986). For the spin-glass case, ultrametricity has only been observed in numerical simulations of the SK model using $N > 64$ spins. Smaller systems are dominated by finite-size effects (see, e.g., Young 1985). For the same reasons, it is not possible to observe ultrametricity directly in Boltzmann machines with only ten units. Unfortunately, because the number of learning cycles to reach convergence increases very rapidly with N (Ackley *et al* 1985), and because each learning cycle requires longer annealings as N increases, it would be extremely difficult to simulate large

enough instances of the encoder problem to search for ultrametricity directly. However, it is possible to gain indirect evidence by comparison with a study of the SK model for $4 \leq N \leq 20$ spins by Young and Kirkpatrick (1982).

Since Boltzmann machines involve global energy optimisations using simulated annealing, it is the low-energy states that are of interest. For these small realisations of the SK model, Young and Kirkpatrick (1982) found that the number of spins, ΔN, which have to be flipped to go from the ground state to the first excited state on average varies as

$$\Delta N = 0.72 N^{1/2}. \tag{12}$$

For $N = 10$, $\Delta N = 2.3$. In our 4-2-4 encoder simulations, we can take the final sets of w, compute the 1024 energies and calculate ΔN directly. If ΔN is found to be close to 2.3, then the distribution of low-energy states is similar to that in small instances of the SK model and it would be reasonable to surmise that large Boltzmann machines might show ultrametric structure in the Hamming distance, which is the natural distance measure. If, on the other hand, there are no correlations between low-energy states, they should be distributed randomly, which implies $\Delta N = N/2 = 5$. Figure 3(a) shows a histogram of the Hamming distance (number of flips) from the ground state to the first excited states for all 150 simulations. (There are no significant differences between machines which converged and those which did not.) The results peak strongly around the random value $N/2$ and the only sign of SK-like behaviour is in a small number of machines where $\Delta N = 2$ or 3. There is no reason to support that there is anything special about the first excited state, and figures 3(b)-(d) show similar results for the

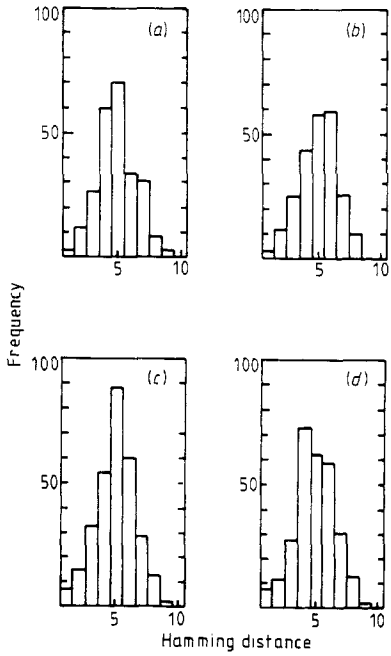

Figure 3. The Hamming distance of low-energy excited states from the ground state for all 150 normal T machines. (a) Ground state to first excited state; (b) ground state to second excited state; (c) ground state to third excited state; (d) ground state to fourth excited state.

next three low-lying states. For those machines which have $\Delta N = 2$ or 3 for the ground state to first excited state transition, it is found that the number of spin flips from the ground state to the second, third and fourth excited states are close to random values. Thus there is no evidence for SK spin-glass behaviour in any of the Boltzmann machine simulations.

6. Conclusions

The networks of artificial 'neurons' proposed recently are novel variants of the Ising model which differ from those that arise in disordered magnetic systems. We have investigated some of the statistical mechanics of one of the more promising models: Hinton and Sejnowski's Boltzmann machine. It is found that the annealing schedule proposed by Ackley *et al* is adequate to attain a Boltzmann distribution of states, on which the algorithm for the minimisation of G depends. However, the necessity to reach a Boltzmann distribution means that the algorithm will require massive computations for large networks. Furthermore, there is a window of annealing temperatures at which learning is possible. We have used direct calculations of the partition function to characterise the number of states which are thermally accessible at effective annealing temperatures. The density of states gives some insight into the sensitivity of the learning rate to annealing temperatures.

Of the better known physical systems, these networks are closest to spin glasses. We have therefore looked for similarities to well characterised spin-glass models such as the SK model. In simulations of small systems, it is not possible to observe such interesting phenomena as ultrametricity directly. However, we are able to make comparisons with previous studies of small instances of the SK spin glass. We find a different structure in the low-energy states which result from the Boltzmann machine learning algorithm. In Boltzmann machines, a random distribution of low-energy states in terms of Hamming distance is observed.

Acknowledgments

I thank David Wallace, David Sherrington, Scott Kirkpatrick and Elizabeth Gardner for useful discussions, and a referee for some helpful suggestions.

References

Ackley D H, Hinton G E and Sejnowski T J 1985 *Cog. Sci.* **9** 147
Amit D J, Gutfreund H and Sompolinsky 1985a *Phys. Rev. Lett.* **55** 1530
—— 1985b *Phys. Rev.* A **32** 1007
Bruce A D, Canning A, Forrest B, Gardner E and Wallace D J 1986 *Proc. Conf. on Neural Networks for Computing, Utah* (New York: American Institute of Physics)
Gardner E 1986 *J. Phys. A: Math. Gen.* **19** L1047
Hinton G E and Sejnowski T J 1983 *Proc. 5th Ann. Conf. Cog. Sci. Soc. Rochester, NY* p 1
Hopfield J J 1982 *Proc. Natl Acad. Sci. USA* **79** 2554
Jardine N and Sibson R 1971 *Mathematical Taxonomy* (New York: Wiley)

Kirkpatrick S, Gelatt C D and Vecchi M P 1983 *Science* **220** 671
Kirkpatrick S and Sherrington D 1978 *Phys. Rev.* B **17** 4384
Kullback S 1959 *Information Theory and Statistics* (New York: Wiley)
Mezard M, Parisi G, Sourlas N, Toulouse G and Virasoro M 1984 *Phys. Rev. Lett.* **52** 1156
Rumelhart D E, Hinton G E and Williams R J 1986 *Parallel Distributed Processing* ed D E Rumelhart and J L McClelland (Cambridge, MA: MIT Press)
Sherrington D and Kirkpatrick S 1975 *Phys. Rev. Lett.* **32** 1792
Wallace D J 1985 *Proc. Conf. on Advances in Lattice Gauge Theory, Tallahassee* (Singapore: World Scientific)
Young A P 1985 *J. Appl. Phys.* **57** 3361
Young A P and Kirkpatrick S 1982 *Phys. Rev.* B **25** 440