Theory of associative memory in randomly connected Boolean neural networks

1989 J. Phys. A: Math. Gen. 22 2233

(http://iopscience.iop.org/0305-4470/22/12/022)

J. Phys. A: Math. Gen. **22** (1989) 2233-2263. Printed in the UK

Theory of associative memory in randomly connected Boolean neural networks

K Y M Wong and D Sherrington

Department of Physics, Imperial College, University of London, London SW7 2BZ, UK

Received 10 February 1989

Abstract. The Aleksander model of neural networks replaces the connection weights of conventional models by logic devices (or Boolean functions). Learning is achieved by adjusting the Boolean functions stepwise via a 'training-with-noise' algorithm. We present a theory of the statistical dynamical properties of the randomly connected model and demonstrate that, in the limit of large but dilute connectivity c of the nodes, the storage capacity for associative memory is of the order $(2/c^2)2^c$, which corresponds, roughly speaking, to an average of one nearest-neighbouring pattern stored at site distances 2 on each node. Two parameters are introduced into the learning algorithm: q_r and q_c being respectively the probabilities to register a correct bit and erase an incorrect one. The effects of varying q_r, q_c and the training noise level on the storage capacity (after very long training) are discussed. In the limit of low training noise level, the training algorithm is equivalent to the so-called 'proximity rules'. Study of its retrieval properties shows that the model can be described as 'short ranged', whereas the Hopfield model is 'long ranged'. The advantages and disadvantages of introducing the intermediate u state into the system are also discussed.

1. Introduction

In the past few years, there has been an upsurge of interest in neural network models in both the fields of physics and of computer science. In the physics community, the introduction of the Hopfield-Little model [1, 2] has enabled statistical mechanical concepts to be applied to the system [3]. Since then, various statistical properties of the model have been extensively studied. These include equilibrium [4] and dynamical [5] properties, and optimal storage capacity [6]. In the computer science community, interest has been focused more on feedforward networks and Boltzmann machines [7]. An error propagation algorithm [8] has been proposed for learning in such systems, but simulations have shown that the required number of training steps for some 'hard-learning' problems is excessively large [9].

Recently, Aleksander [10] has proposed an alternative neural network model which exhibits remarkable performance for some 'hard-learning' problems. Conventional models such as Hopfield-Little [1, 2] learn by adjusting the connection weights between the nodes and retrieve by updating the states of the nodes according to the local field. The Aleksander model, on the other hand, replaces the connection weights by logic devices, or random access memories (RAM), i.e. the state of a node is updated according to a Boolean function of the states of those nodes feeding it. Learning is achieved by adjusting the Boolean functions stepwise via a 'training-with-noise' algorithm.

0305-4470/89/122233+31$02.50

Immediately, we see a further merit of the Aleksander model in its use of readily accessible RAM technology. Besides, since there are 2^c adjustable parameters for a RAM of connectivity c, it has a potential for large storage capacity. However, with a few exceptions [11, 12], Boolean neural networks have not been widely studied in physics, and their statistical properties are not yet known. In a previous letter [13], we presented a theory for the statistical properties of a randomly connected Boolean model for storing uncorrelated patterns, in the thermodynamic limit that the number of nodes N approaches infinity. There we explained the model briefly with the help of a few examples and, with some appropriate approximations, showed that the dynamics of the system is described by a recursion relation for the fractional Hamming distance between the state configuration of the nodes and one of the stored patterns, the quality of retrieval and the basin of attraction being determined by the fixed points of the relation. In the limit of large connectivity c of the nodes, we derived an expression for the storage capacity for associative memory.

In this paper, we study the Boolean model in more detail, and consider various factors affecting the storage capacity. An attempt to improve the performance of the network, originally proposed by Aleksander, is to allow the network to store, besides bits of 1 and 0, an intermediate undefined (or u) state, i.e. a state which outputs randomly 1 or 0 whenever it is involved in the network dynamics. We shall discuss the advantages and disadvantages of introducing the undefined state, and show that the overall improvement in the storage capacity is only marginal.

We shall also introduce two more parameters into the learning algorithm: the registration probability q_r and the correction probability q_c being respectively the probabilities to register a correct bit and erase an incorrect one. It turns out that the ratio q_c/q_r determines, after very long training, the fraction of bits in the u state, and hence the behaviour of the system.

Another feature of the Aleksander model is the use of noisy example patterns during training, so that noisy input patterns can be 'recognised' during retrieval, i.e. associative memory is possible. However, the training noise itself causes disruption of the stored information. We shall discuss the effects of varying q_c/q_r and the training noise level on the storage capacity (after very long training).

Some insights about the learning algorithm will be obtained by comparison with a class of learning rules called the proximity rules which, roughly speaking, allocate the stored bits by their 'proximity' to the bits of the patterns in the RAM space. This leads us to propose the majority rule, which gives a higher storage capacity than any noise-trained algorithm.

We shall also compare the retrieval properties of the Boolean and conventional models, and find that, in a sense to be discussed below, the Boolean model can be considered as 'short ranged', in contrast to the 'long ranged' models of Hopfield and Little.

The plan of the paper is as follows. In §§ 2-5, we illustrate the basic concepts by considering, in order, the behaviour of the network in the following cases: simple storage of one pattern (§ 2), storage of one pattern for associative memory (§ 3), storage of a pattern and its complement (§ 4), and storage of two correlated patterns (§ 5). This facilitates consideration of the more complicated case of storing p uncorrelated patterns in §§ 6-9. We derive the storage capacity for associative memory in § 6 and consider its dependence on various factors in § 7. In § 8, we compare the training-with-noise algorithm with the proximity rules, and in § 9, we compare the retrieval properties of the Boolean and conventional neural models. Section 10 is devoted to our conclusion.

2. Basic formulation and simple storage of one pattern

Consider a network of N *nodes* of states 1 or 0, each being fed by c others. Each node is a variable logic device whose state may be described by a complete truth table. In other words, if the ith node is fed by the i_1th, ... i_cth nodes, then its output V_i may be 1 or 0 depending on the *input sequence* $(V_{i_1}, \ldots, V_{i_c})$. The ith node is therefore specified by a Boolean function F_i whose domain consists of 2^c elements mapping onto the values 1 or 0. See figure 1.

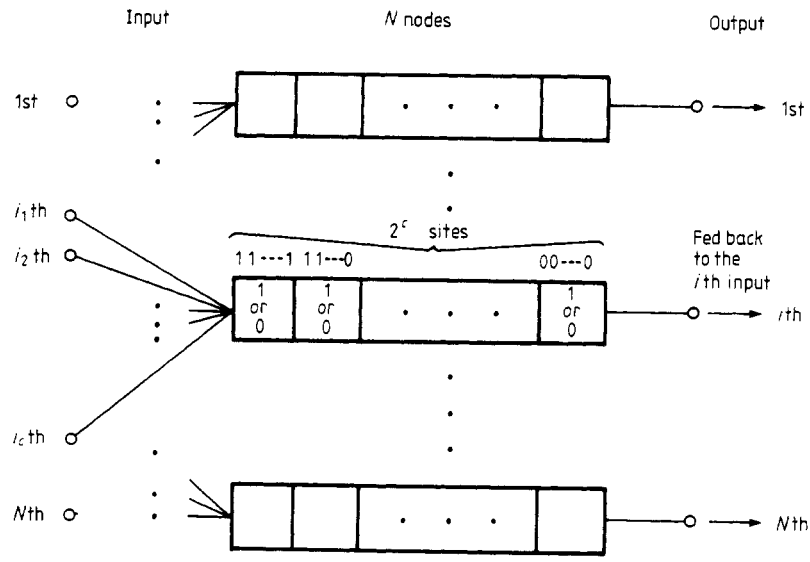

Figure 1. The Boolean neural network.

Two possible kinds of dynamics can be used to determine the time evolution of the system. For synchronous (parallel) dynamics, the output of all nodes at time $t+1$ are updated simultaneously according to the corresponding Boolean functions of the input sequence at time t. Therefore, the equation of motion is:

$$V_i(t+1) = F_i(V_{i_1}(t), \ldots, V_{i_c}(t)) \qquad i = 1, \ldots, N. \tag{2.1}$$

For asynchronous dynamics, each node is updated with a probability τ^{-1} per unit time, and the updated output is determined by the *instantaneous* value of the Boolean function.

We shall be concerned with the situation where the connectivity of the nodes is random, but quenched. The network therefore has the same architecture as the so-called Kauffman model [14, 15].

Alternatively, we can treat each node as the output of a set of 2^c *sites* (or 'pigeon holes'), the *address* of each being a c-bit word of 1s and 0s. The output corresponding to the input sequence $(V_{i_1}, \ldots, V_{i_c})$ is stored in the site whose address is identical to the sequence. An input sequence will therefore yield as output the data stored in the site with the identical address. We shall frequently refer to this pigeon-hole picture in subsequent discussions, as it generates valuable insight.

Memory is stored in the Boolean function F_i. In the later sections we shall discuss several methods of training the F_i to learn patterns. Here we note only one simple condition for ensuring that a single pattern is persistently maintained; this is to store the bits of the pattern at the sites which are addressed by the pattern itself. In other words, a pattern $\{\zeta_i;\ i=1,\dots N\}$ is maintained if once initiated by the allocation

$$F_i(\zeta_{i_1},\dots\zeta_{i_c})=\zeta_i \qquad i=1,\dots N. \tag{2.2}$$

Retrieval is concerned with the consequences of presenting a possibly noisy pattern to the network and studying the evolution of the node states. During retrieval, the network dynamics can be studied by monitoring the time evolution of the *distances* between the state configuration of the nodes and those of the stored patterns; the distance between two configurations is the fraction of different bit-entries. In general, this is not sufficient, for we have to take into account the detailed correlation of the output configuration with the Boolean functions F_i and their input configurations. However, it has been proposed [16] that correlation effects are negligible for $c \ll \ln N$. This is the so-called *annealed approximation* [15]. We shall employ it below.

To ensure an unbiased initial state, and to allow for a more flexible response, Aleksander modified the above two-state system into a three-state one by introducing an *undefined* (or u) state, i.e. a site in u state has an equal probability of outputting a 1 or 0 whenever it is addressed. As we shall see, the statistical properties of the three-state system are much more intricate, and their generalisation to the two-state system is rather straightforward. We shall therefore concentrate our discussion on the three-state system, generalising to the two-state one where appropriate. A further comparison of the two systems will be published elsewhere [17].

In the Aleksander model, the sites of all nodes are initialised to u. Let us first consider the simplest case of storing only one pattern in the network with the allocation (2.2), but otherwise with all sites remaining in the u state. Within the annealed approximation, the average distance x between the state configuration of the nodes and the stored pattern satisfies the recursion relation [15, 16]:

$$x(t+1)=f(x(t)) \qquad \text{for parallel dynamics} \tag{2.3}$$

or

$$\tau\,\mathrm{d}x(t)/\mathrm{d}t=f(x(t))-x(t) \qquad \text{for asynchronous dynamics} \tag{2.4}$$

where

$$f(x)=\tfrac{1}{2}[1-(1-x)^c]. \tag{2.5}$$

This equation can easily be derived: the distance x gives the probability that an input bit is in error when compared with the stored pattern. The site storing the correct output bit is addressed only if all the input bits to a node are correct, this happening with a probability of $(1-x)^c$. Otherwise, with probability $[1-(1-x)^c]$, a site in the u state is addressed, outputting the correct bit with a probability of $\frac{1}{2}$. Hence the expression of $f(x)$ in (2.5), which is the same as that obtained by Derrida and Pomeau [15] for the evolution of distance between two configurations for an annealed Kauffman net. We note that $f(x)$ has the same form for a two-state system within the annealed approximation, if the sites are randomly initialised and only the allocation of (2.2) imposed.

$f(x)$ is called the *retrieval function*, and the curve of $f(x)$ against x is called the *retrieval curve*. As the system evolves, $x(t)$ approaches one of the stable fixed points x^* of the curve, i.e. $f(x^*)=x^*$. Graphically, the fixed points are given by the intersection

of the retrieval curve and the line $y = x$. The stable fixed points are those with slope $|f'(x^*)| < 1$. A stable fixed point at or near $x = 0$ means that the network has associative memory, i.e. starting with an initial configuration which has only partial agreement with the stored pattern, it eventually retrieves the pattern. Unfortunately, $x = 0$ is an *unstable* fixed point of the retrieval curve for $c > 2$. For large c, the stable fixed point x^* approaches $\frac{1}{2}$ and even for $c = 3$, $x^* = 0.382$. In other words, for $c > 2$, the network has at best very poor associative memory if the pattern is *only* stored at the correct sites. (Note that here and below we consider only the average evolution in the annealed approximation. In a quenched network storing a single pattern, the u states will ensure eventual retrieval in $O(2^N)$ steps, when the probability that an evolving configuration falls into the 'trap' of the stored pattern becomes significant, but this is irrelevant in the thermodynamic limit.)

3. Training with noise and storage of one pattern for associative memory

To ensure associative memory of a single pattern, an obvious but uninteresting method is to store the correct bit in every site of a node. The information storage in this way, however, is excessive. By introducing the notion of training with noise [10, 18], it is sufficient to store the correct bit in a small fraction of the sites. Let us start with all sites in the u state. During a training step, a slightly noisy version of the pattern to be stored is presented to the input of the nodes, and the correct bit of the pattern is stored in the addressed site. (If a correct bit is already stored in the site it is, of course, unaltered.) Finally, after s noisy training steps, the system is also trained with the noiseless pattern. Now, if d is the probability that a bit of such an *example pattern* is in error, the recursion relation (2.5) has to be replaced by:

$$f(x) = \tfrac{1}{2} \sum_{r=1}^{c} \binom{c}{r} [1 - d^r (1-d)^{c-r}]^s x^r (1-x)^{c-r}. \tag{3.1}$$

Again, this equation can easily be derived: consider a site at a *site distance* r from the pattern, i.e. its address is r bits different from the input sequence of the (correct) pattern. The probability that this site is addressed in a training step is $d^r(1-d)^{c-r}$. The probability that this site is still unaddressed after s training steps, and hence is still in the u state, is $[1-d^r(1-d)^{c-r}]^s$. Such a site has a probability $\frac{1}{2}$ of giving an incorrect bit during retrieval. Now suppose that, during retrieval, the input configuration has a fractional error x when compared with the correct pattern. The probability that the considered site is being addressed is therefore $x^r(1-x)^{c-r}$. Summing over all the possible sites, we obtain (3.1). Since the network is also trained with the noiseless pattern after the training steps, the $r = 0$ term vanishes. (Again, we note that the retrieval function for the two-state system is the same within the annealed approximation.)

As shown in figure 2, the fixed point at $x = 0$ changes from unstable to stable after a number of training steps and associative memory becomes possible. The number of training steps to achieve this is

$$s = -\frac{\ln(c/2)}{\ln[1 - d(1-d)^{c-1}]} \tag{3.2}$$

or, strictly, the next greatest integer. The optimal noise to minimise the number of

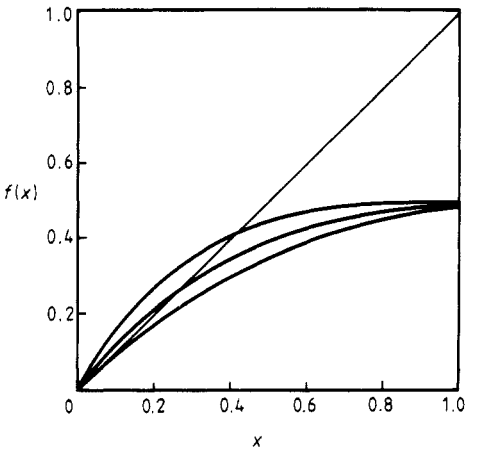

Figure 2. The retrieval curve of storing one pattern by 'training with noise' for $c = 4$, $d = \frac{1}{4}$ and, from top to bottom, $s = 1$, 4 and 7.

steps is

$$d = 1/c. \tag{3.3}$$

From the slope of the retrieval function near $x = 0$, we see that associative memory depends on the extent to which the correct bit is stored in sites at site distances 1 from a pattern. With $d = 1/c$, these sites are most frequently addressed, thus minimising the number of training steps.

The optimal number of steps can be as small as three (for $c = 3$), while for $c \gg 1$ it approaches $2.718c \ln(c/2)$, which is a surprisingly small fraction of 2^c, the total number of sites per node. Hence, the number of training steps to achieve associative memory is encouragingly small, implying also a small spread of the stored information on each node and hence offering the potential for storage of many patterns.

4. The training algorithm and the storage of complementary patterns

For the storage of more than one pattern, it is inevitable that conflicting bits may have to be stored at the same site. The following generalised *training-with-noise* algorithm is therefore introduced.

(*1*) *All the sites are initialised to u states.*

(*2*) *During each training step, an example pattern is presented to the input of the nodes*:

(*a*) *if the correct bit is already stored in an addressed site, it is left unaltered*;

(*b*) *if an addressed site is in the u state, the correct bit of the example pattern is stored at the site with a probability* q_r (*or the registration probability*);

(*c*) *if the incorrect bit is already stored in an addressed site, it is changed to a u state with a probability* q_c (*or the correction probability*).

In Aleksander's original algorithm, $q_c = q_r = 1$.

It is illuminating to consider training the network with a pattern $\{\zeta_i\}$ and its complement $\{\bar{\zeta}_i\}$ (i.e. $\bar{\zeta}_i = 1 - \zeta_i$). Consider a site on a particular node at a site distance r from that accessed by pattern $\{\zeta_i\}$ (and hence at a site distance $c - r$ from $\{\bar{\zeta}_i\}$). The

site may be in the u state, or store a bit from pattern $\{\zeta_i\}$, or from its complement. Let these probabilities be P_0, P_+ and P_- respectively, satisfying

$$P_+ + P_0 + P_- = 1. \tag{4.1}$$

These probabilities depend on the sequence of training steps. Thus if the pattern $\{\zeta_i\}$ is presented as the sth training example, then the probabilities before and after the step are related by the matrix

$$\begin{pmatrix} P_+(s) \\ P_0(s) \\ P_-(s) \end{pmatrix} = \begin{pmatrix} 1 & q_r\eta_r & 0 \\ 0 & 1-q_r\eta_r & q_c\eta_r \\ 0 & 0 & 1-q_c\eta_r \end{pmatrix} \begin{pmatrix} P_+(s-1) \\ P_0(s-1) \\ P_-(s-1) \end{pmatrix} \tag{4.2}$$

where $\eta_r = d^r(1-d)^{c-r}$ is the probability that the site is addressed by the input sequence of the example pattern. This equation is a direct consequence of the training-with-noise algorithm. The 3×3 matrix relating to the two probability distributions is called the *training matrix* of the pattern $\{\zeta_i\}$. The training matrices $\mathbf{T}_+$ and $\mathbf{T}_-$ of the patterns $\{\zeta_i\}$ and $\{\bar{\zeta}_i\}$ respectively are therefore

$$\mathbf{T}_+ = \begin{pmatrix} 1 & q_r\eta_r & 0 \\ 0 & 1-q_r\eta_r & q_c\eta_r \\ 0 & 0 & 1-q_c\eta_r \end{pmatrix} \qquad \mathbf{T}_- = \begin{pmatrix} 1-q_c\eta_{c-r} & 0 & 0 \\ q_c\eta_{c-r} & 1-q_r\eta_{c-r} & 0 \\ 0 & q_r\eta_{c-r} & 1 \end{pmatrix}. \tag{4.3}$$

Starting from an initial probability distribution $\boldsymbol{P}(0) = (P_+(0), P_0(0), P_-(0))^T$, the probability distribution $\boldsymbol{P}(s)$ after the training sequence $T_{i(1)}, T_{i(2)}, \ldots, T_{i(s)}$ is given by

$$\boldsymbol{P}(s) = T_{i(s)} \ldots T_{i(2)} T_{i(1)} \boldsymbol{P}(0). \tag{4.4}$$

Averaging over all the possible training sequences, we have

$$\langle \boldsymbol{P}(s) \rangle = [(\mathbf{T}_+ + \mathbf{T}_-)/2]^s \boldsymbol{P}(0). \tag{4.5}$$

The behaviour of the system can best be analysed in the limit $s \to \infty$ when $\boldsymbol{P}(s)$ should approach an *equilibrium distribution* $\boldsymbol{P}^*$. In this limit, we expect that $\boldsymbol{P}^*$ should satisfy

$$\boldsymbol{P}^* = [(\mathbf{T}_+ + \mathbf{T}_-)/2]\boldsymbol{P}^*. \tag{4.6}$$

Because of the normalisation condition (4.1) of $\boldsymbol{P}$, the averaged training matrix $(\mathbf{T}_+ + \mathbf{T}_-)/2$ always has an eigenvalue equal to 1, and the corresponding eigenvector gives the equilibrium distribution

$$\boldsymbol{P}^* = \frac{1}{\eta_r^2 + (q_c/q_r)\eta_c\eta_{c-r} + \eta_{c-r}^2} \begin{pmatrix} \eta_r^2 \\ (q_c/q_r)\eta_r\eta_{c-r} \\ \eta_{c-r}^2 \end{pmatrix}. \tag{4.7}$$

It should, however, be noticed that the equilibrium is a dynamic one, in the sense that an individual site may flip between 1, u and 0 from one training step to another, although the *overall* distribution is unchanged.

The *disruption*, i.e. the probability that the site outputs an incorrect bit with respect to the pattern $\{\zeta_i\}$, is therefore

$$a_r = \tfrac{1}{2}(1 - m_r) \tag{4.8}$$

where

$$m_r = \frac{\eta_r^2 - \eta_{c-r}^2}{\eta_r^2 + (q_c/q_r)\eta_r\eta_{c-r} + \eta_{c-r}^2}. \tag{4.9}$$

We shall refer to m_r as the *trained polarisation* at a site distance r from the pattern $\{\zeta_i\}$. For later reference, we also write down the probability u_r of a site in the u state at a site distance r,

$$u_r = \frac{(q_c/q_r)\eta_r\eta_{c-r}}{\eta_r^2 + (q_c/q_r)\eta_r\eta_{c-r} + \eta_{c-r}^2}. \tag{4.10}$$

It is convenient to simplify the above expressions by introducing the concept of a *training-noise temperature*. Using analogies from thermodynamics, we define the noise temperature $T = \beta^{-1}$ by the ratio equation $d:(1-d) = e^{-\beta}:1$. In other words, the probabilities of a bit being correct or incorrect in an example pattern are respectively $(1+e^{\mp\beta})^{-1}$. This simplifies (4.9) and (4.10) into

$$m_r = E_h \frac{e^{\beta(h-r)} - e^{-\beta(h+c-r)}}{e^{\beta(h-r)} + e^{-\beta(h+c-r)}} \tag{4.11}$$

$$u_r = \coth(2\beta h) O_h \frac{e^{\beta(h-r)} - e^{-\beta(h+c-r)}}{e^{\beta(h-r)} + e^{-\beta(h+c-r)}} \tag{4.12}$$

where

$$\beta h = \tfrac{1}{2}\cosh^{-1}(q_c/2q_r) \tag{4.13}$$

and E_h and O_h are even and odd operators defined by

$$E_h(f(h)) = \tfrac{1}{2}(f(h) + f(-h))$$

$$O_h(f(h)) = \tfrac{1}{2}(f(h) - f(-h))$$

for any function $f(h)$. Clearly, m_r corresponds to the magnetisation (polarisation) of an Ising spin in a field $(h - r + c/2)$.

Before we proceed, it is legitimate to question whether the above expressions of disruptions, averaged over all training sequences, are valid for a particular training sequence chosen at random, as intuition tells us that the last few example patterns affect the system most. In our subsequent discussions, various quantities like the retrieval error and the storage capacity will be derived. In general, they involve non-linear expressions of the disruptions. It is therefore invalid to perform the average over training sequences *before* deriving these quantities. Instead, a distribution of these quantities should be considered for an ensemble of training sequences.

To consider an exactly solvable case, we shall henceforth modify the learning algorithm to an *averaged algorithm*, which states the following.

During each training step, examples of all patterns to be stored are presented to the input. For each node, one of the patterns is chosen at random, and the information of that pattern is stored at the site addressed by its own input sequence, according to the previous algorithm.

The validity of the above equations for this algorithm is obvious, since the corresponding training matrix in each step is the average of those of all patterns.

The retrieval curve can now be derived easily. Let x be the distance between the state configuration of the nodes and the pattern $\{\zeta_i\}$. The retrieval function after very long training is then given by

$$f_x(x) = \sum_{r=1}^{c-1} \binom{c}{r} a_r x^r (1-x)^{c-r} + x^c. \tag{4.14}$$

Again, to ensure better performance, we have, after the training procedure, also stored the correct bits of both patterns at the appropriate sites. Thus the $r=0$ term is dropped and the $r=c$ term is replaced by x^c.

To demonstrate the effects of training, we also write down for $q_c = q_r = 1$ the retrieval function $f_s(x)$ after s training steps,

$$f_s(x) = f_\infty(x) + \sum_{r=1}^{c-1} \binom{c}{r} \left(\sum_{\pm} [1 - \tfrac{1}{2}(\eta_r \pm \sqrt{\eta_r \eta_{c-r}} + \eta_{c-r})]^s \times \frac{\eta_r - \eta_{c-r}}{4(\eta_r \pm \sqrt{\eta_r \eta_{c-r}} + \eta_{c-r})} \right) x^r (1-x)^{c-r}. \tag{4.15}$$

As shown in figure 3, the fixed points of the retrieval curve at $x=0$ and 1 become stable after a number of training steps, showing the network is able to retrieve both patterns, and $x=\frac{1}{2}$ becomes unstable. The unstable fixed point can therefore be considered as the *basin boundary* separating the two basins of attraction, after sufficient training for retrieval.

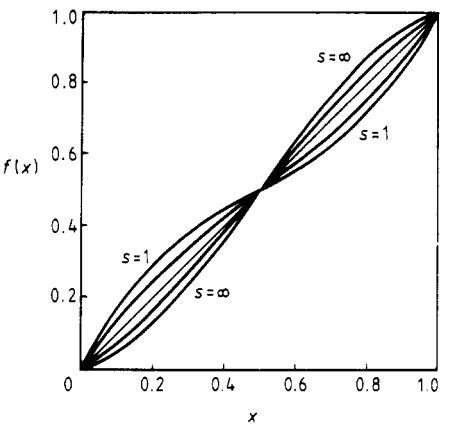

Figure 3. The retrieval curve of storing two complementary patterns for $c=4$, $d=\frac{1}{4}$, $q_c/q_r=1$ and for $s=1$, 7, 30, ∞ respectively. The curve switches from non-retrieving to retrieving at $s=16$.

We now consider how the above analysis can be adapted to the two-state system. Without the u state, the training-with-noise algorithm has to be modified. While a correct bit of an example pattern is still left unaltered in an addressed site, the incorrect bit is directly changed to the correct one with a probability q_c (without going through any intermediate state). The retrieval function in this case is different from that for the three-state system. The derivation, however, is very similar. The extent of training in various sites is now described by two probabilities P_+ and P_- instead of three. The averaged training matrix is therefore 2×2 and is given at a site distance r by

$$\bar{\mathbf{T}} = \begin{pmatrix} 1 - q_c \eta_{c-r} & q_c \eta_r \\ q_c \eta_{c-r} & 1 - q_c \eta_r \end{pmatrix}. \tag{4.16}$$

Starting from a random initial state, the retrieval function for $q_c = 1$ after s training steps is given by

$$f_s(x) = f_\infty(x) + \sum_{r=1}^{c-1} \binom{c}{r} [1 - \tfrac{1}{2}(\eta_r + \eta_{c-r})]^s \frac{\eta_r - \eta_{c-r}}{2(\eta_r + \eta_{c-r})} x^r (1-x)^{c-r} \tag{4.17}$$

where

$$f_\infty(x) = \sum_{r=1}^{c-1} \binom{c}{r} \frac{\eta_{c-r}}{\eta_r + \eta_{c-r}} x^r (1-x)^{c-r} + x^c. \tag{4.18}$$

It is interesting to note that after very long training, the behaviour of the two-state system approaches that of the three-state one with $q_c/q_r = 2$.

5. Storage of two correlated patterns

Correlated patterns are most conveniently specified by replacing the Boolean states V_i of the nodes with Ising variables $S_i = \pm 1$, namely $S_i = 2V_i - 1$ for $V_i = 1, 0$. The overlap between patterns $\{S_i\} = \{\xi_i^1\}$ and $\{S_i\} = \{\xi_i^2\}$ is then defined as

$$\rho = \frac{1}{N} \sum_i \xi_i^1 \xi_i^2. \tag{5.1}$$

Hence $\rho = 0$ implies uncorrelated patterns, and $\rho = -1$ implies complementary patterns. In this section we consider the storage and retrieval of two correlated patterns.

Whereas in the case of two complementary patterns in which all nodes are trained to output opposite bits for the two patterns, for two correlated patterns of general overlap ρ, a fraction $(1+\rho)/2$ of the nodes are trained to output the same bits. The network dynamics therefore have to be specified by two variables. Let x and y be the fractional errors of the state configuration with respect to pattern 1 for those nodes outputting respectively the same and different bits for patterns 1 and 2. In other words, $P(S_i = -\xi_i^1 | \xi_i^1 = \xi_i^2) = x$ and $P(S_i = -\xi_i^1 | \xi_i^1 = -\xi_i^2) = y$. After very long training, we then have

$$\left.\begin{aligned} x(t+1) &= f(x(t), y(t)) \\ y(t+1) &= g(x(t), y(t)) \end{aligned}\right\} \text{for parallel dynamics} \tag{5.2}$$

or

$$\left.\begin{aligned} \tau\, \mathrm{d}x(t)/\mathrm{d}t &= f(x(t), y(t)) - x(t) \\ \tau\, \mathrm{d}y(t)/\mathrm{d}t &= g(x(t), y(t)) - y(t) \end{aligned}\right\} \text{for asynchronous dynamics} \tag{5.3}$$

where

$$f(x, y) = 0 \tag{5.4a}$$

$$g(x, y) = \sum_{r,s,t} \frac{c!}{s!(c-r-s)!\,t!(r-t)!} \left(\frac{1+\rho}{2}\right)^{c-r} \left(\frac{1-\rho}{2}\right)^{r} x^s (1-x)^{c-r-s} y^t (1-y)^{r-t}$$

$$\times \frac{1}{2}\left(1 - \frac{\eta_{s+t}^2 - \eta_{s+r-t}^2}{\eta_{s+t}^2 + (q_c/q_r)\eta_{s+t}\eta_{s+r-t} + \eta_{s+r-t}^2}\right). \tag{5.4b}$$

These expressions are easy to understand. For nodes with the same output bit, all sites are filled with their common bits after very long training. For nodes with opposite output bits for patterns 1 and 2, the expression in $g(x, y)$ corresponds to the fact that among the c inputs $\{j\}$ of a node, s are such that $-S_j = \xi_j^1 = \xi_j^2$, $c-r-s$ are such that $S_j = \xi_j^1 = \xi_j^2$, t are such that $-S_j = \xi_j^1 = -\xi_j^2$, and $r-t$ are such that $S_j = \xi_j^1 = -\xi_j^2$. The site addressed by this configuration is therefore at a site distance $s+t$ from pattern 1,

and $s+r-t$ from pattern 2. Since the trained polarisation can be shown to be independent of s, the relation can be further simplified to yield

$$f(x, y) \equiv f(x) = 0 \tag{5.5a}$$

$$g(x, y) \equiv g(y) = \sum_{r=0}^{c} \binom{c}{r} \left(\frac{1+\rho}{2}\right)^{c-r} \left(\frac{1-\rho}{2}\right)^{r} \times \sum_{t=0}^{r} \binom{r}{t} \frac{1}{2} \left(1 - \frac{\eta_t^2 - \eta_{r-t}^2}{\eta_t^2 + (q_c/q_r)\eta_t\eta_{r-t} + \eta_{r-t}^2}\right) y^t (1-y)^{r-t} \tag{5.5b}$$

and the two variables x and y are completely decoupled.

For ρ not very close to 1, $g(y)$ has three fixed points as in (4.14). The fixed point at $y = \frac{1}{2}$ is unstable, acting as the basin boundary. The other two stable fixed points are, however, no longer located at $y = 0$ and 1; in other words, error is present in the retrieved patterns.

For ρ very close to 1, the two patterns become so strongly correlated that they cannot be distinguished during retrieval. Let us consider the case in which $c \gg 1$ and $e^{-\beta} \ll 1$. In this case, the trained polarisations reduce to step functions and $g(y)$ reduces to

$$g(y) = \tfrac{1}{2} + \tfrac{1}{2} c\varepsilon \, e^{-c\varepsilon} \int_{1-y}^{y} dt \left[I_0(2c\varepsilon\sqrt{t(1-t)}) + \left(\frac{t}{1-t}\right)^{1/2} I_1(2c\varepsilon\sqrt{t(1-t)}) \right] \tag{5.6}$$

where $\varepsilon = (1-\rho)/2$ is the distance between the two patterns and the $I_\nu(x)$ are modified Bessel functions. The slope of $g(y)$ at $y = \frac{1}{2}$ becomes

$$g'(\tfrac{1}{2}) = c\varepsilon \, e^{-c\varepsilon} (I_0(c\varepsilon) + I_1(c\varepsilon)) \tag{5.7}$$

which is equal to 1 for $c\varepsilon = 1.8494$. Thus for $\varepsilon < 1.8494 c^{-1}$, $y = \frac{1}{2}$ becomes a stable fixed point, and there is only one basin of attraction: the system cannot distinguish the two patterns.

6. Storage of p uncorrelated patterns

We are now ready to study the storage of p *uncorrelated* patterns. The storage capacity is limited by the number of adjustable variables per node. Since there are 2^c sites per node, it is natural to consider p much larger than 1 but less than $2^c (c \gg 1)$.

The network dynamics is much simplified within two further approximations.

(1) *The orthogonal approximation.* The overlap of uncorrelated patterns is of the order $N^{-1/2}$, and it is important that this microscopic overlap does not significantly affect the performance of the network. Consider a microscopic overlap ρ between two patterns. The probability that they address the same site on a node is $[(1+\rho)/2]^c$, and the probability that they prescribe conflicting bits to the site is $(1-\rho)/2$. Hence the probability of a pattern being disrupted by others is $[(1+\rho)/2]^c (1-\rho)/4$, the extra factor of $\frac{1}{2}$ coming from the fact that the site outputs an incorrect bit with a probability of $\frac{1}{2}$. Averaging over a Gaussian distribution of ρ, the disruption probability for each of the p patterns is $(p/2^{c+2}) \exp(c^2/2N)$, the exponential factor being due to the microscopic overlaps. Hence the condition that the microscopic overlaps can be neglected is $c \ll N^{1/2}$ which, in the thermodynamic limit, is naturally satisfied in the regime $c \ll \ln N$ already assumed within the annealed approximation.

(2) *The mean-field approximation.* If, during retrieval, we monitor the probability of error of the Ising configuration compared with a nearby stored pattern, which we

label as pattern 1, then this probability is dependent on the number of patterns giving the same output as pattern 1 at a node. The larger the number of patterns having the same ith output as pattern 1, the more the ith node is trained with the bit of pattern 1, and hence the smaller the probability of error from the ith node. Therefore, in general, these probabilities cannot be represented by a single distance x.

Since the patterns are uncorrelated, the dynamics of the mean-field approximation can be described by p distinct variables $\{x_b;\ b=0,\ldots,p-1\}$, where x_b is the output error with respect to pattern 1 among those nodes which have b patterns with the same output bit as pattern 1. The averaged output distance is therefore a binomial average of the x_b given by

$$x=\frac{1}{2^{p-1}}\sum_{b=0}^{p-1}\binom{p-1}{b}x_b. \tag{6.1}$$

However, when the number of stored patterns p is much greater than 1, the average number giving the same output is $p/2$ with a standard deviation of the order $p^{1/2}$, so that the relative deviation is of the order $p^{-1/2}$, and becomes small as p becomes large, justifying the replacement of conditional probabilities by a single mean-field parameter x.

In order to derive the retrieval function $f(x)$ for stored pattern 1, starting from a configuration having a finite overlap with that pattern but negligible overlap with the other patterns, it is necessary to evaluate the disruptions of pattern 1 at various sites. For a particular node, let A be the set of patterns having the same output as pattern 1, and $\bar{A}$ its complement. Now consider a site on the node at site distances r^μ from each of the sites addressed by pattern μ. In analogy with the previous two-pattern example, the equilibrium occupation probability distribution of the site is given by the eigenvector of $p^{-1}\Sigma_\mu \mathbf{T}_\mu$ corresponding to the eigenvalue 1. The training matrix $\mathbf{T}_\mu$ is given by

$$\mathbf{T}_\mu=\begin{cases}\begin{pmatrix}1 & q_r\eta(r^\mu) & 0\\ 0 & 1-q_r\eta(r^\mu) & q_c\eta(r^\mu)\\ 0 & 0 & 1-q_c\eta(r^\mu)\end{pmatrix} & \text{if } \mu\in A\\ \begin{pmatrix}1-q_c\eta(r^\mu) & 0 & 0\\ q_c\eta(r^\mu) & 1-q_r\eta(r^\mu) & 0\\ 0 & q_r\eta(r^\mu) & 1\end{pmatrix} & \text{if } \mu\in\bar{A}\end{cases} \tag{6.2}$$

where $\eta(r^\mu)=d^{r^\mu}(1-d)^{c-r^\mu}$. Consequently, the trained polarisation and the u state probability at the site, averaged over all the other patterns, are given by

$$m_r=E_h\left\langle\!\!\left\langle\frac{\Sigma_\mu\sigma^\mu\exp[\beta(h\sigma^\mu-r^\mu)]}{\Sigma_\mu\exp[\beta(h\sigma^\mu-r^\mu)]}\right\rangle\!\!\right\rangle \tag{6.3}$$

$$u_r=\coth(2\beta h)O_h\left\langle\!\!\left\langle\frac{\Sigma_\mu\sigma\exp[\beta(h\sigma^\mu-r^\mu)]}{\Sigma_\mu\exp[\beta(h\sigma^\mu-r^\mu)]}\right\rangle\!\!\right\rangle \tag{6.4}$$

where $\sigma^\mu=+1$ if $\mu\in A$, and -1 if $\mu\in\bar{A}$, and $\langle\!\langle\ \rangle\!\rangle$ represents averaging over patterns 2 to p. Applying the identity $R^{-1}=\int_0^\infty \mathrm{d}z\exp(-Rz)$ to the denominators and expressing the resulting integrand as a partial derivative with respect to h, the expressions factorise over the patterns μ. Thus

$$m_r=E_h\int_0^\infty\frac{\mathrm{d}z}{z}\left(-\frac{1}{\beta}\frac{\partial}{\partial h}\right)\prod_\mu\langle\!\langle\exp\{-z\exp[\beta(h\sigma^\mu-r^\mu)]\}\rangle\!\rangle. \tag{6.5}$$

Noting that r^{μ} ($\mu \neq 1$) follows a binomial distribution, we get

$$m_r = E_h \int_0^{\infty} \frac{dz}{z}\left(-\frac{1}{\beta}\frac{\partial}{\partial h}\right)\exp(-z\,e^{\beta(h-r)}) \times\left\{\left(\frac{\exp(-z\,e^{\beta h})+\exp(-z\,e^{-\beta h})}{2}\right)\sum_{s=0}^{c}\binom{c}{s}\frac{1}{2^c}\exp(-z\,e^{-\beta s})\right\}^{p}. \tag{6.6}$$

Performing the differentiation with respect to h and using integration by parts, we finally arrive at an expression for the disruption a_r,

$$a_r = E_h \frac{\alpha}{2}\sum_{s=0}^{c}\binom{c}{s}e^{-\beta(h+s)}\int_0^{\infty} dz\exp\left(-z(e^{\beta(h-r)}+e^{-\beta(h+s)}) - \alpha\sum_{t=0}^{c}\binom{c}{t}\{1-\exp[-z\cosh(\beta h)\,e^{-\beta t}]\cosh[z\sinh(\beta h)\,e^{-\beta t}]\}\right) \tag{6.7}$$

where $\alpha = p/2^c$ is the *storage ratio.* The retrieval function follows naturally:

$$f(x) = \sum_{r=0}^{c}\binom{c}{r}a_r x^r (1-x)^{c-r}. \tag{6.8}$$

As far as the equilibrium distribution is concerned, the performance of the system is best when the training-noise temperature is low, although this has to be compensated for by increasing the number of training steps. In the limit of very low noise temperature, namely $e^{-\beta} \ll c^{-1}$, figure 4 shows the retrieval curve for x and α each of the order of $2/c^2$. In this regime, we find that

$$a_0 = \alpha/4 \tag{6.9}$$

$$a_1 = \alpha c/4 \tag{6.10}$$

$$a_2 = I(\alpha c^2/2)\alpha c^2/8 \tag{6.11}$$

$$a_r = 1/2 \qquad \text{for } r \geqslant 3 \tag{6.12}$$

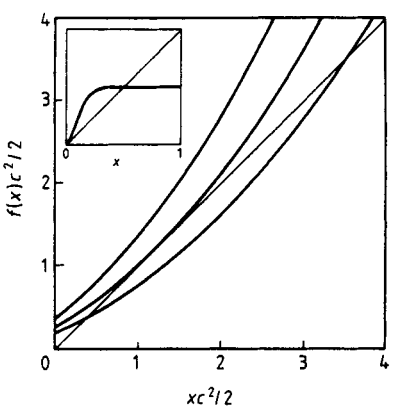

Figure 4. The retrieval curve of storing p uncorrelated patterns for x and α of the order $2/c^2$ and low noise temperature. The values of $\alpha c^2/2$ for the three curves are 0.8, 1.0875 and 1.5, from bottom to top respectively. The inset is the first curve for the full range of x for $c = 20$, showing the fixed points near $x = 0$ and $\frac{1}{2}$. Here $q_c/q_r = 1$.

where

$$I(y)=\int_0^\infty \mathrm{d}z \exp\left\{-z-y\left[1-\mathrm{e}^{-z/2}\cosh\left(\frac{z}{2}\tanh(\beta h)\right)\right]\right\}. \tag{6.13}$$

The vertical intercept of the retrieval curve, $\alpha/4$, gives the output error when (noiseless) patterns are presented to the network input. This intercept is effectively at zero only for $\alpha<4/N$, implying that the storage capacity for errorless retrieval is $\alpha=4/N$. The same result is also derived from a consideration of the optimal storage capacity [19].

For x of the order $2/c^2$, only terms up to $r=2$ are important. The retrieval function is therefore quadratic in x. There exists a critical value α_c, given by

$$\alpha_c=\tilde{\alpha}(2/c^2) \tag{6.14}$$

where

$$\tilde{\alpha}=2/(1+\sqrt{I(\tilde{\alpha})}). \tag{6.15}$$

For $q_c/q_r=1$, α_c is $1.0875(2/c^2)$. As shown in figure 4, the retrieval function for $\alpha<\alpha_c$ has two fixed points, one stable and one unstable. The stable fixed point is non-zero, showing that the retrieved pattern has a certain error. The unstable fixed point indicates the basin boundary, separating the basins of attractors near $x=0$ and $\frac{1}{2}$, corresponding to retrieval and non-retrieval respectively. At $\alpha=\alpha_c$, the two fixed points coincide, and for α slightly above α_c, the neighbourhood of the fixed points becomes a 'bottleneck' area. Starting from an initial x smaller than the bottleneck, x approaches it on iteration. It stays there for a number of steps before it eventually diverges towards the fixed point near $\frac{1}{2}$. This transient behaviour diminishes as α increases away from the critical region. Thus for $\alpha>\alpha_c$, the retrieval curve does not have any fixed points near $x=0$, and the stored pattern cannot be retrieved. $\alpha_c 2^c$ is therefore the *storage capacity for associative memory* of uncorrelated patterns.

That the maximum storage ratio α_c is of the order $2/c^2$ illustrates the structure of storage in each node. At the storage ratio α, the average number of patterns whose addresses are at site distances r from a random reference site is equal to $\alpha\binom{c}{r}$. Therefore, the value found for α_c implies that at the critical storage ratio there is an average of about one pattern at a site distance 2. That the storage capacity scales as $c^{-2}2^c$ suggests that the present model is a very powerful one for memory; by contrast, a dilute asymmetric Hopfield-Little system with the same network topology but with synaptic storage has a maximum storage capacity for uncorrelated patterns of only $0.64c$ [16], which is much smaller for large c.

Figure 5(a) shows the positions of the two fixed points near $x=0$ as a function of the storage ratio α for $\alpha<\alpha_c$. The lower fixed point gives the retrieval error and the upper fixed point gives the radius of attraction. As α increases, the retrieval error curve becomes increasingly steep until it becomes vertical at α_c, where a first order transition of the equilibrium configuration takes place. For $q_c/q_r=1$, the retrieval error jumps discontinuously at $\alpha=\alpha_c$ from $2.3836/c^2$ to an order of $\frac{1}{2}$. The radius of attraction decreases with the storage ratio up to α_c, where it merges with the retrieval error.

Again, it is interesting to compare the system with its conventional counterpart, namely the dilute asymmetric Hopfield-Little network [16]. In this conventional model, the retrieval function is given by

$$f(x)=\frac{1}{2}\left[1-\mathrm{erf}\left(\frac{1-2x}{\sqrt{2\alpha}}\right)\right] \tag{6.16}$$

where $\alpha=p/c$. As shown in figure 5(b), the retrieval error, given by the stable fixed

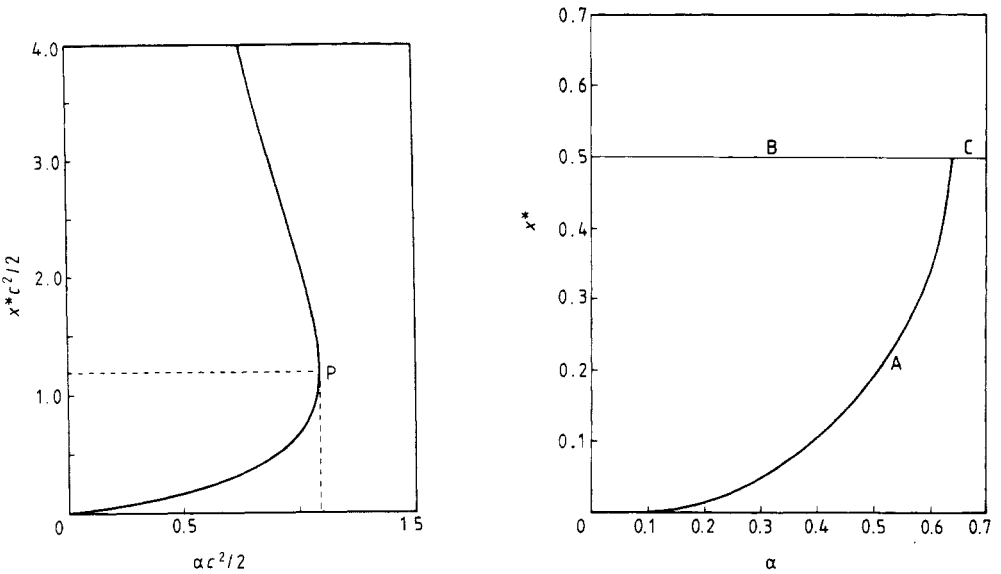

Figure 5. (*a*) The dependence of the two fixed points on the storage ratio for $q_c/q_r = 1$. The critical point P is given by $\alpha_c c^2/2 = 1.0875$ and $x^* c^2/2 = 1.1918$. (*b*) The corresponding curves for the dilute asymmetric Hopfield-Little model [16]: for $\alpha < 0.64$ curve A gives the retrieval error and line B gives the radius of attraction; for $\alpha > 0.64$ retrieval is no longer possible and the error is given by line C.

point of the curve, undergoes a *second*-order transition at the critical value $\alpha_c = 2/\pi = 0.64$, in contrast with the *first*-order transition in the present network. The *fully connected* symmetric Hopfield network [3], however, exhibits a first-order transition of the retrieval error, more analogous to the system we have been considering.

On the other hand, the radius of attraction of the dilute asymmetric Hopfield-Little model [16] is independent of α and takes the value $\frac{1}{2}$, in contrast with the α-dependent radius of attraction in the present model. This large basin of attraction is related to the 'long-ranged' nature of the conventional model as discussed in § 8. Rather, our curve for the radius of attraction as a function of the storage ratio α in the present model can be compared, roughly speaking, with the stability K in perceptron models [20]. Since the stability K is a measure of the size of the basins of attraction [21], the two models share the common feature that the storage of more patterns reduces the size of the basin of attraction of each one.

Finally, in this section it should, however, be stressed that although the retrieval error is given by a fixed point x^*, it does not necessarily imply that the final configuration is a stable one: it may happen that in the long-time limit, the configuration evolves throughout the totality of the configuration hypersphere of distance x^* from the stored pattern, or explores only a subspace, or even approaches a stable point in the configuration space. These possibilities will be discussed in § 9.

7. Effects of various training parameters

We have mentioned that the maximum storage ratio for associative memory is $1.0875(2/c^2)$ for $q_c/q_r = 1$ and for low training-noise temperature. In general, however,

the storage of the system depends on both the probability ratio q_c/q_r and the training-noise temperature.

Since q_c and q_r respectively determine the rates at which a site enters and leaves the u state during training, the ratio q_c/q_r determines the fraction of u sites in the system after very long training. However, as shown in figure 6, the storage capacity at low training-noise temperature is only weakly dependent on q_c/q_r. For low training-noise temperature $\tilde{\alpha}$ is largest at $q_c/q_r = 0$ where it takes the value 1.0968. For $q_c/q_r \gg 1$, the asymptotic value of a_2 is

$$a_2 \xrightarrow{q_c/q_r \gg 1} \tfrac{1}{2}[1-\exp(-\alpha c^2/4)] \tag{7.1}$$

giving $\tilde{\alpha} \xrightarrow{q_c/q_r \gg 1} 1.0634$, which is smaller than the largest $\tilde{\alpha}$ by only 3%.

A related issue is to compare the three-state system with the two-state one without the u states. Using arguments similar to those presented in § 4, we can show that the retrieval behaviour of the two-state system approaches that of the three-state system with $q_c/q_r = 2$ after long training. Its storage capacity is therefore given by $\tilde{\alpha} = 1.0823$. Thus the network performances of both systems as associative memories are comparable after sufficiently long training in the thermodynamic limit, despite the anticipated superiority [10] of the three-state nodes.

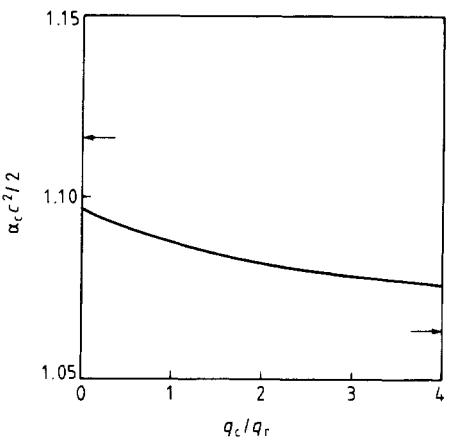

Figure 6. The dependence of the storage capacity $\alpha_c c^2/2$ on the ratio q_c/q_r for low training noise. The right arrow gives the asymptotic storage capacity when $q_c/q_r \gg 1$, and the left arrow gives the storage capacity corresponding to the majority rule (§ 7).

For higher training-noise levels where $e^{-\beta} \sim O(c^{-1})$, the expressions for the disruptions up to site distance 2, by virtue of (6.7), become

$$a_0 = \frac{\alpha}{4}\left(1+\frac{q_c}{q_r}(e^{\gamma}-1)\right) \tag{7.2}$$

$$a_1 = \frac{\alpha c}{4}\left(1+\frac{q_c}{q_r}\left(\frac{e^{\gamma}-1-\gamma}{\gamma}\right)\right) \tag{7.3}$$

$$a_2 = \frac{\alpha c^2}{8}\left(J_1\left(\frac{\alpha c^2}{2}\right)+\frac{e^{\gamma}-1-\gamma-\gamma^2/2}{\gamma^2/2}J_2\left(\frac{\alpha c^2}{2}\right)\right) \tag{7.4}$$

where $e^{-\beta} = \gamma/c$ and

$$J_1(y) = 2\int_0^\infty dz \exp\left[-z\left(2 + y\frac{e^\gamma - 1 - \gamma - \gamma^2/2}{\gamma^2/2}\right) - y\{1 - e^{-z}\cosh[z\tanh(\beta h)]\}\right] \tag{7.5}$$

$$J_2(y) = \frac{2}{\cosh(\beta h)}\int_0^\infty dz \cosh[\beta h + z\tanh(\beta h)] \times \exp\left[-z\left(1 + y\frac{e^\gamma - 1 - \gamma - \gamma^2/2}{\gamma^2/2}\right) - y\{1 - e^{-z}\cosh[z\tanh(\beta h)]\}\right] \tag{7.6}$$

and the expression for the storage parameter $\tilde{\alpha}$ has to be replaced by

$$\tilde{\alpha} = 2\left\{1 + \frac{q_c}{q_r}\frac{e^\gamma - 1 - \gamma}{\gamma} + \left[\left(1 + \frac{q_c}{q_r}(e^\gamma - 1)\right)\left(J_1(\tilde{\alpha}) + \frac{e^\gamma - 1 - \gamma - \gamma^2/2}{\gamma^2/2}J_2(\tilde{\alpha})\right)\right]^{1/2}\right\}^{-1} \tag{7.7}$$

As the noise parameter γ increases, the asymptotic behaviour of a_2 is

$$\begin{aligned} a_2 &\xrightarrow{\gamma \gg 1} \frac{\alpha c^2}{4}\frac{e^\gamma}{\gamma^2}\frac{q_c}{q_r} && \text{for } q_c/q_r \neq 0 \\ a_2 &\xrightarrow{\gamma \gg 1} \tfrac{1}{2} && \text{for } q_c/q_r = 0 \end{aligned} \tag{7.8}$$

and the storage capacity becomes

$$\begin{aligned} \tilde{\alpha} &\xrightarrow{\gamma \gg 1} \frac{2(\sqrt{2}-1)}{q_c/q_r}\gamma e^{-\gamma} && \text{for } q_c/q_r \neq 0 \\ \tilde{\alpha} &\xrightarrow{\gamma \gg 1} 3 - \sqrt{5} = 0.7639 && \text{for } q_c/q_r = 0. \end{aligned} \tag{7.9}$$

As shown in figure 7, the storage capacity decreases with the noise parameter for all values of q_c/q_r. In contrast to the case of low training-noise temperature, the storage

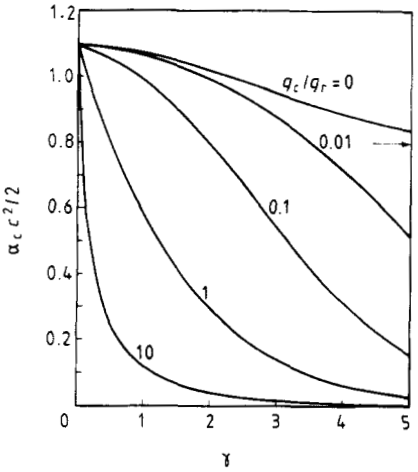

Figure 7. The dependence of the storage capacity $\alpha_c c^2/2$ on the noise parameter γ for $q_c/q_r = 0, 0.01, 0.1, 1, 10$. The arrow gives the storage capacity for $q_c/q_r = 0$ when $1 \ll \gamma \ll \ln c$.

capacity reduction is very sensitive to the ratio q_c/q_r, indicating that the u sites resulting from the noisy training procedure play an important role in disrupting the stored information.

It is interesting to note that the performance of the system is least susceptible to the training-noise level for $q_c/q_r = 0$. This is because the lowest order of the training-noise correction at site distances 0 and 1 comes from the presence of u sites. The fraction of u sites diminishes for $q_c/q_r \to 0$, whereas algorithms with large values of q_c/q_r have a greater chance to disrupt correct bits to u through the training-with-noise procedure. Thus for $q_c/q_r = 0$, a training noise of the order c^{-1} leaves the disruptions at site distances 0 and 1 unaltered, as evident in (7.2) and (7.3). It merely randomises sites at site distance 2, as in (7.8), giving a finite storage capacity of $\tilde{\alpha} = 0.7639$ as γ becomes much greater than unity. Further calculations show that the storage capacity eventually deteriorates only when the noise parameter γ becomes of the order of $\ln c$, in which case the disruptions due to incorrect bits become significant at site distances 0 and 1.

It seems paradoxical that the system performs best for low training-noise levels, while training with noise was originally introduced by Aleksander [10] to ensure associative memory. Equally puzzling, at first, is the fact that better storage performance is attained for zero or small values of q_c/q_r, while the step of correcting incorrect bits into u states in the training-with-noise algorithm was originally introduced to improve the training procedure. While explanatory discussions will be detailed elsewhere [17], we remark, however, that the above results apply to the *equilibrium* state of the network (i.e. after an infinite number of training steps). Algorithms with suitably higher levels of training noise and larger values of q_c/q_r, on the other hand, take fewer training steps to approach the equilibrium state, and hence are more efficient as far as the training procedure is concerned. It follows that the strategy for best storage performance is to use algorithms with low training-noise levels and small values of q_c/q_r, although this does not economise the training time.

8. Proximity rules

It is instructive to consider the equilibrium distribution of a site in the limit of low training-noise level. In this limit, we see that at any site

$$(P_+, P_0, P_-) = \frac{1}{n_A^2 + (q_c/q_r) n_A n_{\bar{A}} + n_{\bar{A}}^2} (n_A^2, (q_c/q_r) n_A n_{\bar{A}}, n_{\bar{A}}^2). \tag{8.1}$$

Here, n_A and $n_{\bar{A}}$ are the number of nearest neighbours, respectively, belonging to sets A and $\bar{A}$ defined in § 6. In particular, $(P_+, P_0, P_-) = (1, 0, 0)$ if the nearest pattern belongs to A, and $(P_+, P_0, P_-) = (0, 0, 1)$ if the nearest pattern belongs to $\bar{A}$.

We therefore conclude that the training-with-noise algorithm is equivalent to the following *proximity rule* in the limit of low training-noise level.

The content of a site follows the pattern which is its nearest neighbour, if there is only one such. If it has more than one pattern as its nearest neighbour, the site is filled according to the relative probability $P_+ : P_0 : P_- = n_A^2 : (q_c/q_r) n_A n_{\bar{A}} : n_{\bar{A}}^2$.

The correction due to finite training noise is of the order $ce^{-\beta}$.

The above probability ratio can be viewed as a manifestation of detailed balance. At equilibrium, the transition rate from one state to the other should be the same in the forward and backward directions. Thus, for example, the transition rates from the

correct to u state and vice versa are respectively $P_+ q_c n_{\bar{A}}/p$ and $P_0 q_r n_A/p$. Detailed balance therefore yields $P_+ : P_0 = n_A : (q_c/q_r) n_{\bar{A}}$.

The equivalence of the training algorithm and the proximity rule can be demonstrated by studying the disruptions. Consider a site at a site distance r from a particular pattern. The probability of having another equidistant pattern is $\alpha\binom{c}{r}$, corresponding for $r = 0$, 1 to the disruptions

$$a_0 = \alpha/4 \tag{8.2}$$

$$a_1 = \alpha c/4. \tag{8.3}$$

(Here we have taken into account the probability of $\frac{1}{2}$ that the output bit of the interfering patterns are different, and another $\frac{1}{2}$ that the output is incorrect.) These values are precisely those found earlier.

The disruption at site distances 2 is most interesting. Since a site has an average of $\alpha c^2/2$ patterns at site distances 2, half of them having the same output bit and half of them different on average, the disruption is given by the Poisson-averaged expression

$$a_2 = \sum_{r,s} \frac{\exp(-\alpha c^2/4)}{r!} \left(\frac{\alpha c^2}{4}\right)^r \frac{\exp(-\alpha c^2/4)}{s!} \left(\frac{\alpha c^2}{4}\right)^s E_h \frac{(r+1)\,\mathrm{e}^{\beta h} - s\,\mathrm{e}^{-\beta h}}{(r+1)\,\mathrm{e}^{\beta h} + s\,\mathrm{e}^{-\beta h}} \tag{8.4}$$

where the (r, s)th term corresponds to the case $n_A = r+1$ and $n_{\bar{A}} = s$ when there are $r+s$ nearest neighbours at site distances 2. Using the mathematical techniques for manipulating the pattern-averaged trained polarisation introduced in § 6, we arrive at the correct expression (6.11) for a_2.

Similarly

$$a_r = \tfrac{1}{2} \qquad \text{for } r \geqslant 3. \tag{8.5}$$

The general expression (6.7) for the disruption can be interpreted similarly. It is a summation of a total of c terms, the sth term corresponding to the disruption by a pattern at a site distance s.

It is now clear why algorithms with different ratios of q_c/q_r only have a slight difference in their storage capacities. All of them have the same disruptions at various site distances, except where at site distances 2 the fraction of u sites is different.

When $q_c/q_r \gg 1$, a site at a site distance 2 is, according to the proximity rule, almost certainly in the u state whenever it has another nearest neighbour at the same site distance, this happening with a probability $1-\exp(-\alpha c^2/4)$. Thus we have

$$a_2 \xrightarrow{q_c/q_r \gg 1} \tfrac{1}{2}[1-\exp(-\alpha c^2/4)] \tag{8.6}$$

giving $\tilde{\alpha} \xrightarrow{q_c/q_r \gg 1} 1.0634$, which is the result obtained in the last section.

The u state probabilities can likewise be calculated using the proximity rule. We obtain

$$u_0 = \frac{\alpha}{2} \frac{q_c/q_c}{2+q_c/q_r} \tag{8.7}$$

$$u_1 = \frac{\alpha c}{2} \frac{q_c/q_r}{2+q_c/q_r} \tag{8.8}$$

$$u_2 = \frac{\alpha c^2}{4} I\left(\frac{\alpha c^2}{2}\right) \frac{q_c/q_r}{2+q_c/q_r} \tag{8.9}$$

$$u_r = \exp(-\alpha c^2/2)\frac{q_c/q_r}{2+q_c/q_r} + \frac{\alpha c^2}{4\tanh(2\beta h)} J\left(\frac{\alpha c^2}{2}\right) \qquad \text{for } r \geqslant 3 \tag{8.10}$$

where

$$J(y) = \int_0^\infty \mathrm{d}z \sinh[\beta h - z\sinh(\beta h)] \times \exp\{[-z\cosh(\beta h) - y\{1-\exp[-z\cosh(\beta h)]\cosh[z\sinh(\beta h)]\}]\}. \tag{8.11}$$

These results are identical to those obtained by performing the pattern averaging explicitly in (6.4) and taking the limit $\mathrm{e}^{-\beta} \ll c^{-1}$.

It is interesting to evaluate the overall fraction of u-state sites, which is

$$\langle u_r \rangle = \frac{1}{2^c}\sum_r \binom{c}{r} u_r \approx u_3 = \exp(-\alpha c^2/2)\frac{q_c/q_r}{2+q_c/q_r} + \frac{\alpha c^2}{4\tanh(2\beta h)} J\left(\frac{\alpha c^2}{2}\right). \tag{8.12}$$

The two terms in the above expression have interesting interpretations. With probability $\exp(-\alpha c^2/2)$, a site has no nearest neighbours up to a site distance 2, and its content is, according to the proximity rule, determined by nearest neighbours at site distances 3. The average number of nearest neighbours at site distances 3 is $\alpha c^3/6 \gg 1$, and we have $n_A = n_{\bar{A}} = \alpha c^3/12$ with almost certainty, resulting in the first term. The second term describes the u-state probability when the site has nearest neighbours at site distances 2.

When $q_c/q_r \gg 1$, the fraction of u sites becomes

$$\langle u_r \rangle \xrightarrow{q_c/q_r \gg 1} \exp(-\alpha c^2/2) + [1-\exp(-\alpha c^2)/4]^2. \tag{8.13}$$

This is because, for $q_c/q_r \gg 1$, a site is almost certainly in the u state when either there are no nearest neighbours at site distances 2, or there exist some with differing output bits. (These two cases correspond respectively to the two terms in the above expression.)

The u-state probability is plotted in figure 8 for q_c/q_r from 0 to 4 at the critical storage ratio α_c. It approaches the value 0.5153 in the limit $q_c/q_r \gg 1$. It is remarkable that as the ratio q_c/q_r increases, the storage capacity only decreases slightly (see figure 5) although the fraction of u sites increases quite significantly. This is because associative memory depends primarily on there being only a small disruption at a site distance 1 from a pattern, and these sites are statistically insignificant. The sites at site distances 2 from some pattern, although statistically significant, only have a secondary consequence for associative memory.

It is interesting to see whether we can further increase the storage capacity beyond that of the above proximity rule. In this respect, we note that the storage capacity of the proximity rule is largest when $P_0 = 0$ (i.e. $q_c/q_r = 0$) where $\tilde{\alpha} = 1.0968$. Now consider a modified proximity rule in which $P_+ : P_0 : P_- = n_A^z : 0 : n_{\bar{A}}^z$ where z is an exponent to be determined. For $z = 1$, the trained polarisation at a site is

$$\frac{n_A - n_{\bar{A}}}{n_A + n_{\bar{A}}} = \frac{n_A^z - n_{\bar{A}}^z}{n_A^2 + 2n_A n_{\bar{A}} + n_{\bar{A}}^2}$$

and the rule is equivalent to a training algorithm with $q_c/q_r = 2$, giving $\tilde{\alpha} = 1.0823$. For $z = 2$, the rule is equivalent to a training algorithm with $q_c/q_r = 0$ and hence $\tilde{\alpha} = 1.0968$. It is therefore natural to expect that the storage capacity increases with z, since the trained polarisations become more and more favourable. In the limit of $z \to \infty$, we obtain the following *majority rule.*

The content of a site follows the pattern which is its nearest neighbour, if there is only one such. If it has more than one pattern as its nearest neighbour, the content follows the majority. If there is no majority, the site is randomly filled with 1 *or* 0 *with equal probability.*

We could also consider the use of u states for sites with no majority. There would be no consequence for retrieval error within the realm of validity of the annealed approximation.

In this case, the disruption at a site distance 2 is given by

$$a_2 = \sum_{r,s} \frac{\exp(-\alpha c^2/4)}{r!}\left(\frac{\alpha c^2}{4}\right)^r \frac{\exp(-\alpha c^2/4)}{s!}\left(\frac{\alpha c^2}{4}\right)^s \frac{1}{2}[1-\operatorname{sgn}(r+1-s)] \tag{8.14}$$

which gives a storage capacity of

$$\tilde{\alpha} = 2\left[1+\left(\frac{1-e^{-\tilde{\alpha}}(I_0(\tilde{\alpha})+I_1(\tilde{\alpha}))}{\tilde{\alpha}}\right)^{1/2}\right]^{-1} = 1.1165. \tag{8.15}$$

This leads, therefore, to a further 2% increase in the storage capacity beyond that of the training-with-noise algorithm (see figure 6).

We can also calculate the fraction of sites with no majority at $\tilde{\alpha}$. This gives the maximum fraction of sites that can be filled by u without any consequences for the behaviour of the system within the annealed approximation. Its value is

$$\langle u_r\rangle \simeq u_3 = \sum_{r=1}^{\infty} \frac{\exp(-\tilde{\alpha}/2)}{r!}\left(\frac{\tilde{\alpha}}{2}\right)^r \frac{\exp(-\tilde{\alpha}/2)}{r!}\left(\frac{\tilde{\alpha}}{2}\right)^r = e^{-\tilde{\alpha}}(I_0(\tilde{\alpha})-1) = 0.1103 \tag{8.16}$$

(see figure 8).

The majority rule leads us to a new way of training the network. We call this the *diffusion algorithm.*

(1) Each pattern is stored at the correct sites of each node. If more than one pattern addresses a site, its content is determined by the majority rule.

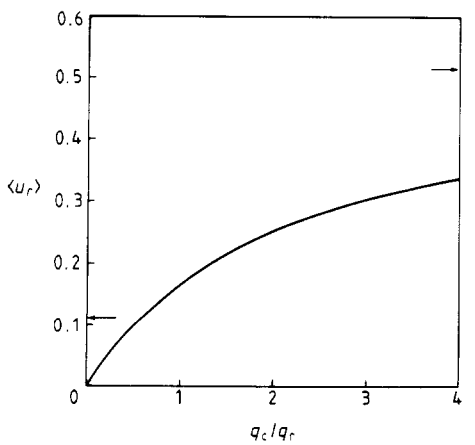

Figure 8. The dependence of the fraction of sites in u states on the ratio q_c/q_r at the critical storage ratio. The right arrow gives the asymptotic fraction of u sites when $q_c/q_r \gg 1$, and the left arrow gives the fraction of sites with no majority (i.e. $n_A = n_{\bar{A}}$) at the critical storage ratio of the majority rule.

(2) The content of these sites are copied to their nearest neighbours which are still unaddressed. If more than one pattern addresses a site, its content is determined by the majority rule.

(3) The above process (2) is continued until all the sites are filled.

9. Retrieval properties

The average retrieval properties of the Boolean model are determined by the retrieval function in the approximations stated above. In § 6, we have compared with the Hopfield model such aspects as the storage capacity and retrieval error. In order to compare the two models further, we are going to study two more situations.

First we consider the evolution of two different initial configurations $\{S_i\}$ and $\{\tilde{S}_i\}$ having the same distance from a stored pattern (say pattern 1). Because of the presence of u sites, outputting randomly 1 or 0 every time they are addressed, we can distinguish the following three types of dynamics governing the evolution of the two configurations when addressing u sites.

(1) *Same dynamics for each site.* This means that, at each instant, the *same* random bit is output for each configuration when they address the *same* site in a u state, but the output is independent when they address different sites in u states. This is equivalent to storing at each time instant randomly 'quenched' values of 1 or 0 at the sites in u states, so that the same output is obtained when addressed by either configuration.

(2) *Independent dynamics.* This means that, at each instant, independently random bits are output for the two configurations if they address sites in u states (even when, say, they address the same site in a u state).

(3) *Same dynamics for each node.* This means that, at each instant, the same random bit is output from a node for each configuration when they both address a site in a u state (whether the sites addressed are the same or not).

These three types of dynamics are summarised in figure 9.

Let us first consider the type (1) dynamics, the same for each site. As in the work of Derrida *et al* [16], we consider three variables: x and $\tilde{x}$ being respectively the

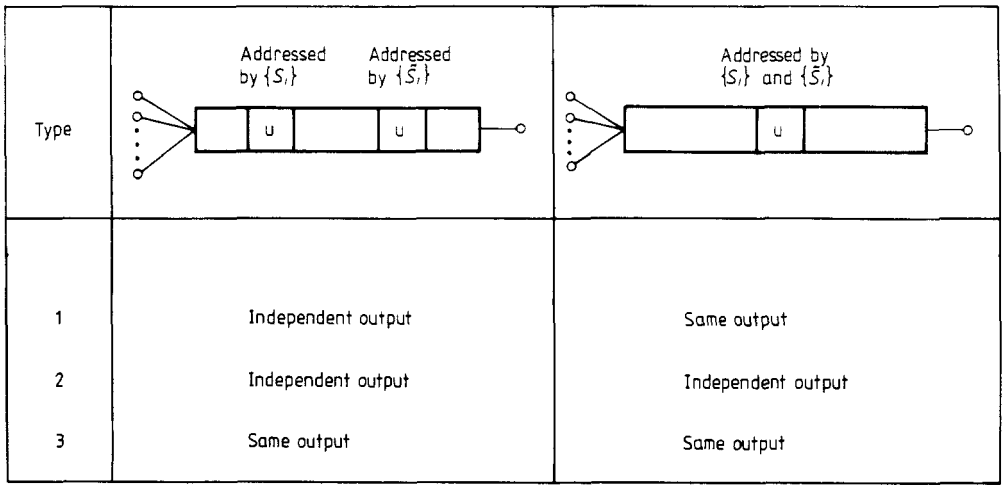

Figure 9. The three types of dynamics governing the evolution of two configurations.

distances of $\{S_i\}$ and $\{\tilde{S}_i\}$ with respect to pattern 1, and y being the distance between $\{S_i\}$ and $\{\tilde{S}_i\}$.

The evolutions of x and $\tilde{x}$ are still determined by the retrieval function (6.8). For an analysis of the evolution of y, let us consider the four probabilities:

$$p_1 = \text{Prob}(\xi_i^1 = S_i = \tilde{S}_i) = 1 - \tfrac{1}{2}(x + \tilde{x} + y) \tag{9.1}$$

$$p_2 = \text{Prob}(\xi_i^1 = S_i = -\tilde{S}_i) = \tfrac{1}{2}(-x + \tilde{x} + y) \tag{9.2}$$

$$p_3 = \text{Prob}(\xi_i^1 = -S_i = \tilde{S}_i) = \tfrac{1}{2}(x - \tilde{x} + y) \tag{9.3}$$

$$p_4 = \text{Prob}(\xi_i^1 = -S_i = -\tilde{S}_i) = \tfrac{1}{2}(x + \tilde{x} - y). \tag{9.4}$$

To trace the discrepancy between the two configurations, we have to consider the probability that either of them, but not both, is disrupted. Hence the evolution of y is determined by the following output function:

$$g(y) = \sum_{\substack{n_1+n_2+n_3+n_4=c \\ (n_2,n_3)\neq(0,0)}} \frac{c!}{n_1!\,n_2!\,n_3!\,n_4!} p_1^{n_1} p_2^{n_2} p_3^{n_3} p_4^{n_4} [a_{n_3+n_4}(1 - a_{n_2+n_4}) + a_{n_2+n_4}(1 - a_{n_3+n_4})]. \tag{9.5}$$

Here, the term corresponding to $(n_2, n_3) = (0, 0)$ is omitted. This is because when the two configurations address different sites, the disruption probabilities are in general independent, but this is not the case when the two configurations address the same site.

As x and $\tilde{x}$ approach x^*, the above function reduces to

$$g(y) = \frac{\alpha c^2}{4}\left(1 + \frac{c^2}{2} I x^*\right) y - \frac{\alpha c^4}{32} I y^2 \tag{9.6}$$

where $I = I(\alpha c^2/2)$ is the integral given by (6.13). Since $g'(0)$ is equal to $f'(x^*)$ by virtue of (6.8), it is less than 1 for $\alpha < \alpha_c$, and $y = 0$ is the only stable fixed point. This means that near a stored pattern, two different initial configurations converge to the same fixed point, at least in the annealed approximation. This picture of a simple attractor is in direct contrast with the complex attractor in the case of the dilute asymmetric Hopfield-Little model [16].

For type (2) dynamics (independent dynamics), there will be output discrepancy between the two configurations even when input discrepancy is absent. This is the case when they address the same site in a u state, hence outputting a different bit with probability $\frac{1}{2}$. Thus an extra term

$$\frac{1}{2}\sum_r \binom{c}{r} u_r p_1^{c-r} p_4^r$$

should be included in the retrieval function (9.5), and as x and $\tilde{x}$ approach x^*, (9.6) has to be augmented by the term

$$\frac{q_c/q_r}{2 + q_c/q_r}\left[x^* - \frac{\alpha c^2}{8}\left(1 + \frac{c^2}{2} I x^*\right) y + \frac{\alpha c^4}{64} I y^2\right].$$

Now the retrieval function for y has a non-zero stable fixed point, and the attractor is therefore complex as in the case of the dilute asymmetric Hopfield-Little model [16].

Type (3) dynamics (same dynamics for each node) gives a retrieval function the same as (9.6) for $y \sim O(c^{-2})$, since in this regime the probability that the two configurations address different sites in u states is negligible. In this case, the attractor is again a simple one.

Another interesting situation arises when a configuration $\{S_i\}$ has considerable overlap with two patterns, $\{\xi_i^1\}$ and $\{\xi_i^2\}$. We shall consider the case for which the two patterns 1 and 2 differ by a distance ε considerably less than $\frac{1}{2}$; the other $p-2$ patterns being random. As in § 5, the evolution of the system is determined by two parameters: x and y being the fractional errors with respect to pattern 1 for those nodes outputting respectively the same and different bits for patterns 1 and 2. The retrieval functions for $\varepsilon \ll 1$ are given by

$$f(x,y)=\sum_{s,t}\frac{e^{-c\varepsilon y}}{s!}(c\varepsilon y)^s\frac{e^{-c\varepsilon(1-y)}}{t!}[c\varepsilon(1-y)]^t\sum_r\binom{c-s-t}{r}b(s+r,t+r)x^r(1-x)^{c-s-t-r} \tag{9.7}$$

$$g(x,y)=\sum_{s,t}\frac{e^{-c\varepsilon y}}{s!}(c\varepsilon y)^s\frac{e^{-c\varepsilon(1-y)}}{t!}[c\varepsilon(1-y)]^t\sum_r\binom{c-s-t}{r}c(s+r,t+r)x^r(1-x)^{c-s-t-r} \tag{9.8}$$

where $b(r,s)$ and $c(r,s)$ are the disruptions of pattern 1 at site distances r from pattern 1 and s from pattern 2, for nodes outputting the same and different bits of the two patterns respectively. For $q_c/q_r=1$ in the limit of low training-noise temperature, proximity rule considerations yield

$$b(0,0)=\alpha/7 \tag{9.9}$$

$$b(0,r)=b(r,0)=\alpha/4 \qquad \text{for } r\geqslant 1 \tag{9.10}$$

$$b(1,1)=\alpha c/7 \tag{9.11}$$

$$b(1,r)=b(r,1)=\alpha c/4 \qquad \text{for } r\geqslant 2 \tag{9.12}$$

$$b(2,2)=(\alpha c^2/14)K(\alpha c^2/2) \tag{9.13}$$

$$b(2,r)=b(r,2)=(\alpha c^2/8)I(\alpha c^2/2) \qquad \text{for } r\geqslant 3 \tag{9.14}$$

where $I(y)$ is given in (6.13),

$$K(y)=\frac{7}{2}\int_0^\infty dz\cos\left(\frac{\pi}{6}+\frac{3}{2}\right)\exp\left\{-\frac{3\sqrt{3}}{2}z-y\left[1-\exp\left(-\frac{\sqrt{3}}{2}z\right)\cos\frac{z}{2}\right]\right\} \tag{9.15}$$

and

$$c(r,s)=\theta(s-r)a_r+\theta(r-s)(1-a_s) \tag{9.16}$$

where the a_r are the single-pattern disruptions employed in the previous sections.

When ε decreases from $\frac{1}{2}$, the increasing correlation of the two patterns causes mutual disruption. Thus the critical storage ratio at which the network fails to retrieve the two patterns drops. This drop becomes significant when $\varepsilon \geqslant \ln c/c$. Indeed, when $c\varepsilon\, e^{-c\varepsilon}\sim O(c^{-1})$, the retrieval functions for (x,y) near $(0,0)$ reduce to

$$f(x,y)=\frac{\alpha}{4}\left(1+c^2\varepsilon y+\frac{I}{4}(c^2\varepsilon y)^2\right)+\frac{\alpha c^2}{4}\left(1+\frac{I}{2}c^2\varepsilon y\right)x+\frac{\alpha c^4}{16}Ix^2 \tag{9.17}$$

$$g(x,y)=\tfrac{1}{2}e^{-c\varepsilon}. \tag{9.18}$$

The critical storage ratio α_1 for the retrieval of the two patterns is then given by

$$\frac{\alpha_1 c^2}{2}=2\left[\left[1+\frac{\mu}{4}I\left(\frac{\alpha_1 c^2}{2}\right)+\left\{I\left(\frac{\alpha_1 c^2}{2}\right)\left[1+\frac{\mu}{2}+\frac{\mu^2}{16}I\left(\frac{\alpha_1 c^2}{2}\right)\right]\right\}^{1/2}\right]\right]^{-1} \tag{9.19}$$

where $\mu = c^2\varepsilon\, e^{-c\varepsilon}$. Since ε decreases as μ increases, μ is a parameter measuring the overlap of the two patterns. The dependence of α_1 on the overlap parameter μ is shown in figure 10(a).

For $\varepsilon \sim O(c^{-1})$, the critical storage ratio α_1 drops sharply. The storage in this regime is best expressed in terms of r_0 which is the average site distance between nearest-neighbouring patterns, namely $\alpha = (\frac{c}{r_0})^{-1} (1 \ll r_0 \ll c)$. As derived in the appendix, the critical r_0 for associative memory of the two patterns obeys the scaling relation

$$\left(\frac{r_0}{ec\varepsilon\sqrt{y^*(1-y^*)}}\right)^{2r_0} \sim \sqrt{r_0}\, c\left(\frac{\exp(-c\varepsilon)}{7(2\pi)^{3/2}c^2\varepsilon^2 y^*(1-y^*)}\right) \tag{9.20}$$

where y^* is the fixed point of the retrieval function (5.6).

For $\varepsilon < 1.8494c^{-1}$, the two patterns have become so strongly correlated that the system cannot distinguish them. In other words, there is only one stable fixed point near $x = 0$ with $y = \frac{1}{2}$. In the range $\varepsilon \sim O(c^{-3/2})$, the retrieval function near this fixed point is given by

$$f(x, \tfrac{1}{2}) = \frac{\alpha}{7}\left(1 + \frac{c^3\varepsilon^2}{4} + \frac{c^6\varepsilon^4}{128}K\right) + \frac{\alpha c^2}{7}\left(1 + \frac{c^3\varepsilon^2}{8}K\right) + \frac{\alpha c^4}{28}Kx^2. \tag{9.21}$$

The critical storage ratio α_2 for which the system fails to retrieve the mixed pattern is given by

$$\frac{\alpha_2 c^2}{2} = \frac{7}{2}\left[\left[1 + \frac{c^3\varepsilon^2}{8}K\left(\frac{\alpha_2 c^2}{2}\right) + \left\{K\left(\frac{\alpha_2 c^2}{2}\right)\left(1 + \frac{c^3\varepsilon^2}{4} + \frac{c^6\varepsilon^4}{128}K\left(\frac{\alpha_2 c^2}{2}\right)\right)\right\}^{1/2}\right]\right]^{-1}. \tag{9.22}$$

For $\varepsilon = 0$, $\alpha_2 c^2/2 = 1.9155$. For $\alpha > \alpha_2$, the system does not remember anything, whereas for $\alpha < \alpha_2$, the system remembers the patterns but cannot distinguish them. The dependence of α_2 on the distance ε is shown in figure 10(b).

We can therefore deduce the phase diagram schematically shown in figure 11(a). Like the Hopfield model [16], we have three phases and a triple point. In general,

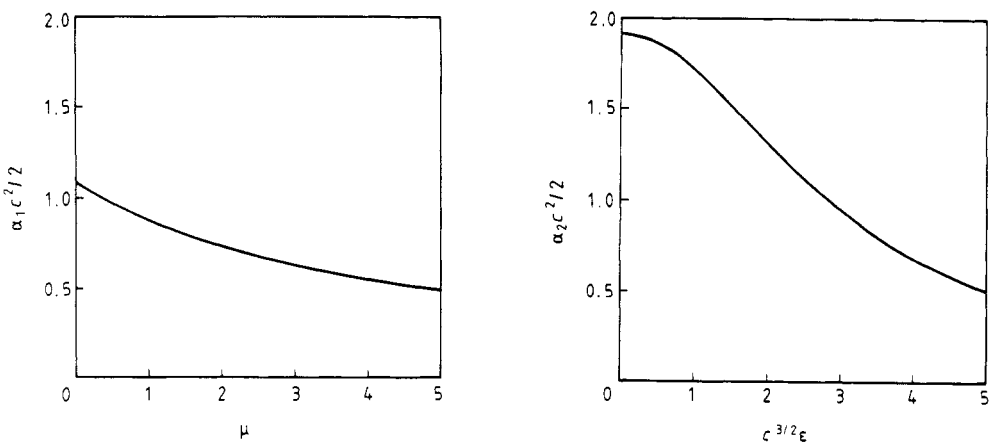

Figure 10. (a) The dependence of the storage ratio $\alpha_1 c^2/2$ on the overlap parameter μ for $q_c/q_r = 1$. (b) The dependence of the storage ratio $\alpha_2 c^2/2$ on the distance $c^{3/2}\varepsilon$ for $q_c/q_r = 1$.

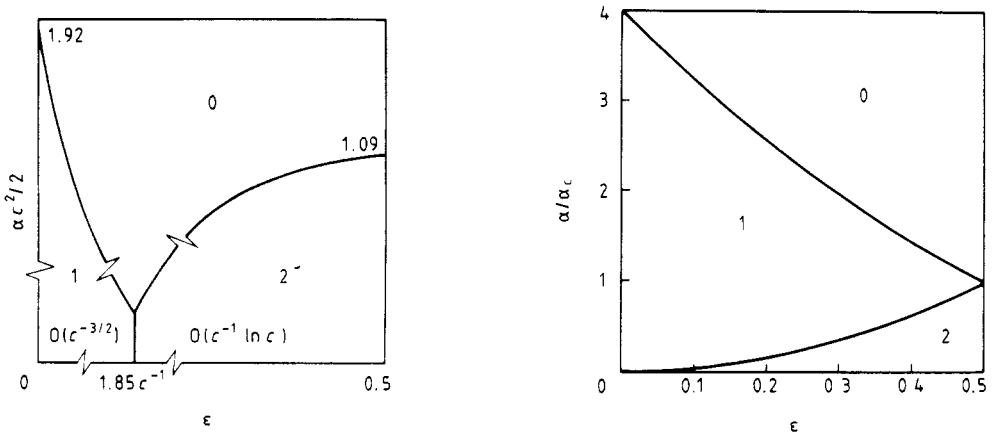

Figure 11. (*a*) A schematic phase diagram in the space of pattern distance and storage ratio for storing two correlated patterns among other random patterns. The phases 0, 1 and 2 are respectively the non-retrieval, non-distinguishing and retrieval phases. (*b*) The corresponding phase diagram for the dilute asymmetric Hopfield–Little model [16].

when too many patterns are stored, the system fails to retrieve; when the stored patterns are fewer, the system is capable of retrieving, but fails to distinguish patterns that are too close together.

For comparison, the phase diagram for the Hopfield model [16] is shown in figure 11(*b*). We immediately notice differences in the relative positions of the three phases. The non-distinguishing phase is present for all distances between the two correlated patterns in the Hopfield system, but is present only for small distances in the present model. This is because, in our system, information about a pattern is stored locally at a neighbourhood of sites and so, when considerably different patterns address different neighbourhoods of sites, no confusion arises. By contrast, the output of the Hopfield model is the sum of a signal term and a noise term [4], whose interference effect is less dependent on the Hamming distance.

These comparisons between the two models lead us to conclude that the Boolean model, with the low training-noise temperature in its training-with-noise algorithm considered in this paper, is a 'short-ranged model, in the sense that information about patterns is stored in localised neighbourhoods within each node. Interference between considerably different patterns is minimal, but the basins of attraction are relatively small. On the other hand, the Hopfield model is a long-ranged model, in the sense that information about patterns is stored distributively. Interference between patterns is always present, but the basins of attraction are relatively large (see also figure 5).

To illustrate this comparison concretely, suppose we fill a site in the ith node of the Boolean model in Ising representation with the number

$$\operatorname{sgn} \sum_j J_{ij} S_j$$

where the J_{ij} are the coupling strengths of the Hopfield model with the same topology, and $\{S_j\}$ is the address of the site. The dynamics of this network is then exactly that of the Hopfield model. Since information about a pattern in the Hopfield model is embedded in the coupling strengths, it is clear that information storage is 'delocalised' among all the sites of a node, in contrast to the 'localised storage' of the algorithms

discussed in this paper. This apparently accounts for the different properties of the two systems.

The high efficiency of the Aleksander network for some hard learning problems [10], is also a consequence of the localised manner of storing information. In the parity problem, say, complementary bits must be stored in nearest-neighbouring sites of each node. Learning information site by site is therefore the most natural way.

10. Conclusion

We have performed a statistical analysis of the learning and retrieval properties of randomly connected Boolean neural networks as applied to pattern storage and recognition. In addition, we have examined various features introduced by Aleksander to make a Boolean network function as an associative memory, including the u state, the training with noisy examples and the correction step.

The u state was believed to embody three advantages. First, it randomises the choice of the next site when an ambiguity is experienced in a retrieval process. This reduces the possibility of cyclic dynamics which prevent the retrieval of correct patterns. However, this is essentially irrelevant for networks of low enough connectivity, for which the annealed approximation is valid. Second, it ensures an unbiased initial state, so that no pre-stored unfavourable information slows down the learning procedure. This is concerned with the dynamics of learning and will be discussed elsewhere [17]. Third, three-state systems were believed to store extra information with the intermediate u state. However, since we have found that the system improves its storage capacity with the decrease of q_c/q_r, and hence the fraction of u sites, we have some reservations with the usual argument that the possibility of outputting u states invariably improves the system performance. This is explained below.

To be sure, the probability of a site in a u state is, according to the proximity rule,

$$\frac{(q_c/q_r)n_A n_{\bar{A}}}{n_A^2+(q_c/q_r)n_A n_{\bar{A}}+n_{\bar{A}}^2}$$

which is maximum at $n_A = n_{\bar{A}}$. In other words, a node is more likely to respond with a 'don't know' when the information derived from its training is ambiguous. However, this advantage can be overshadowed by the presence of other undesirable 'don't know' responses resulting from the training procedure. Our study of the majority rule algorithm has revealed that it is always better to fill a site with the majority bit, if there is any, than with a 'don't know'. The training-with-noise procedure, according to the proximity rule, inevitably fills a site having a majority bit with some 'don't know' responses, and these undesirable 'don't know' responses reduce the storage capacity to a value below that of the $q_c/q_r = 0$ system. For $q_c/q_r > 2$, the proportion of u sites is so large that the storage capacity is even less than that of the corresponding two-state system.

We believe that within the training-with-noise scheme, three-state systems are able to have a larger storage capacity because the introduction of the intermediate state makes the sites more likely to store the majority bit. This can be seen by considering how a majority bit is disrupted to become the minority bit during the training procedure. It takes two consecutive switches in the state to complete the disruption in three-state systems, whereas only one is required in the two-state system, a fact which is reflected

in the ratio $P_+/P_- = (n_A/n_{\bar{A}})^2$ *and* $n_A/n_{\bar{A}}$ in three-state and two-state systems respectively (independent of the ratio q_c/q_r in three-state systems). Thus, provided that the fraction of u sites, which is determined by the ratio q_c/q_r, is not too large, the noise-trained three-state systems will have a higher storage capacity than the corresponding two-state system. Indeed, we have shown that three-state systems with $q_c/q_r < 2$ have a consistently higher storage capacity than the analogously noise-trained two-state system.

However, we have shown that the overall improvement in the storage performance (after very long training) is at best only marginal when we introduce the intermediate state.

We have also demonstrated the virtue of the training-with-noise algorithm. To enable the network to function as an associative memory, information about a pattern has to be diffused into the neighbourhood of the appropriate sites, which is achieved in our case by the noisy training steps. In fact, as shown in the single pattern example in § 3 and will be fully discussed elsewhere [17], systems with appropriately high training-noise temperatures take fewer steps to attain associative memory. By contrast, systems with vanishingly low levels of training noise take increasingly more steps to train, and systems with zero training-noise temperatures never learn.

However, a finite training noise also has a disadvantage: it inevitably causes disruption and the storage capacity after very long training is reduced.

We have also studied the effects of introducing the correction step, which was believed to facilitate an iterative improvement during training. By varying the parameter q_c, we have arrived at the apparently surprising conclusion that systems which are more reluctant to correct (i.e. more stubborn) have a higher storage capacity. This is because after very long training, systems which are too easy to correct tend to confuse themselves with too many 'don't know' responses.

However, stubborn networks do pay a price. As discussed elsewhere [17], systems with vanishingly small q_c take more steps to attain their ideal storage capacities at equilibrium, and systems with $q_c = 0$ never attain their ideal performance.

The majority rule algorithm, which has a higher storage capacity than any noise-trained systems discussed so far, reveals a weakness of the training-with-noise procedure, in that noise-trained systems never attain the optimal storage. Even the best of them, the one with a vanishingly small q_c and a vanishingly low training-noise temperature after very long training, has a finite probability of outputting the minority bit at a site, according to the proximity rule. Fortunately, the difference in storage capacity in this case is only 2%.

We have seen that, in common with the Hopfield model, the present model exhibits a memory threshold, beneath which it can store with only small errors, but at which it experiences a memory catastrophe. Furthermore, the radius of attraction depends on the number of stored patterns in a qualitatively similar way to the stability K in the perceptron.

Differences exist between the two models, though. These differences can be traced to the short-ranged nature of the low training-noise Boolean model in contrast to the long-ranged Hopfield model. Since information about a pattern is stored locally in the sites, interference between patterns is minimal in the Boolean model. Consequently, the attractor of a stored pattern is simple (for at least one type of dynamics), and the non-distinguishing phase of two correlated patterns is greatly reduced.

Concerning the storage capacity of the network, the training-with-noise algorithm enables it to store roughly $(2/c^3)2^c$ patterns per bond, though the majority rule can even store slightly more. This capacity is much larger than the corresponding Hopfield

net, where $2/\pi$ patterns per bond can be stored, indicating that Boolean nets have a great potential as neural memories.

In terms of the number of stored patterns per bit of information, the Boolean model gives a value of $2/c^2$ compared with 0.1 for the Hopfield model with clipped synapses [22]. This relatively low performance is apparently due to the relatively few bits of stored information used in one retrieval step. On the other hand, however, this very disadvantage can be a technical merit at the same time, since hardware implementation may be easier with the subsequently fewer wirings.

To be fair, it may not be possible to say which is the better model, since a point-by-point comparison is likely to indicate that one is better in some areas of application but worse in others. In particular, by limiting our discussion to associative memories, we have not considered the well known advantage of Boolean nodes in dealing with linearly non-separable functions. It will therefore be interesting to study the Boolean model further in order to reveal its strengths and weaknesses. Various generalisations of the Hopfield model have been studied, including alternative learning rules [23], storage of correlated patterns [24], storage of pattern sequences [25] and many others. It would be exciting to explore and compare the capabilities of the Boolean models in at least some of these aspects, and to look for areas that utilise their unique characteristics.

Dedication and acknowledgments

We dedicate this work to the memory of Elizabeth Gardner, who stimulated us and others with her demonstrations of the application of statistical mechanics to neural networks and spin glasses. We acknowledge valuable discussions with Elizabeth and also with I Aleksander, W K Kan, B Derrida, J Shapiro and M Moore. This work was supported financially by the Science and Engineering Research Council of the United Kingdom, and was initiated at the CECAM Workshop on Complex Optimization and Stochastic Computing (Orsay, September 1987).

Appendix: Retrieval of two correlated patterns of distance $\varepsilon \sim O(c^{-1})$

We shall derive the relation (9.20) in this appendix. In the regime of very low storage ratio, the retrieval function $g(x, y)$, equation (9.8), approaches that for the storage of two correlated patterns in the absence of other patterns. The fixed point y^* is therefore that of the stable fixed point of (5.6). The transition from retrieval to non-retrieval takes place when the coefficient of x in $f(x, y^*)$ is essentially 1. According to (9.7), this coefficient is

$$c \sum_{s,t} \frac{\exp(-c\varepsilon y^*)}{s!} (c\varepsilon y^*)^s \frac{\exp[-c\varepsilon(1-y^*)]}{t!} [\varepsilon(1-y^*)]^t (b(s+1, t+1) - b(s, t)]. \tag{A1}$$

The disruptions $b(s, t)$ in this low storage regime for $q_c/q_r = 1$ is given by

$$b(s, t) = \begin{cases} \dfrac{\alpha}{4} \dfrac{c^{\min(s,t)}}{\min(s, t)!} & \text{for } \min(s, t) < r_0 \text{ and } s \neq t \\ \dfrac{\alpha}{7} \dfrac{c^s}{s!} & \text{for } s = t < r_0 \\ \frac{1}{2} & \text{for } \min(s, t) \geqslant r_0 \end{cases} \tag{A2}$$

where r_0 is the average site-distance between nearest neighbouring patterns, and is related to the storage ratio α by

$$\alpha \simeq \left(\frac{r_0}{ec}\right)^{r_0}. \tag{A3}$$

The coefficient (A1) can therefore be written as the sum of three terms:

$$\frac{\alpha c}{4} e^{-c\varepsilon} \sum_{s<t, s<r_0} \frac{(c\varepsilon y^*)^s}{s!} \frac{[c\varepsilon(1-y^*)]^t}{t!} \frac{c^{s+1}}{(s+1)!}$$

$$+\frac{\alpha c}{4} e^{-c\varepsilon} \sum_{t<s, t<r_0} \frac{(c\varepsilon y^*)^s}{s!} \frac{[c(o(1-y^*)]^t}{t!} \frac{c^{t+1}}{(t+1)!}$$

$$+\frac{\alpha c}{7} e^{-c\varepsilon} \sum_{s<r_0} \frac{(c\varepsilon y^*)^s}{s!} \frac{[c\varepsilon(1-y^*)]^s}{s!} \frac{c^{s+1}}{(s+1)!}. \tag{A4}$$

Consider first the last term in (A4), which can be written as

$$\frac{\alpha c}{7} e^{-c\varepsilon} \int_{-\pi}^{\pi} \frac{d\theta}{2\pi} \int_{-\pi}^{\pi} \frac{d\phi}{2\pi} \int_{-\pi}^{\pi} \frac{d\psi}{2\pi} e^{-i\theta} \frac{1-\exp(ir_0\psi)}{1-\exp(i\psi-0^+)}$$

$$\times \exp[c\varepsilon y^* e^{-i\psi-i\phi-i\theta} + c\varepsilon(1-y^*) e^{i\phi} + c e^{i\theta}]. \tag{A5}$$

Performing the integration over θ using the method of steepest descent, we have

$$\frac{\alpha c}{7} e^{-c\varepsilon} \int_{-\pi}^{\pi} \frac{d\phi}{2\pi} \int_{-\pi}^{\pi} \frac{d\psi}{2\pi} \frac{1-\exp(ir_0\psi)}{1-\exp(i\psi-0^+)} \exp(\tfrac{3}{4}i\psi + \tfrac{3}{4}i\phi)$$

$$\times \frac{\exp[c\varepsilon(1-y^*) e^{i\phi} + 2\sqrt{c^2\varepsilon y^*}\, e^{-i\psi/2-i\phi/2}]}{[4\pi c(\varepsilon y^*)^{3/2}]^{1/2}}. \tag{A6}$$

Another integration over ϕ, again using the method of steepest descent, yields

$$\frac{\alpha c}{7} e^{-c\varepsilon} \int_{-\pi}^{\pi} \frac{d\psi}{2\pi} \frac{1-\exp(ir_0\psi)}{1-\exp(i\psi-0^+)}$$

$$\times \frac{\exp(\frac{2}{3}i\psi) \exp[3c^3\sqrt{\varepsilon^2 y^*(1-y^*)}\, e^{-i\psi/3}]}{2\pi\sqrt{3}\, c[\varepsilon^2 y^*(1-y^*)]^{2/3}} \tag{A7}$$

and a final use of the method of steepest descent allows us to write the expression as

$$\frac{\alpha c\sqrt{r_0}\, e^{-c\varepsilon}}{7(2\pi)^{3/2} c^2\varepsilon^2 y^*(1-y^*)} \left(\frac{e^3 c^3 \varepsilon^2 y^*(1-y^*)}{r_0^3}\right)^{r_0}. \tag{A8}$$

Similar calculations show that the first two terms of (A1) are of the order r_0^{-1} higher, and hence negligible. The transition from retrieval to non-retrieval, given by (9.20), is obtained by putting (A3) into (A8), which is then equated to 1.

References

[1] Hopfield J J 1982 *Proc. Natl Acad. Sci. USA* **79** 2554
[2] Little W A 1974 *Math. Biosci.* **19** 101
[3] Amit D J, Gutfreund H and Sompolinsky H 1987 *Ann. Phys., NY* **173** 30
[4] Bruce A D, Gardner E J and Wallace D J 1987 *J. Phys. A: Math. Gen.* **20** 2909
[5] Gardner E, Derrida B and Mottishaw P 1987 *J. Physique* **48** 741
[6] Gardner E 1987 *Europhys. Lett.* **4** 481
[7] Hinton G E and Sejnowski T J 1986 *Parallel Distributed Processing* vol 1, ed J L McClelland and D E Rumelhart (Cambridge, MA: MIT Press) p 282

[8] Rumelhart D E, Hinton G E and Williams R J 1986 *Parallel Distributed Processing* vol 1, ed J L McClelland and D E Rumelhart (Cambridge, MA: MIT Press) p 318
[9] Shapiro J and Moore M 1987 private communication
[10] Aleksander I 1988 *Neural Computing Architectures* ed I Aleksander (London: Kogan Page) p 133
[11] Paternello S and Carnevali P 1987 *Europhys. Lett.* **4** 503
[12] Carnevali P and Paternello S 1987 *Europhys. Lett.* **4** 1199
[13] Wong K Y M and Sherrington D 1988 *Europhys. Lett.* **7** 197
[14] Kauffman S A 1969 *J. Theor. Biol.* **22** 437
[15] Derrida B and Pomeau Y 1986 *Europhys. Lett.* **1** 45
[16] Derrida B, Gardner E and Zippelius A 1987 *Europhys. Lett.* **4** 167
[17] Wong K Y M and Sherrington D 1989 The dynamics of learning in Boolean neural networks, to be published
[18] Gardner E, Stroud N and Wallace D J 1987 *J. Phys. A: Math. Gen.* **22** 2019
[19] Gardner E and Derrida B 1989 *J. Phys. A: Math. Gen.* **22** 1975, 1983
[20] Gardner E 1988 *J. Phys. A: Math. Gen.* **21** 257
[21] Krauth W and Mézard M 1987 *J. Phys. A: Math. Gen.* **20** 745
[22] Sompolinsky H 1987 *Proc. Heidelberg Colloq. on Glassy Dynamics and Optimization (June 1986)* ed J L von Hemmen and I Morgenstern (Berlin: Springer) p 485
[23] Personnaz L, Guyon I and Dreyfus G 1985 *J. Physique Lett.* **46** 1359
[24] Amit D J, Gutfreund H and Sompolinsky H 1987 *Phys. Rev.* A **35** 2293
[25] Sompolinsky H and Kanter I 1986 *Phys. Rev. Lett.* **57** 2861