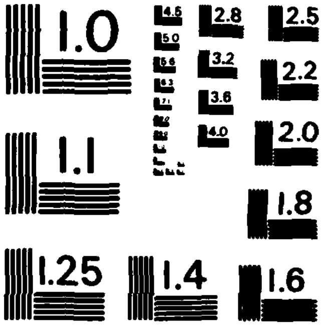
AD-A160 644 BOLTZMANN MACHINES AND ARTIFICIAL INTELLIGENCE(U) ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND) D G BOUNDS JUN 85 RSRE-MEMO-3826 DRIC-BR-97007 1/1

UNCLASSIFIED F/G 17/7 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNLIMITED

BR97007

2

AD-A160 644

RSRE
MEMORANDUM No. 3826

ROYAL SIGNALS & RADAR ESTABLISHMENT

BOLTZMANN MACHINES AND ARTIFICIAL INTELLIGENCE

Author: D G Bounds

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

DTIC
ELECTE
OCT 2 9 1985
D

RSRE MEMORANDUM No. 3826

DTIC FILE COPY

UNLIMITED

85 10 28 061

UNLIMITED

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3826

TITLE: BOLTZMANN MACHINES AND ARTIFICIAL INTELLIGENCE

AUTHOR: D G Bounds

DATE: June 1985

SUMMARY

Ackley, Hinton and Sejnowski have recently proposed an algorithm, named a Boltzmann Machine, which is capable of learning to recognise the underlying structure in a set of patterns presented to it. The main purposes of this memorandum are: to introduce Boltzmann Machines to those who are not familiar with them; to outline how Boltzmann Machines may prove useful in the knowledge acquisition problem in artificial intelligence; to report some new results for a model problem; and to sketch out the relationship between Boltzmann Machines and the spin-glass problem.

Accession For
NTIS GRA&I ☒
DTIC TAB □
Unannounced □
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

QUALITY INSPECTED 4

Copyright
C
Controller HMSO London
1985

UNLIMITED

RSRE MEMORANDUM 3826

BOLTZMANN MACHINES AND ARTIFICIAL INTELLIGENCE

D.G. Bounds

CONTENTS

1. INTRODUCTION

The useful property of a Boltzmann Machine is that it can learn the underlying constraints which characterise a problem domain simply by being shown examples from that domain.

There is a long history of attempts to write computer programs that can learn from experience. Such systems fall broadly into two categories. The first or "top down" approach uses mathematical logic to follow the consequences of a set of general rules of thumb, usually in conjunction with some further rules which are specific to the problem domain being studied. New knowledge learned is encoded as additional rules. These programs are most easily realised using a symbolic reasoning language such as Lisp or Prolog. Perhaps the most impressive example of this sort is Lenat's Eurisko program [Lenat, 1984(a,b)] which has shown useful behaviour in fields as diverse as number theory and VLSI design, and has also enabled its author, who had never entered the competition before, to win a national U.S. war game championship two years running.

The second, or "bottom up" approach originated in attempts to model neural networks. Such models consist of a network of simple processing units connected together by links of different strengths. Perception is viewed as "a parallel, distributed computation in which a large network settles into a particular state" under the influence of sensory input [Hinton, 1984; quoted in Bridle, 1984]. New knowledge is encoded by changes in the strengths of the connections. These systems are similar to models of other physical systems in that they involve numerical rather than symbolic computations, and the key to their behaviour lies in questions of stability rather than mathematical logic. Systems based on neural networks were a major research activity in the 1960's but work almost ceased following the publication of a critical study [Minsky, 1968] of the most popular model (perceptrons). Nevertheless, these models did give some insight into theoretical computer science [Minsky, 1968] and have also resulted in an interesting pattern recognition device, WISARD [Aleksander, 1983]. There has been a resurgence of interest recently in network models [Feldman, 1982] which stems in part from their relevance to parallel computation. A second impetus is that Kirkpatrick has provided a very useful tool, optimisation by simulated annealing [Kirkpatrick, 1983], for finding the most stable states of network models.

One promising network model, named a Boltzmann Machine by its inventors, has recently been described in a series of papers by Hinton, Sejnowski and coworkers [Ackley, 1985; Hinton, 1983(a,b); Hinton, 1984]. The principal advantages of this formulation over previous models are as follows.
(1) Boltzmann Machines have a general learning rule which does not involve assumptions about the problem under study.
(2) They are capable of modelling higher order constraints, that is intrinsic properties of the data which are not directly constrained by the input. The ability to do this was one of the key factors which perceptron models lacked. This matter is discussed further in Section 4.
(3) They have a mathematical structure based on classical statistical mechanics, to which some information theory is added. It is therefore possible to use the techniques of statistical mechanics to study the behaviour of the Machines. Such powerful methods of analysis make it much easier to understand how the Machines should be operated.
(4) They have a parallel architecture. This is vital because very large networks will be necessary for practical applications.

Boltzmann Machines may have useful applications in visual image processing [Hinton, 1983(b)], speech processing [Bridle, 1984], synthetic aperture radar [Luttrell, 1985], and communications [Pritchard, 1984] among others. The suitability of the Boltzmann Machine architecture for various computational tasks common to artificial intelligence programs (for example set intersection, transitive closure, etc) have been considered by Fahlman, Hinton and Sejnowski [Fahlman, 1983]. One purpose of this memorandum is to outline how Boltzmann Machines might aid or even take the place of the human domain expert when building expert systems: a combination of "top down" and "bottom up" approaches.

In Section 2 a general Boltzmann Machine is described in enough detail to enable the reader to understand what it is and how it works. In Section 3 the use of Boltzmann Machines to get knowledge into Expert Systems is discussed. Section 4 is a brief account of what useful lessons can be learned from earlier work on perceptrons. One main conclusion which emerges is that it is vital to get a thorough understanding of small problems in order to appreciate the limitations of larger systems. For this reason a model problem has been examined in some detail and the preliminary results are reported in Section 5. Section 6 outlines the relation of Boltzmann Machines to spin-glasses. Spin-glass physics may provide insights both into technical details of how Boltzmann Machines may best be operated and their ultimate limitations. Section 7 contains the conclusions.

2. BOLTZMANN MACHINES

This section is not a complete technical description of the Boltzmann Machine algorithm. For that the reader should consult the elegant papers of Hinton, Sejnowski and coworkers [Ackley, 1985; Hinton, 1983(a,b); Hinton, 1984]. The aim here is an overview which is intended to amplify those aspects which are the most important.

A Boltzmann Machine consists of a set of units. Except in a few special cases, some units are devoted to input, some to output, and the rest are internal units which do neither. At any instant a Boltzmann Machine is running in one of three modes: a training mode where both inputs and outputs are clamped; a free-running mode where neither inputs nor outputs are clamped; or an operational ("testing for completion" in Hinton's words [Hinton, 1984]) mode where some inputs are clamped and the Machine, if it has learned properly, produces appropriate outputs. The free-running mode is necessary because data collected in this mode is used together with data collected in the training mode to provide the feedback mechanism by which the machine learns. A learning cycle is a sequence where the machine runs in training and free-running modes alternately until a set of patterns has been shown, after which the Machine's internal model is updated. After enough learning cycles the Machine obtains an optimum model and is then operational. How does this come about?

Each member of the set of units $S=\{s_i\}$ may be in one of two states: on or off. The units are joined by bidirectional links, $W=\{w_{ij}\}$, which may take positive or negative values. A positive link between two units implies that they tend to be on or off together; a negative link means that they teno to adopt opposite states. The links are symmetric, $w_{ji}=w_{ij}$.

This structure is sufficient to define a configurational "energy" of the whole system of n units. In the work of Hinton et al. the numerical values corresponding to on and off are 1 and 0 respectively. Provided that the on state of a unit i is represented by $s_i=1$, the configurational energy is

$$E(W,S) = - \sum_{i=1}^{n} \sum_{j=i}^{n} w_{ij}\, s_i\, s_j \qquad (1)$$

The self terms in eqn. (1) are threshold terms which appear because of the theory's origin in neural network modelling. The details are not relevant here, but they may be found in [Hinton, 1984; see also Hopfield, 1982].

A pattern is presented to the Machine by constraining its input and output units to be in particular states. An energy may then be assigned to the pattern by finding the global minimum of eqn (1) subject to the constraints imposed by the pattern. Let σ be the subset of S constrained by the pattern. For pattern α the state of σ is σ_α. Then $E(W,S) = E(W,\sigma_\alpha,\sigma')$ where σ' is the complement of σ. One seeks the global minimum of $E(W,\sigma_\alpha,\sigma')$ w.r.t. σ'. Since this global minimum is characteristic of pattern α, we label it $E_\alpha(W,\sigma_\alpha,\sigma')$. Thus the energy is minimised by altering the states of all units which are not constrained by the pattern; the link strengths being held constant. The first problem, then, is to find an effective global minimisation method.

The only method which is guaranteed to find the global minimum is an exhaustive search of all possible values of the cost function. Exhaustive search is computationally out of the question for functions of many variables, though no other method stands a reasonable chance of finding the global minimum on surfaces where the minimum is a deep, narrow, isolated well as in figure 1(a). Fortunately the problem here is not so pathological. For many cost functions, including the free energy of spin-glasses, many of the local minima are almost as deep as the global minimum (see fig 1(b)). In these cases it is sufficient to find any one of these low-lying local minima because the global minimum is not significantly lower. Of course, it is still necessary to escape from higher local minima.

The solution is to adopt a method which has been widely used in computer simulation studies of condensed matter, especially in spin-glass physics. The technique, optimistion by simulated annealing (OSA), has recently been applied successfully to a wide range of optimisation problems including VLSI design [Kirkpatrick, 1983]. The basic idea is this. An optimisation problem involves a cost function to be minimised, in this case the configurational energy, and a set of configurations generated by trial moves. Gradient methods such as steepest descent accept only those moves which reduce the cost function. Such algorithms have no way to escape from local minima, low or high, which is not the behaviour required. To avoid this one can choose new configurations randomly and take the lowest value of the cost function found after a large number of random choices. Randomising algorithms can accept moves that result in higher values of the cost function, but these moves are accepted blindly. Since the probability of finding a near-optimal configuration is proportional to the number of near-optimal configurations divided by the total number of possible configurations, randomising algorithms perform less well as the dimensionality of the search space increases. OSA also allows some moves which increase the cost function, but the moves are accepted in a controlled manner. The basis of the method was proposed first by Metropolis et al. [Metropolis, 1953]. If a trial move decreases the cost function it is accepted. If a trial move increases the cost function, it is accepted with probability $\exp(- E/\tilde{T})$ where E is the increase in the cost function and $\tilde{T}$ is a quantity with the same dimensions ($\tilde{T}=kT$ for the energy minimisation considered here). Because of the physical analogy, $\tilde{T}$ is referred to as a temperature in the OSA literature regardless of what cost function is involved. From hereon we shall adopt this convention, drop the tilde and set k=1.

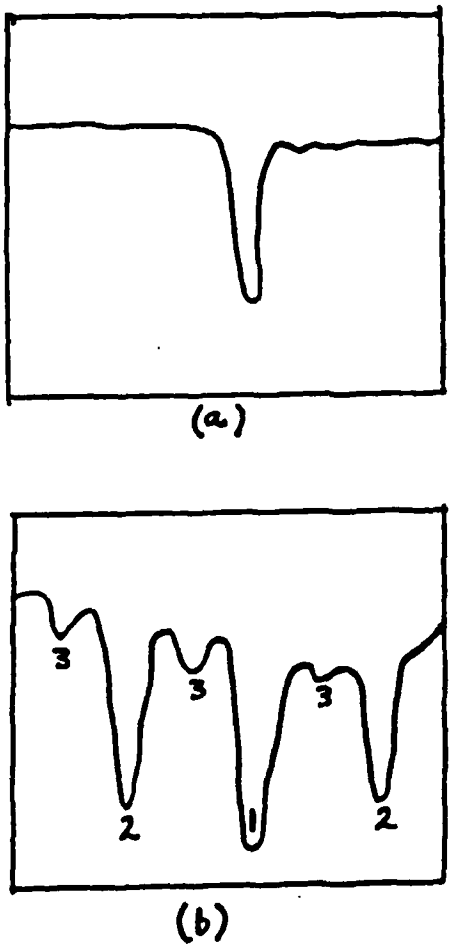


Fig. 1 (a) A surface with a single deep global minimum
(b) A surface with several minimum with values of the cost function similar to the global minimum value. 1 is the global minimum; 2 are acceptable local minima; 3 are high local minima which must be escaped from.

A system evolving under these rules will eventually reach thermal equilibrium at any given temperature, and the relative probabilities of two global states will be given subsequently by the Boltzmann distribution

$$(P_\lambda / P_\mu) = \exp[-(E_\lambda - E_\mu)/T] \qquad (2)$$

where P_λ is the probability of being in global state λ with energy E_λ. At high temperatures the probability of accepting uphill moves is greater and equilibrium is reached more quickly. At low temperatures equilibration takes longer but the system is more heavily weighted towards low energy states. A good strategy is therefore to begin the search at high temperature and then slowly reduce T: hence the term "optimisation by simulated annealing". Note that this method is unlike gradient algorithms in that it does not find the minimum then stay there. The algorithm merely spends a larger proportion of its time near the minimum as the temperature is lowered. To stay in the global minimum would require T=0 in which case equilibration would take infinite time.

We now come to a central problem in all network models, one which Boltzmann Machines solve in a novel fashion. Having captured some aspect of a set of patterns by minimising $E(W,\sigma,\sigma')$ w.r.t. σ' for each pattern, how can this knowledge be encoded by altering the strengths of the connections? To see how a Boltzmann Machine does this, consider how many patterns it is possible to capture.

Let the n units be divided into two subsets: a non-empty set V of visible units with v members and a set H of hidden units with h members *. The visible units are the Machine's interface with the outside world, the input and output units. Only visible units may be clamped by a training pattern. The hidden units are the internal units which allow the Machine to model the environment.

Since each unit may be either on or off, there are 2^v possible global states amongst the visible set. It should therefore be possible to model 2^v patterns. However, the energy expression (1) only has $(v+h)[1 + (v+h-1)/2]$ possible values, and many of these are degenerate. Hence it is not possible to get a perfect internal model capable of distinguishing between all possible 2^v states unless $(v+h)[1 + (v+h-1)/2] \geqslant 2^v$, which requires exponentially large h. While this may be feasible for very small v, it will be impossible for any useful application. Here we have used "pattern" to denote a state of all visible units, input and output, i.e. the whole external environment of the machine. The argument is also true if by pattern one means a state of the input units only. The important point is that the lesser part of the inequality is exponential. Note that although in any practical application one would use some of the visible units for output, the formalism does not require this. In fact, in the encoder problem described in section 5.1 output units are not necessary since it is clear from the connection strengths when learning is complete.

---

* footnote:

The following relationships hold between the sets V,H,S,σ and σ':

(i) $V + H = S$;

in training mode:

(ii) $\sigma = V$, $\sigma' = H$;

in free-running mode:

(iii) $\sigma = 0$, $\sigma' = S$;

in operational mode:

(iv) $\sigma = A$, where $A \subset V$,
$\sigma' = H + A' = S - A$

Because a perfect model is not usually possible in practice, the problem is to obtain an optimum model given some smaller number of hidden units. One measure of the goodness of a model is the G metric or information gain introduced by Kullback [Kullback, 1959]:

$$G = \sum_{\alpha} P_{\alpha} \ln[P_{\alpha} / P^{f}_{\alpha}] \qquad (3)*$$

where P_α is the probability of the network being in state α of the visible units when the state is determined by the input and P^f_α is the corresponding probability when the machine is running freely without a 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 all input patterns perfectly. Otherwise G is positive ana the best model is that which minimises G. Since P_α depends on E_α:

$$G(W) = \sum_{\alpha} P_{\alpha}(E_{\alpha}) \ln[P_{\alpha}(E_{\alpha})/P^{f}_{\alpha}(E_{\alpha})]$$

$$= \sum_{\alpha} P_{\alpha}(W,\sigma_{\alpha},\sigma') \ln[P_{\alpha}(W,\sigma_{\alpha},\sigma')/P^{f}_{\alpha}(W,\sigma_{\alpha},\sigma') \qquad (4)$$

The question posed three paragraphs ago can now be answered: the strengths of the connections should be altered in order to minimise G.

The second, and vital, reason why OSA was used in the energy minimisation is now evident. After equilibration the distribution of global states is the Boltzmann distribution. Because the log probability of a global state is then proportional to its energy, and the energy is a linear function of the strengths of the links, there is a simple expression for the partial derivatives $\partial G/\partial w_{ij}$ (see [Hinton, 1984] appendix A):

$$\partial G/\partial w_{ij} = -(1/T)(p_{ij} - p^{f}_{ij}) \qquad (5)$$

where

$$p_{ij} \overset{def}{=} \sum_{\alpha}\sum_{\beta} P_{\alpha\beta}\, s^{\alpha\beta}_{i}\, s^{\alpha\beta}_{j} \qquad (6)$$

* footnote

In a physical system, the interpretation of G in thermodynamic terms is that it is the entropy produced in the system by the change in global state from {P^f} to {P} [Schlogl, 1971].

and

$$p^f_{ij} \overset{def}{=} \sum_\lambda \sum_\mu p^f_{\lambda\mu} \, s_i^{\lambda\mu} \, s_j^{\lambda\mu} \qquad (7).$$

The outer summations in (6) and (7) run over states of the visible units; the inner loops over states of the hidden units. $s_i^{\alpha\beta}$ is therefore the state of unit i when the visible units of the Machine are in state α and the hidden units are in state β. One therefore has the derivatives necessary to minimise G, with the caveat that $\{p_{ij}\}$ and $\{p^f_{ij}\}$ must be measured when the machine has reached equilibrium. When the numerical values corresponding to a unit being on or off are 1 and 0 respectively, p_{ij} and p^f_{ij} are trivial to calculate since p_{ij} is then just the average probability that both units are in the on state when a pattern is clamping the visible units, and p^f_{ij} is the same quantity when running free of input.

Two further points should be made before leaving this discussion. First, it is not necessary to have a Boltzmann distribution over global states: the requirement is really that ln P_α must be a linear function of W. Second, except for the case where there are no hidden units [Hinton, 1984], G-space may also have local minima. The problem of finding the global minimum therefore arises in G-space just as it did in the energy minimisation and the same remarks apply. It is possible in principle to minimise G by annealing. However, in practice this approach has not been adopted even for the simple model problems in [Hinton, 1984]. It is not hard to see why.

The minimisation of G is an outer loop. At each step within the loop many annealings are performed to minimise E for each pattern. If there are n units in the Boltzmann Machine then there are $O(n^2)$ links. The energy has to be minimised w.r.t. at most n variables, but G has to be minimised w.r.t. $O(n^2)$ variables. As shown in section 5.3, much longer annealing schedules are needed as the number of variables increases. Annealing to find the minimum in G-space is therefore computationally extremely expensive. In most of the work done so far [Hinton, 1984] occasional uphill steps in G have been made possible by introducing some noise into the estimates of $\{p_{ij}\}$ and $\{p^f_{ij}\}$. This is simply achieved by collecting them over only a rather small number of time steps. It seems likely that less ad hoc methods will replace this method [Derthick, 1984]. However, finding good search methods for G-space is a key problem with Boltzmann Machines and unless it can be solved it is hard to see how Boltzmann Machines will be able to achieve good internal models for useful applications when n is large.

Finally, we have described the Boltzmann Machine algorithm in roughly the same order as the original authors [Hinton,1984; Ackley,1985]. We began with an architecture and energy expression, and introduced G only because the number of possible patterns is exponentially large. A more fundamental view is obtained by turning the argument on its head. The quantity which measures how well the Machine is modelling its environment is G, and the central problem is to find that model which corresponds to the global minimum of G. All the rest: the energy expression, simulated annealing, etc, are simply one computational method to achieve that aim.

3. BOLTZMANN MACHINES AND EXPERT SYSTEMS

Expert systems are computer programs in which a set of rules is used to mimic the behaviour of a human expert in some well-defined problem domain such as medical diagnosis. A user interacts with an expert systems in the same way that he would consult a specialist; explaining the problem, giving further details where required, and so on. In return he obtains advice and possible solutions to his problems.

The rules which classify objects in the problem domain are obtained either explicitly from experts (a programmer, or "knowledge engineer", observes the specialist solving some model problems and tries to encode what he does), or indirectly from experts via computer induction aids such as TIERESIAS [Davis, 1982] (a suite of computer programs which do, amongst other things, the job of the knowledge engineer). In either case the task of getting knowledge out of the head of the specialist and into the computer program is laborious. This "knowledge acquisition" problem is widely regarded as the main limiting factor in the development of new expert systems [Feigenbaum, 1984].

It is worth pointing out that the performance of current expert systems is at best only as good as that of the specialist on whom the system was modelled. Specialists often disagree. A graphic illustration of this problem has been described by Hopford [Hopford, 1984]. ARE Portland were engaged on a project to build an expert system to give advice on sonar lineups. The problem is to produce the best lineup of sonar devices to detect a given target at the longest possible range, with the equipment servicable at the time and in the oceanographic conditions which prevail.

An expert system was built with the aid of two naval specialists. The first, who was also the knowledge engineer, had a good working knowledge of sonar. The second is the navy's leading expert on sonar lineups and is the author of the navy's tactical manuals on the subject. The technique used was that the two specialists wrote the first set of rough rules then tried them on some sample problems in order to refine the rule set. To test the final system, six RN sonar officers were asked to produce sonar lineups for a set of 50 sample scenarios. Considerable difficulty was found in getting the sonar officers to agree. Furthermore, large differences became apparent between the first two specialists. On a scale where complete agreement between them on all 50 samples implies a correlation coefficient of 1, and complete disagreement minus 1, the correlation coefficient between the two specialists peaked around +0.2. In other words the correlation would have been almost as good if they had made random decisions. The correlations between the expert system and each specialist was somewhat better, so to some extent the model mediated between the two experts.

A further problem arises because expert systems, like human experts, often have to make decisions on the basis of incomplete evidence. Furthermore, they do so using rules of thumb which are plausible but not, usually, absolute. Most systems therefore need to reason in the presence of uncertainty in both their rules and their data. At this point the elegant rigour of logic programming tends to dissolve into numerical "confidence factors" (judicious guesses provided by the specialist) associated with each rule. Reasoning - finding the most plausible outcome for some given set of data - now becomes a matter of satisfying conflicting weak constraints. An expert systems with these properties begins to look like a Boltzmann Machine.

It is helpful to consider a concrete example. We choose MYCIN [Shortliffe, 1976] which was one of the most successful of the early expert systems and which has provided the model for many systems built since. MYCIN was developed to provide consultative advice on the diagnosis and treatment of infectious diseases. Time is of the essence in this problem and it is largely this which makes reasoning with uncertainty necessary.. For example a specimen from a patient may show signs of bacterial growth after several hours, but it may take several days for a positive identification of the organism. The physician must therefore decide whether or not to start treatment and what drugs to use before there is enough information for certain identification. Obviously many military, and for that matter industrial and commercial, decisions must be taken in the same climate.

The MYCIN program has about 450 rules, each with associated confidence factors. A typical rule is:

IF: { (1) the site of the culture is blood, and
(2) the organism is able to grow aerobically, and
(3) the organism is able to grow anaerobically }
THEN: there is evidence that the aerobicity of the organism is
facultative (0.8)
or anaerobic (0.2).

The confidence factors are in parentheses.

Let H_1 be the hypothesis that the situation clause of the IF-THEN rule (enclosed in { } brackets) is true; H_2 be the hypothesis that the organism is facultative; and H_3 be the hypothesis that the organism is anaerobic. If the truth of hypothesis H_i is represented by the state (on or off) of a visible unit s_i in a Boltzmann Machine, then the confidence factors imply that the link strengths in the Boltzmann Machine are $w_{12}=0.8$ and $w_{13}=0.2$. (This rule makes no statement about w_{23}.) Thus the MYCIN rule structure maps onto a Boltzmann Machine. The rules define the connectivity of the units and the confidence factors define the link strengths. Note that the way H_1 is written suggests that some of the other rules in the program require H_1 to be split into three separate hypotheses. This does not affect the argument. It simply means that the three separate hypotheses are represented by three visible units, and H_1 is represented by a hidden unit whose state depends upon the logical AND of the three separate hypotheses. This is just the sort of higher order constraint which the hidden units in a Boltzmann Machine are capable of capturing [Hinton, 1984; Ackley, 1985].

Because confidence factors are related to the link strengths of a Boltzmann Machine, it is possible to learn them if a set of suitable test data exists, and thus avoid the necessity to make guesses by trial and error until the program behaves in the correct way. This is an important advantage since the numerical stability of pathways through the network of confidence factors is a problem which has not been solved by the writers of expert systems. In expert systems confidence factors are attempts to put numbers on non-numerical concepts such as "possible" and "probable". One would like the reasoning process to be robust to the values chosen. If changing a confidence factor somewhere in the network from, say, 0.60 to 0.61 produces a different answer for the same set of input, then the system is unstable and one would have little confidence in the results. This problem does occur in expert systems. During construction of MYCIN, for example, there were several cases in which a new rule caused existing rules to be applied incorrectly or to cease being applied altogether [Cohen, 1982]. The problem was not solved completely, and for this reason MYCIN does not offer a "best" solution but a bundle of the most likely alternatives [Clancy,1984]. Clearly, the stability problem becomes more acute as the number of rules increases.

Several combinations of Boltzmann Machines and expert systems can be envisaged depending on what proportion of the overall task is to be done by the Boltzmann Machine. In the simplest case both rules and rough estimates of the confidence factors could be set by specialists. Given some training data, a Boltzmann Machine could refine the confidence factors and ensure a stable network. However, there is a further advantage since the Boltzmann Machine could constantly update the confidence factors on the basis of cases it encountered in operation. Such a hybrid system should therefore be able to learn from its operational experience. This is possible in very few expert systems.

At a higher level of Boltzmann Machine involvment, specialists might specify the rules but not give values for any confidence factors. These would need to be learned entirely by the Boltzmann Machine; a much harder task. There is a spectrum of possibilities where specialists set only those confidence factors they have some knowledge about and the Boltzmann Machine is left to discover the rest. The more prior knowledge about a problem that can be incorporated, in this case by pre-setting the strengths of some connections, the more quickly a Boltzmann Machine should be able to find a good solution.

A third possibility would require the Boltzmann Machine to cerive both the rules and the confidence factors. Unless at least some rules were given, it seems likely that only small problems could be tackled in this way. Where prior knowledge is available it makes sense to incorporate it at the outset, though one strength of the Boltzmann Machine is precisely this ability to find a solution in the absence of prior knowledge.

To summarise, if Boltzmann Machines can be scaled up to a useful size then they offer the following advantages for expert systems. They provide an intelligent aid for producing the knowledge base and the inference engine. They make the knowledge base and the reasoning mechanism less dependent on any individual specialist. They should lead to stable networks of confidence factors, in contrast to the present methods which usually ignore this problem. And they provide a capability for continuous learning during the operational life of the system.

All this is possible only if a set of training data exists. While this is not available for every problem, it is for many. After all, the human experts whose skills are programmed into expert systems got their knowledge from somewhere; and the more highly specialised the task, the more likely that their relevent knowledge came from direct observations rather than secondhand from textbooks or lectures. The six sonar officer's lineups for the fifty scenarios are just the sort of training data needed, and the fact that they didn't all agree provides the competing weak constraints that Boltzmann Machines were designed for.

Finally, although a medical example was chosen, military expert systems exist with similar structures. For example, Hopford describes an American surface ship or submarine mounted surveillance system called CLAIMS. "CLAIMS is an enormously complex knowledge-based system. It is multi-level, dynamic and operates in real-time with imprecise data." [Hopford, 1984]. A typical production rule is:

IF: the source was lost due to fade-out in the near past, and
a similar source started up in another frequency, and
the locations of the two sources are close

THEN: they are the same source with confidence factor 0.3.

This rule is of the same form as the MYCIN example shown earlier. A vast body of training data is available for this problem, at least in principle, from naval exercises where submarines have attempted to escape sonar detection.

It is helpful to consider a concrete example. We choose MYCIN [Shortliffe, 1976] which was one of the most successful of the early expert systems and which has provided the model for many systems built since. MYCIN was developed to provide consultative advice on the diagnosis and treatment of infectious diseases. Time is of the essence in this problem and it is largely this which makes reasoning with uncertainty necessary. For example a specimen from a patient may show signs of bacterial growth after several hours, but it may take several days for a positive identification of the organism. The physician must therefore decide whether or not to start treatment and what drugs to use before there is enough information for certain identification. Obviously many military, and for that matter industrial and commercial, decisions must be taken in the same climate.

The MYCIN program has about 450 rules, each with associated confidence factors. A typical rule is:

```
IF: { (1)  the site of the culture is blood, and
      (2)  the organism is able to grow aerobically, and
      (3)  the organism is able to grow anaerobically }
THEN:  there is evidence that the aerobicity of the organism is
                 facultative (0.8)
              or anaerobic (0.2).
```

The confidence factors are in parentheses.

Let H_1 be the hypothesis that the situation clause of the IF-THEN rule (enclosed in { } brackets) is true; H_2 be the hypothesis that the organism is facultative; and H_3 be the hypothesis that the organism is anaerobic. If the truth of hypothesis H_i is represented by the state (on or off) of a visible unit s_i in a Boltzmann Machine, then the confidence factors imply that the link strengths in the Boltzmann Machine are $w_{12}=0.8$ and $w_{13}=0.2$. (This rule makes no statement about w_{23}.) Thus the MYCIN rule structure maps onto a Boltzmann Machine. The rules define the connectivity of the units and the confidence factors define the link strengths. Note that the way H_1 is written suggests that some of the other rules in the program require H_1 to be split into three separate hypotheses. This does not affect the argument. It simply means that the three separate hypotheses are represented by three visible units, and H_1 is represented by a hidden unit whose state depends upon the logical AND of the three separate hypotheses. This is just the sort of higher order constraint which the hidden units in a Boltzmann Machine are capable of capturing [Hinton, 1984; Ackley, 1985].

Because confidence factors are related to the link strengths of a Boltzmann Machine, it is possible to learn them if a set of suitable test data exists, and thus avoid the necessity to make guesses by trial and error until the program behaves in the correct way. This is an important advantage since the numerical stability of pathways through the network of confidence factors is a problem which has not been solved by the writers of expert systems. In expert systems confidence factors are attempts to put numbers on non-numerical concepts such as "possible" and "probable". One would like the reasoning process to be robust to the values chosen. If changing a confidence factor somewhere in the network from, say, 0.60 to 0.61 produces a different answer for the same set of input, then the system is unstable and one would have little confidence in the results. This problem does occur in expert systems. During construction of MYCIN, for example, there were several cases in which a new rule caused existing rules to be applied incorrectly or to cease being applied altogether [Cohen, 1982]. The problem was not solved completely, and for this reason MYCIN does not offer a "best" solution but a bundle of the most likely alternatives [Clancy,1984]. Clearly, the stability problem becomes more acute as the number of rules increases.

Several combinations of Boltzmann Machines and expert systems can be envisaged depending on what proportion of the overall task is to be done by the Boltzmann Machine. In the simplest case both rules and rough estimates of the confidence factors could be set by specialists. Given some training data, a Boltzmann Machine could refine the confidence factors and ensure a stable network. However, there is a further advantage since the Boltzmann Machine could constantly update the confidence factors on the basis of cases it encountered in operation. Such a hybrid system should therefore be able to learn from its operational experience. This is possible in very few expert systems.

At a higher level of Boltzmann Machine involvment, specialists might specify the rules but not give values for any confidence factors. These would need to be learned entirely by the Boltzmann Machine; a much harder task. There is a spectrum of possibilities where specialists set only those confidence factors they have some knowledge about and the Boltzmann Machine is left to discover the rest. The more prior knowledge about a problem that can be incorporated, in this case by pre-setting the strengths of some connections, the more quickly a Boltzmann Machine should be able to find a good solution.

A third possibility would require the Boltzmann Machine to derive both the rules and the confidence factors. Unless at least some rules were given, it seems likely that only small problems could be tackled in this way. Where prior knowledge is available it makes sense to incorporate it at the outset, though one strength of the Boltzmann Machine is precisely this ability to find a solution in the absence of prior knowledge.

To summarise, if Boltzmann Machines can be scaled up to a useful size then they offer the following advantages for expert systems. They provide an intelligent aid for producing the knowledge base and the inference engine. They make the knowledge base and the reasoning mechanism less dependent on any individual specialist. They should lead to stable networks of confidence factors, in contrast to the present methods which usually ignore this problem. And they provide a capability for continuous learning during the operational life of the system.

All this is possible only if a set of training data exists. While this is not available for every problem, it is for many. After all, the human experts whose skills are programmed into expert systems got their knowledge from somewhere; and the more highly specialised the task, the more likely that their relevent knowledge came from direct observations rather than secondhand from textbooks or lectures. The six sonar officer's lineups for the fifty scenarios are just the sort of training data needed, and the fact that they didn't all agree provides the competing weak constraints that Boltzmann Machines were designed for.

Finally, although a medical example was chosen, military expert systems exist with similar structures. For example, Hopford describes an American surface ship or submarine mounted surveillance system called CLAIMS. "CLAIMS is an enormously complex knowledge-based system. It is multi-level, dynamic and operates in real-time with imprecise data." [Hopford, 1984]. A typical production rule is:

IF: the source was lost due to fade-out in the near past, and
a similar source started up in another frequency, and
the locations of the two sources are close

THEN: they are the same source with confidence factor 0.3.

This rule is of the same form as the MYCIN example shown earlier. A vast body of training data is available for this problem, at least in principle, from naval exercises where submarines have attempted to escape sonar detection.

4. BOLTZMANN MACHINES AND PERCEPTRONS

In the Introduction it was noted that systems based on neural networks, of which the perceptron models were typical, were a major research activity in the 1960's until the publication of Minsky and Papert's book [Minsky, 1968]. After that work almost ceased. What went wrong? Do Boltzmann Machines overcome the problems which led to the demise of perceptrons, or is a new generation of scientists falling into the same traps 25 years later?

Boltzmann Machines have many similarities to perceptrons. Both are neural network models and are intended to give some insight into how the brain learns. Both have feedback mechanisms where the strengths of connections are increased or decreased depending on how well the device is modelling its environment. Both are amenable to parallel computation, and are probably suited to analogue devices. And in both cases there was an explosion of interest after publication of the first papers. "Rosenblatt's (the inventor of perceptrons) schemes quickly took root, and soon there were perhaps as many as a hundred groups, large and small, experimenting with the model either as a 'learning machine' or in the guise of 'adaptive' or self-organising' networks or 'automatic control' systems. The results of these hundreds of projects and experiments were generally disappointing and the explanations inconclusive. The machines usually work quite well on very simple problems but deteriorate very rapidly as the tasks assigned to them get harder. The situation isn't usually improved much by increasing the size and running time of the system." [Minsky, 1968]. Except that the hundreds of experiments are only just starting and it is not yet known whether they can solve hard problems, the same could be said about Boltzmann Machines today.

There are two main grounds on which perceptrons failed. The first is the credit-assignment problem which arises because to be capable of non-trivial calculations, networks must contain some hidden elements whose states are not directly constrained by the input. These hidden units are there to capture the underlying hidden structure in the patterns; for example, the fact that it is the logical AND of three separate hypotheses which is the important quantity in the MYCIN rule cited earlier. When a network gives the wrong result it can be very difficult to know which of the many connection strengths are wrong. As Minsky and Papert pointed out, perceptrons have no mechanism to solve this problem. Hinton et al. [Hinton, 1984; Ackley, 1985] have argued persuasively and have also provided a convincing demonstration, the shifter problem [Hinton, 1984], that Boltzmann Machines can overcome this problem. This is an important step forward.*

However, it remains an open question whether Boltzmann Machines can solve the second problem which bedevilled perceptrons. This concerns the rate of learning and determines whether they can be scaled up to solve useful problems. One of the cornerstones of perceptron theory was the perceptron convergence theorem. This guaranteed that the learning process would

* footnote:

The credit-assignment problem is not restricted to network models; it also arises in rule-based systems. Consider for example a program with a set of rules to play chess. It is desired to refine the rules on the basis of the program's performance over a number of games of chess. If the program is only told at the end of each game that it has won or lost, it will be very difficult to know which rules should be changed. In rule-based systems the credit-assignment problem will arise whenever a program performs a sequence of actions before receiving any feedback.

eventually find a correct setting of the network parameters if one existed. This must have been one factor which encouraged people to continue to tackle hard problems even in the absence of much success. Minsky and Papert argued that perceptron devices were essentially finite state machines. An optimum state could therefore be found in principle by trying each state in turn. A convergence theorem is then only useful if it says something about the rate of learning relative to a random or an exhaustive search. The perceptron convergence theorem did not do this. In fact, Minsky and Papert showed the existence of a class of problems for which the convergence time increases faster than exponentially with the size of the problem [Minsky, 1968]. The Boltzmann Machine is also guaranteed to find an optimum result. As stated by Hinton et al. [Hinton, 1964], "At present, we have an interesting mathematical result that guarantees that a certain learning procedure will build internal representations which allow the connection strengths to capture the underlying constraints that are implicit in a large ensemble of examples taken from a domain." The crucial question is, how fast?

We do not yet have an answer to this question, so it is not known whether Boltzmann Machines can be scaled up to solve useful problems in acceptable times. However, in view of the history of perceptron research, it seemed more profitable at the present stage to gain some insight by studying a small problem which can be understood than to build a big machine and try to solve a "real" problem. For this reason we have examined Hinton's encoder problem in some detail in order to see how the algorithm scales with problem size. The main reason for choosing this problem is that it is large enough to show interesting behaviour but small enough that thermodynamic quantities can be calculated exactly. So far we have focussed on the energy minimisation. Our preliminary results have revealed some problems which will become increasingly important as the number of units increases. However, we think these problems can be overcome by making some parts of the algorithm more sophisticated and by the use of suitable parallel architectures. The problem of minimising G is much more difficult, but we have found some evidence that the rate of learning is sensitive to the annealing temperatures. Dependence on a familiar physical property gives hope that physical analogies may also be fruitful for this problem, and it seem possible that some recent developments in spin-glass physics may enable more searchable energy landscapes to be constructed. The results are described in detail in Section 5.

5. CALCULATIONS ON THE ENCODER PROBLEM

5.1 DESCRIPTION OF THE PROBLEM

The encoder problem was proposed as a simple abstraction of the task of communicating information among various components of a parallel network [Hinton,1984; Ackley,1985]. It may also be viewed as a simple pattern recognition problem. The visible units are split into two groups, V_1 and V_2, each with ν =v/2 units. All units in V_1 are connected to each other, as are all units within V_2 , but there are no direct connections between V_1 and V_2. Instead, the visible groups communicate via a set of hidden units H with h members. The hidden units are not connected to each other, but each is connected to all units in both visible groups. The problem is to evolve a set of link strengths which allows the visible units to communicate their current state to each other.

In the most general version of this problem each unit in V_1 (and V_2) could be on or off. There are then 2^ν possible states of each group. A more simple version of the problem allows only one unit in a group to be on at any time, in which case there are only ν possible states for the group. In the general case the minimum number of hidden units necessary to permit communication of 2^{ν} states is $h=\nu$; in the simple case it is $h=\log_2 \nu$. When this equality holds we call the network a minimal machine. If there are more hidden units the spare capacity makes it easier for the machine to find a set of codes (i.e. sets of states of hidden units) which solve the problem. However, with spare capacity the power of the learning algorithm becomes apparent. Because the learning algorithm seeks the global minimum of G, which would correspond to the best possible model, having found a solution the machine then goes on to modify the codes so that a maximally spaced code is found [Hinton, 1984; Ackley, 1985].

The particular version of the encoder problem which was investigated by Hinton and coworkers is intermediate between the simple and general cases sketched above. In essence the probability distribution over possible states of the visible groups is strongly biased to favour those where only one unit in a group is on at any time. However, there are small but non-zero probabilities of all other states occuring*. Most of the work was done with $\nu=4$ and h=2, and this system is called the "4-2-4" encoder problem. Full details are given in [Hinton, 1984; Ackley, 1985] and we shall not discuss further technicalities here except where they are necessary to understand the present results.

In encoder problems it is not necessary to run the machine in operational mode in order to see if it has solved the problem; one can tell simply by looking at the link strengths. Hinton has developed a very attractive way of displaying the link strengths [Hinton, 1984], and a typical solution is shown in figure 2 (overleaf) which is taken from [Ackley, 1985].

* footnote:

The set of training patterns consists of ν equiprobable vectors each of length 2ν which specify that one unit in V_1 and the corresponding unit in V_2 are on together with all other units off. Suppose that for all vectors each off bit is set on with probability x and each on bit is set off with probability $(\nu-1)x$. If x=0 one has the simple case and the choice $\nu=4$, h=2 defines a minimal machine. However, this presents a problem if the visible units within V_1 (and V_2) are connected to each other since it leads to large negative link strengths between those units [Hinton, 1984; Ackley, 1985]. This problem can be resolved in two ways: by breaking the links within the visible groups (in which case the first phase of learning - inhibition within the visible groups - becomes unnecessary); or by choosing $x>0$. If x is too large then the probability distribution is no longer biased in favour of states with only one unit in a group on. The system is then closer to the general case and is not solvable with $h<\nu$. The original work [Hinton, 1984; Ackley, 1985] used x=0.05 for $\nu=4$, and the same value is used here.

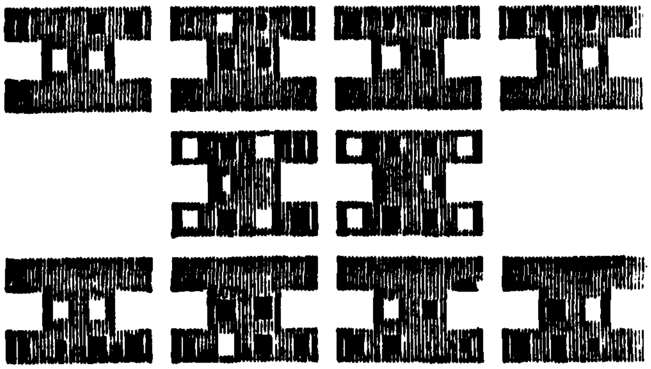


Fig. 2 A solution to the 4-2-4 encoder problem (reprinted from [Ackley, 1985] with permission of the publisher). Each unit is represented by a shaded box. The top four boxes represent the units in V_1, the middle two boxes show the hidden units, and the bottom row shows the units in V_2. Each box shows the strength of that unit's links to all other units in the network. At each position in a box the size of the square indicates the magnitude of a link strength. White squares are positive links; black are negative values. The biases, $\{w_{ii}\}$, appear in the positions which correspond to a unit connected to itself, for example the top left corner of the top left shaded box.

The pattern shown in fig. 2 is a solution because:

(a) each unit in V_1 has developed a unique combination of positive and negative links to the hidden units.

(b) The same is true for V_2.

(c) There is a simple ordered mapping between the codes adopted by V_1 and V_2.

To illustrate how this solves the problem, consider the codes adopted by the left units in V_1 and V_2. Let the units in V_1 be numbered $V_1(1)$ to $V_1(4)$ from the left. The units in H and V_2 are numbered similarly. Now $V_1(1)$ and $V_2(1)$ have positive links to both hidden units. When $V_1(1)$ is clamped on, it therefore tends to switch both hidden units on. When H(1) is on it tends to switch on $V_2(1)$ and $V_2(3)$. If H(2) is switched on it also tends to switch on $V_2(1)$, but because it has a negative link to $V_2(3)$ it counteracts the effect of H(1) on $V_2(3)$. The only unit in V_2 which is switched on by both hidden units being on is $V_2(1)$. Therefore the net result is that when $V_1(1)$ is on it is most probable that $V_2(1)$ is switched on too. The same reasoning holds for other pairs.

It may happen that an apparent solution is discovered after only a few learning cycles. If this pattern of link strengths occured by chance and the machine has not truly modelled its environment, the solution is unstable and a different pattern is seen after a few more cycles. In the present calculations a problem is considered solved on cycle N if the same solution has been found for 10 consecutive cycles. In our experience it was rare for a Machine to lose the solution after this condition is satisfied.

Hinton et al. [Hinton,1984, Ackley, 1985] did not specify their criterion for considering a problem solved, but they quote a median learning time of 110 cycles after 250 experiments. In ten runs of a 4-2-4 encoder program written by R.K. Moore, we obtained a median value of N=114 learning cycles. Although ten runs are not sufficient to get quantitative statistics, we can claim to reproduce Hinton's results in essence.

5.2 UNIT ON-OFF REPRESENTATIONS

The choice of the unit on-off representation, that is the pair of numbers which represent the possible states of a unit, does not just alter parts of the Metropolis algorithm and the way that $\{p_{ij}\}$ and $\{p_{ij}^{f}\}$ are collected. It also affects the logic of the learning algorithm. When a Boltzmann Machine is started, all link strengths are zero. Hence by eqn(1) all possible states have zero energy. As learning progresses the energy of favourable states is lowered relative to unfavourable states by changes in the link strengths. Learning in a Boltzmann Machine therefore consists of lowering the energy of favourable global states. The means of doing this occurs at another level of the algorithm, the energy minimisations by simulated annealing, where the goal is also to find low energy states; this time for fixed values of the link strengths. Because the states of units represent truth values of hypotheses about the Machine's environment, there are two driving forces which may lower the energy. If a pair of units are joined by a positive link, it should be energetically favourable for them to be in the same state. However, it should also be energetically favourable for a pair of units to adopt opposite states if they are joined by a negative link. In the original formulation [Hinton, 1984; Ackley,1985] s_i=1 if unit i is on and 0 if it is off. With this choice eqn(1) shows that a negative contribution to E is found only if units i and j are both on and the link is positive. There is no way in which the interaction between units joined by negative links can lower the energy. However, both desiderata can be satisfied by choosing $s^{on}=+1$, $s^{off}=-1$.

The sign of $E_{ij} = -w_{ij} s_i s_j$ is shown in table 1 for the representations ($s^{on}=1$, $s^{off}=0$) and ($s^{on}=1$, $s^{off}=-1$).

Table 1: Sign of E_{ij}

		s=1,0	s=1,-1
$w_{ij}>0$	$s_i = s_j$ = on	-	-
	$s_i = s_j$ = off	0	-
	$s_i \neq s_j$	0	+
$w_{ij}<0$	$s_i = s_j$ = on	+	+
	$s_i = s_j$ = off	0	+
	$s_i \neq s_j$	0	-

A very different behaviour is evident in the two representations. With the (1,0) representation most contributions neither raise nor lower the energy. The only interactions with non-zero E_{ij} are those where a pair of units are both on, and there is no way of lowering E where links are negative. In contrast, in the (1,-1) case all contributions are either "good" or "bad" energetically; they are never neutral (unless $w_{ij}=0$). Lower energies may be obtained under the right conditions for positive and negative links. One would expect the (1,-1) representation to be a better choice because it shows the correct logic and every term has some effect.

Some experiments were carried out with (1,-1) but we postpone a discussion of the numerical results until Section 5.4 because they are dependent on the temperatures used in the annealing schedule.

5.3 LENGTH OF THE ANNEALING SCHEDULE

It was noted in section 2 that the pair statistics $\{p_{ij}\}$ and $\{p^f_{ij}\}$ must be collected after the machine has reached thermal equilibrium, by which one means d<E>/dt = 0 where <E> is the average energy. In [Hinton, 1984; Ackley,1985] the annealing schedule allowed 80 trial moves per unit (one trial move is one opportunity for a unit to change its state). After this it was assumed that the network had reached equilibrium. It is not obvious that this assumption is justified. The Metropolis method was borrowed from condensed matter physics where typical simulations involve several hundred units (=atoms or molecules). In that field it is usual to allow at least 1000 trial moves per unit before collecting statistics. Shorter times do not ensure that equilibrium is reached. The obvious way to test whether equilibrium is reached in a Boltzmann Machine is to extend the number of timesteps over which statistics are collected at the end of annealing and calculate <E> as a function of t. However, since in the 4-2-4 encoder there are only 2^{10} possible states for a given set of link strengths, it is possible to calculate the exact value of <E> directly. This is then the standard for testing the approximate <E> obtained from the Metropolis algorithm. The mean energy is given by:

$$\langle E\rangle = \sum_{i=1}^{1024} p_i E_i \qquad (8)$$

where p_i is the probability of being in state i of energy E_i:

$$p_i = \exp(-E_i/kT)/Q \qquad (9).$$

and Q is the partition function:

$$Q = \sum_{i=1}^{1024} \exp(-E_i/kT) \qquad (10)$$

which will be discussed in more detail later. So

$$<E> = (1/Q) \sum_{i=1}^{1024} E_i \exp(-E_i/kT) \qquad (11)$$

gives the exact value of <E> in a form which is easy to compute. Similarly

$$<E^2> = (1/Q) \sum_{i=1}^{1024} E_i^2 \exp(-E_i/kT) \qquad (12).$$

Hence it is simple to calculate

$$\mathrm{var}(E) = <E^2> - <E>^2 \qquad (13)$$

which is related to the specific heat (C) of a physical system:

$$C = \mathrm{var}(E)/kT^2 \qquad (14)$$

Table 2 shows a typical comparison of <E> obtained by averaging after simulated annealing using the annealing schedule of [Hinton, 1984; Ackley, 1985] and the exact sum-over-states (SOS) results from eqns (11) and (12). The Boltzmann Machine average was taken over the length of time used to collect $\{p_{ij}^{f}\}$, 10 units of time, at the lowest annealing temperature. The results are for the free-running mode where all ten units are free to flip, and were obtained with a typical set of link strengths which are a solution to the 4-2-4 problem.

Table 2. Energy results for a typical Boltzmann Machine

	BM algorithm	Exact SOS
<E>	22.0	18.9
var(E)	195.6	210.9

It is clear that the annealing schedule is long enough for the system to reach equilibrium, but only because the fluctuations are so large in this small system. Boltzmann Machines capable of useful applications will have hundreds or thousands of units and the ratio of var(E)/<E> will be smaller. It will then be necessary to allow much longer times for equilibration as is the case in the Monte Carlo calculations on solids and liquids. There is already some evidence in Hinton's results that larger numbers of units require longer annealing schedules. In experiments on a 40-10-40 encoder they found that, "To achieve good performance on the completion tests, 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." [Hinton, 1984]. This is actually more serious than a factor of two since the unit of time is proportional to the number of free units. In the operational (testing for completion mode) there are 6 free units in the 4-2-4 case, but 50 in the 40-10-40 encoder.

5.4 EFFECT OF TEMPERATURE ON THE LEARNING RATE

In [Hinton,1984; Ackley,1985] the annealing schedule adopted for the 4-2-4 encoder is given without any comment about how the temperatures were chosen. However, in view of the results in the previous section, it is necessary to ask how the learning time depends on temperature.

The annealing schedule in [Hinton,1984; Ackley,1985] is a vector T=(20,20,15,15,12,12,10,10,10,10) where the i'th element is the temperature at the i'th unit of time. The effect of temperature on learning time was investigated by running ten Boltzmann Machines for each schedule T'=AT, where A is a scaling factor. The results for unit representation (1,0) are shown in table 3. They are arranged in order of increasing N, the number of learning cycles needed for convergence. Within each set of ten Machines, the only difference is the seed for the random number generator. The same seeds were used for each set.

Table 3: Learning time for (1,0) representation

A	0.5	0.75	1.0	1.25	1.5
	41	51	72	61	49
	47	58	76	61	53
	68	71	78	83	61
	91	73	87	92	85
	106	87	110	95	88
	112	107	117	121	89
	157	238	119	>400	131
	207	287	194	>400	183
	380	349	232	>400	247
	>400	>400	>400	>400	>400
median	109	97	114	108	89

Many more calculations are needed before quantitative conclusions can be drawn, but it does appear that the learning time is not very sensitive to the annealing temperatures for the (1,0) case.

The change in energy when unit k is flipped is:

$$\Delta E_k = E_k^{off} - E_k^{on} = (s^{on} - s^{off}) \sum_{i=1}^{n} w_{ki} s_i \qquad (15).$$

For unit representation (1,0) this reduces to Hinton's eqn(4). However, for unit representation (1,-1) the average ΔE is twice as large. To make a comparison with the (1,0) results the base vector T=(40,40,30,30,24,24,20,20,20,20) was used for the (1,-1) case, which ensures that roughly the same number of Metropolis moves are accepted. The results are shown in table 4.

Table 4: Learning time for (1,-1) representation

A	0.5	0.75	1.0	1.25	1.5
	>400	39	43	57	33
	>400	68	105	83	55
	>400	69	117	83	91
	>400	109	123	85	229
	>400	147	126	87	>400
	>400	151	131	178	>400
	>400	186	154	>400	>400
	>400	199	263	>400	>400
	>400	322	>400	>400	>400
	>400	>400	>400	>400	>400
median	>400	149	131	87	>400

It appears from these preliminary results that the learning time is much more temperature dependent than in the (1,0) representation. Why this occurs has not yet been established.

Clearly, more calculations are necessary, but on the evidence of tables 3 and 4 the (1,-1) representation does not lead to faster learning times in practice. If this is true when more results are available, the reason is not obvious. One factor may be that the (1,-1) representation leads to "spikier" energy surfaces which may be more difficult to search. We have observed that the spread of energies typically obtained in Boltzmann Machines, i.e. the difference between the highest and lowest energies of the 1024 states, is much larger for the (1,-1) than for the (1,0) representation. This suggests that the barrier heights between minima may be greater. A second possible factor is that the different symmetry of the energy expression under different representations leads to extra degeneracies in the (1,-1) case. For example, the energy is the same when all units are on as when all units are off in the (1,-1) representation. This is not true for (1,0) except in the trivial case where all w_{ij} are zero.

The way in which temperature influences the learning rate is by altering the number of energy states which are thermally accessible at any stage of the learning algorithm. The partition function Q is a direct measure of the number of states accessible. Consider the limit of Q from eqn (10) where all energies are relative to the energy of the ground state. At low T the exponential goes to zero unless $E_i=E_{min}$, and the limiting value of Q is the degeneracy of the ground state. At high temperatures $\exp(-E_i/T)\rightarrow 1$ for all E_i and the limit of Q is the number of states in the system.

Let us focus on a fixed temperature: the lowest temperature, T_{min}, of the annealing schedule. When a Boltzmann Machine is started all states have the same energy and Q=1024 at all temperatures including T_{min}. As learning progresses and the link strengths are altered, the number of states accessible at T_{min} decreases to some value >1. The situation is similar in some ways to an energy minimisation by simulated annealing where the energy surface is fixed and the number of states accessible is decreased by lowering the temperature. It is natural to ask how Q varies during learning and whether many Boltzmann Machines all tend to reach similar Q values when they have solved the 4-2-4 problem.

To answer these questions Q was calculated from eqn(10) (at T_{min}) at each learning cycle for some typical Boltzmann Machines. Because of the small system size and the rather large step size in the link strengths there are quite large fluctuations in Q between learning cycles. However, some clear trends do emerge. Q drops rapidly over the first few cycles: for all machines the latest cycle on which 30 or more states are accessible at T is cycle 30. All machines rapidly settle down to Q<20, but the fine tuning may be very slow. Q then drops to final values in the range 5-15 when a stable solution is found. This behaviour is reminiscent of Hinton's description of the three stages of learning [Hinton,1984; Ackley,1985] and it is tempting to relate Hinton's stages to particular Q regimes. If, however, there are any quantitative correlations, they are hidden by the large fluctuations. The results are summarised in table 5.

Table 5. Variation of the partition function with learning cycle. N is the cycle on which a stable solution has been found for ten consecutive cycles. Under "last 30" the latest cycle on which Q is 30 or greater is shown; similarly "last 20" gives the latest cycle when Q is >20. Q_N is the value of Q on cycle N.

Run	last 30	last 20	N	Q_N
1	21	44	110	10.5
2	21	39	72	9.4
3	29	42	194	4.4
4	24	55	119	10.1
5	30	40	76	7.5
6	20	60	239	7.6
7	28	51	87	12.0
8	24	50	78	10.3
9	20	54	117	6.3
10	22	42	509	10.4

5.5 LINK STRENGTHS AS CORRELATION COEFFICIENTS

In Hinton's experiments on a 4-2-4 encoder the link strengths could take any even integer value. While this is convenient for fast computation, if the states of units represent truth values of hypotheses it seems logical to regard link strengths as correlation coefficients, i.e. real numbers in the interval $-1 \leq w_{ij} \leq 1$. The speed of integer arithmetic need not be lost if calculations are carried out with an integer cut-off, w_{cut}. This is equivalent to allowing only discrete real values in $-1 \leq w_{ij} \leq 1$. Several schemes can be envisaged. One simple method is to change the link strength w_{ij} by adding or subtracting (depending on the sign of $(p_{ij} - p^{f}_{ij})$) a constant only if the resulting value w'_{ij} would be within $-w_{cut} \leq w'_{ij} \leq w_{cut}$. This is the Hinton method if w_{cut} is infinite. A further advantage of imposing a cut-off is that patterns in link strengths developed early in the learning sequence have less influence as time progresses - the system is more sensitive to more recent experience. This method is being investigated.

6. RELATION OF BOLTZMANN MACHINES TO SPIN-GLASSES

Because Boltzmann Machines are a very recent development, few numerical results are available. It will now be shown that there are close similarities between Boltzmann Machines and spin-glasses. The latter have been the subject of numerous computer simulations since the mid-1970's and many of the results should carry over to Boltzmann Machines.

A spin-glass is simple in concept. Consider a lattice of magnetic atoms. At high temperatures the thermal energy is sufficient to overcome the interactions between the magnetic moments which tend to make the spins align. The spins are therefore able to rotate; there is no net magnetic moment, and the system is paramagnetic. Below the Curie temperature the thermal energy is not large enough to overcome the magnetic interactions: the system then adopts an ordered ferromagnetic phase with a net magnetic moment. However, if the magnetic interactions are weakened by replacing most of the magnetic atoms by atoms of a non-magnetic metal, a different behaviour occurs. At low temperatures the thermal energy is not sufficient to prevent the spins freezing into particular orientations, but the magnetic interactions are too weak to force the long-range order of a ferromagnet. This frozen disordered state is believed to be the structure of the spin-glass phase.

Despite this simple picture, the theoretical treatment of spin-glasses is a very active field of research where many of the major issues are only just becoming tractable. One of the most extensively studied models is that of Sherrington and Kirkpatrick [Kirkpatrick, 1978] for which the energy of a system of n spins, in the absence of an external field, is given by:

$$E = - \sum_{i=1}^{n} \sum_{j=i+1}^{n} J_{ij} s_i s_j \qquad (16)$$

where the spin variables s_i take values ±1, and the interactions J_{ij} are random variables taken from a Gaussian distribution. The energy expression for Boltzmann Machines, eqn (1), differs formally from eqn (16) only by the inclusion of self-terms. However, there are two further differences in the numerical calculations: most Boltzmann Machines adopt the unit on-off representation (1,0), and the link strengths are not random variables. The question therefore arises: do Boltzmann Machines exhibit spin-glass properties?

Most spin-glass calculations have been performed with hundreds of spins, in contrast to the ten units of a 4-2-4 encoder. However, Young and Kirkpatrick have calculated the exact statistical mechanical behaviour for systems with up to 20 spins [Young, 1982]. A principal conclusion from their study is that when a spin-glass is thermally excited out of its ground state, clusters of spins are flipped. We have observed similar low-energy cluster excitations in Boltzmann Machines, and this is direct evidence that Boltzmann Machines do have similar energy surfaces to those of spin-glasses. As an example, table 6 shows the states of units in the lowest energy and first excited states of a typical solution to the 4-2-4 encoder problem. In this example 5 units must be flipped to reach the first excited state.

Table 6. States of units in the ground state and first excited state of a typical Boltzmann Machine. Unit on-off representation (1,0).

Unit number	1	2	3	4	5	6	7	8	9	10
ground state	0	1	0	0	1	0	0	1	0	0
1st excited state	1	0	0	0	1	1	1	0	0	0

The number of spins which have to be flipped on average to go from the ground state to the first excited state of a spin-glass varies as $n^{1/2}$ [Young, 1982]. The n-dependence in Boltzmann Machines has not yet been investigated. However, if more extensive calculations show similar phenomena to those found in simulations on small spin-glass systems, it is reasonable to suppose that large Boltzmann Machines will also have energy surfaces like those of spin-glasses. This would have several consequences. First, it would mean that many of the technical details of the computations could be adapted to Boltzmann Machines. Second, and more important, it should give information about what energy surfaces are searchable by simulated annealing, and perhaps about the limitations of large Boltzmann Machines.

Consider a 1024-10-1024 encoder. Since $h=\log_2 v$, there are enough hidden units to permit each visible group to communicate its state to the other, i.e. a set of codes exist which solve the problem. Unfortunately, the existence of a solution is no guarantee that a Boltzmann Machine will find it since the algorithm is not an exhaustive search procedure. In fact the Boltzmann Machine can only work if the energy space is suitable for search by simulated annealing. There are two requirements for this to be so: that multiple low-energy minima exist and that the barriers between them are not insurmountable. If the energy space is like that of a spin-glass, then multiple low-energy minima do exist. However, the larger the system, the larger the barriers between the minima [Mackenzie, 1982]. This matter will not be pursued here; we merely note that there is a useful body of knowledge to be tapped.

7. CONCLUSIONS

The potential uses for any algorithm capable of getting knowledge into a computer are enormous. Although they have only been demonstrated for model problems which are simple in comparison with the tasks which a useful, practical system would have to perform, Boltzmann Machines do appear to be a significant step forward from earlier network models of learning. The key question is whether they can be scaled up to solve useful problems.

As with any other novel idea, it would be surprising if the initial formulation proves ultimately to be the best. However, before better systems can be devised, it is necessary to understand the original example thoroughly. The work reported here is a contribution towards that goal. We have shown that exact statistical mechanics make it possible to follow the progress of learning in Boltzmann Machines, and to investigate some parts of the algorithm quantitatively. We have also shown that Boltzmann Machines exhibit some of the characteristics of spin-glasses, and much useful information could be gained from this field of research.

Among the many potential applications, we have highlighted the way in which Boltzmann Machines might be used to learn the numerical measures of uncertainty which are used in Expert Systems. Since choosing these numbers is currently one of the major problems in building Expert Systems, Boltzmann Machines may make a big impact here. Other potential applications, such as those listed in the Introduction, are not considered less promising; they are merely outside the scope of this report.

REFERENCES

Ackley, D.H., Hinton, G.E., and Sejnowski, T.J., "A learning algorithm for Boltzmann Machines", Cognitive Science, 9 (1985) 147-168.

Aleksander, I., "Memory networks for practical vision systems", in "Physical and biological processing of images", Ed. Braddick, O.J., and Sleigh, A.C., Springer Series in Information Sciences, 11 (1983) 244-257.

Bridle, J.S., and Moore, R.K., "Boltzmann Machines for speech pattern processing", Proc. Inst. Acoust. Autumn Meeting, Lake Windermere, November 1984.

Clancy, W.H., and Shortliffe, E.H., "Readings in medical artificial intelligence: the first decade", Addison-Wesley, 1984.

Cohen, P.R., and Feigenbaum, E.A., "The handbook of artificial intelligence", Vol. 3, Pitman, London, 1982.

Davis, R., and Lenat, D.B., "Knowledge based systems in Artificial Intelligence", McGraw-Hill, 1982.

Derthick, M., "Variations on the Boltzmann Machine learning algorithm", Technical Report CMU-CS-84-120, Carnegie-Mellon University, August 1984.

Fahlman, S.E., Hinton, G.E., and Sejnowski, T.J., "Massively parallel architectures for AI: NETL, Thistle, and Boltzmann Machines", Proc. Nat. Conf. on Artificial Intelligence, AAAI, Washington, DC, 1983, 109-113.

Feigenbaum, E.A., and McCorduck, P., "The fifth generation", ,London, 1984.

Feldman, J.A., and Ballard, D.H., "Connectionist models and their properties", Cognitive Science, 6 (1982) 205-254.

Hinton, G.E., "Inferring the meaning of direct perception", The Behavioral and Brain Sciences, 3 (1980) 387-388.

Hinton, G.E., and Sejnowski, T.J., "Analysing cooperative computation", Proc. 5th Ann. Conf. Cognitive Science Soc., Rochester, NY, May 1983. (a)

Hinton, G.E., and Sejnowski, T.J., "Optimal perceptive inference", Proc. IEEE Computer Soc. on Computer Vision and Pattern Recognition, Washington, DC, June 1983, 448-453. (b)

Hinton, G.E., Sejnowski, T.J., and Ackley, D.H., "Boltzmann Machines: constraint satisfaction networks that learn", Technical Report CMU-CS-84-119, Carnegie-Mellon University, May 1984.

Hopfield, J.J., "Neural networks and physical systems with emergent collective computational abilities", Proc. Nat. Acad. Sci. USA, 79 (1982) 2554-2558.

Hopford, R.P., "The ASDIC project: lessons learned while setting up an IKBS research programme",Proc. 3rd. Seminar on the Application of Machine Intelligence to Defence Systems, RSRE, June 1984.

Kirkpatrick, S, and Sherrington, D., "Infinite-ranged models of spin-glasses", Phys. Rev., B17 (1978) 4384-4403.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimisation by simulated annealing", Science, 220 (1983) 671-680.

Kullback, S., "Information Theory and Statistics", Wiley, New York, 1959.

Lenat, D.B., "Computer software for intelligent systems", Scientific American, September, 1984, 152-160. (a)

Lenat, D.B., and Brown, J.S., "Why AM and EURISKO appear to work", Artficial Intelligence, 23, (1984) 269-294. (b)

Luttrell, S., RSRE Memorandum 3815.

Mackenzie, N.D., and Young, A.P., "Lack of ergodicity in the infinite-range Ising spin-glass", Phys. Rev. Lett., 49 (1982) 301-304.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., "Equation of state calculations by fast computing machines", J. Chem. Phys., 21 (1953) 1087-1092.

Minsky, M., and Papert, S., "Perceptrons", MIT Press, Cambridge, MA, 1968.

Pritchard, J.A.S., RSRE Memorandum 3788, November 1984.

Schlogl, F., "Produced entropy in quantum statistics", Z. Physik, 249 (1971) 1-11.

Shortliffe, E.H., "Computer based medical consultations: MYCIN", American Elsevier, New York, 1976.

Young, A.P., and Kirkpatrick, S., "Low-temperature behavior of the infinite-range Ising spin-glass: exact statistical mechanics for small samples", Phys. Rev., B25 (1982) 440-451.

REPORTS QUOTED ARE NOT NECESSARILY
AVAILABLE TO MEMBERS OF THE PUBLIC
OR TO COMMERCIAL ORGANISATIONS

UNLIMITED

DOCUMENT CONTROL SHEET

Overall security classification of sheet ....Unclassified........

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference Memorandum 3826	3. Agency Reference	4. Report Security Classification U/C

5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location Royal Signals and Radar Establishment
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title

Boltzmann Machines and Artificial Intelligence

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials Bounds D G	9(a) Author 2	9(b) Authors 3,4...	10. Date	pp. ref.

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement

Descriptors (or keywords)

continue on separate piece of paper

Abstract

Ackley, Hinton and Sejnowski have recently proposed an algorithm, named a Boltzmann Machine, which is capable of learning to recognise the underlying structure in a set of patterns presented to it. The main purposes of this memorandum are: to introduce Boltzmann Machines to those who are not familiar with them; to outline how Boltzmann Machines may prove useful in the knowledge acquisision problem in artificial intelligence; to report some new results for a model problem; and to sketch out the relationship between Boltzmann Machines and the spin-glass problem.

END

FILMED

12-85

DTIC