Exponential storage and retrieval in hierarchical neural networks

J. Phys. A: Math. Gen. 22 (1989) 4707-4728. Printed in the UK

Exponential storage and retrieval in hierarchical neural networks

Charles R Willcox

Rosemount Incorporated, Aerospace Division and the Solid State Technology Center, M/S-A36, 12001 Technology Drive, Eden Prairie, MN 55344, USA

Received 8 December 1988

Abstract. A hierarchical neural network model capable of storing and retrieving an exponential number of states is introduced. Formulated on a spin glass analogy, the network spins (neurons) are organised into a multitier cluster hierarchy such that, for an N-spin sytem, the number of stored states grows exponentially with N. Relaxation occurs at zero temperature by what is essentially a tunnelling process and can be implemented using either a bottom-up or top-down updating procedure. As a result of the encoding prescription, the stored states are highly correlated and can be embedded within an ultrametric topology. The information capacity is determined, as well as the model's ability to content-address its stored memory patterns. Numerical simulations illustrating the operation and effectiveness of various hierarchical systems are also presented.

1. Introduction

Neural network models belonging to the Hopfield [1] and Little [2] class are assemblies of binary-state elements ('neurons') that interact collectively through some form of coupling matrix. These models, which have their origins in the earlier work of McCulloch and Pitts [3], exhibit remarkable computational properties and may be useful as biological memory paradigms. When viewed as models of memory [4, 5] such networks are, however, limited in the number of states they can reliably store. Recently, this particular class of neural networks has benefited from its similarity to spin glasses, which during the past few years, have become relatively well understood [6]. Motivated further by this similarity, we demonstrate that a substantial increase in storage capacity is possible when the stored states are correlated in a way that parallels the hierarchical configuration of states found in spin glasses.

The lack of long-range order in spin glasses, even at low temperatures, is the result of random coupling between spins and gives rise to an exponential number of metastable states [7]. It was Mézard *et al* [8] who showed that these states are hierarchically arranged, i.e. the space of spin glass states exhibit an ultrametric topology. The possibility of constructing 'ordered' spin glass states within neural network systems for use as associative memory devices has been discussed by several authors [9, 10]. In particular, our work was inspired by an attempt to implement the scheme of Dotsenko [9].

The first part of this paper begins with a brief review of the Hopfield neural network model, noting in particular its ability to store states and its overall capacity to store information. Next, we introduce a hierarchical model originally proposed by Dotsenko

[9], but now modified to function at zero temperature (i.e. in the absence of external stochastic noise). In this model, the spins (neurons) are grouped into clusters forming a hierarchy which ultimately governs the interaction amongst the spins and indirectly defines the correlation structure of the stored states. After determining the optimum cluster size which maximises the total number of states, we then examine the information capacity of the stored states, their ultrametric structure and content addressability. Also provided are several numerical examples illustrating the operation and effectiveness of the model. We conclude with a summary of our results and speculate on the possible biological importance of hierarchically organised neural networks.

2. The Hopfield model (±1 implementation)

Consider a set of N Ising spins or 'neurons' which can take on the values ± 1 and are interconnected by a connection strength matrix J_{ij} between the ith and jth spins. The state of the N-spin system at any given moment (t) is specified by the vector $\{s_i(t)\}$. The state of the system is allowed to change by following an *asynchronous* updating prescription. That is, we select at random a single spin $s_i(t)$ within the network and update its present state to a new state $s_i(t+1)$ using

$$s_i(t+1) = \operatorname{sgn}(X_i) \tag{2.1}$$

where X_i is the spin's local field defined by

$$X_i \equiv \sum_{j=1}^{N} J_{ij} s_j(t). \tag{2.2}$$

In this particular implementation it is assumed that the threshold for each spin is zero.

The Hopfield model can store or 'memorise' a given set of approximately orthogonal states $\{s_i^{(r)}\}$, $r = 1, 2, \ldots, P_H$ using the following Hebb [11] algorithm:

$$J_{ij} \equiv \begin{cases} \sum_{r=1}^{P_H} s_i^{(r)} s_j^{(r)} & i \neq j \\ 0 & i = j. \end{cases} \tag{2.3}$$

This definition allows the identification of an energy function E:

$$E(t) = -\tfrac{1}{2} \sum_{ij}^{N} J_{ij} s_i(t) s_j(t) \tag{2.4}$$

with the property that $\Delta E \equiv (E(t+1) - E(t))$ will be a monotonically decreasing function for each new update. Hence, the stored states create local 'basins of attraction' in the state-energy space of the network.

For each stored state, the corresponding basin of attraction will, in general, receive perturbations from the presence of the other stored states. The actual level of interference will depend on the total number of stores attempted. This effect is evident from a signal-to-noise analysis of the local spin field (2.2). Specifically, one finds for some memorised configuration r that the potential in this approximation is given by

$$X_i \approx (N-1) s_i^{(r)} \pm [(P_H - 1)(N-1)]^{1/2} \tag{2.5}$$

showing that, on average, the stored state is reinforced by a term of $\mathcal{O}(N)$ with a competing RMS interference noise term of $\mathcal{O}(NP_H)^{1/2}$.

The interference factor imposes a practical upper limit on the number of stored states possible using the Hebb algorithm (2.3). If the number of stored states exceeds N the network tends to saturate and lose its ability to reliably store images. In this saturated or 'chaotic' regime, the network is analogous to a spin glass and is now dominated by the spurious states which are admixtures of the original patterns. These states also serve as equally likely memory attractors and hence can interfere with the memory recall capability of the network. In the chaotic state, the model is essentially the Sherrington-Kirkpatrick [12] long-range interaction spin glass model which exhibits in the low-temperature regime an exponential number of metastable states. Bray and Moore [7] have shown that the number of metastable states increases exponentially with N.

At zero temperature, essentially perfect recall can be achieved in the Hopfield model provided the ratio of the number of stored vectors P_H to N (denoted by α) is much less than $(2\ln N)^{-1}$ where N is assumed to be large [13]. That is, when α is sufficiently small the energy minima correlated with the stored vectors represent global minima of the system. Amit *et al* [14] using the replica method [15], and other investigators [16] using computer simulations, have studied the effects of increasing α on the general storage capability of the network. In particular, they identify two phase transitions. The first transition occurs when the spurious states, corresponding to admixtures of several patterns, have energies lower than the energy minima associated with the nominal vectors being stored. As a consequence, bit errors begin to appear in some of the recalled vectors. A further increase in α brings on the second phase transition and the disappearance of any correlation between minima and nominal vectors. Amit *et al* [14] found that the two phase transitions occur at

$$\alpha_1^c = 0.051 \qquad \alpha_2^c = 0.138. \tag{2.6}$$

Replica symmetry breaking effects† will slightly modify these results. Numerical calculations performed by Amit *et al* [18] indicate that the replica breaking effects are indeed small, finding α_2^c to be 0.145. These results are also supported by recent rigorous results reported by Newman [19]. Up until now we have assumed N to be large. However, when N is small, finite-size effects can become important and will be significant for the individual clusters in the hierarchical model which follows. For a discussion on finite-size effects in the context of the Hopfield model, see, for instance, Wallace [16].

For more general storage prescriptions other than (2.3), the maximum storage capacity for random uncorrelated states which are stable against successive updates (2.1) approaches $2N$ [20]. However, in actual cases where α is large (e.g. 0.5 and 1.0) the content addressability apparently suffers [21]. Finally, we note that in the Hopfield model for the simple case when the states are uncorrelated and P_H is small enough that essentially no errors occur in the retrieved patterns, the information capacity of the network is $I_H = P_H N$ (see appendix 1, (A1.2)). In appendix 1 we also show that, if the patterns are constrained to a fixed magnetisation M_H, then the information capacity becomes

$$I_H = P_H \log_2\left(\frac{N!}{[\frac{1}{2}(N+M_H)]![\frac{1}{2}(N-M_H)]!}\right). \tag{2.7}$$

A more complicated expression applies when errors in the recalled patterns are allowed.

† An introductory discussion of replica symmetry breaking can be found in, e.g., [17].

3. The hierarchical model

The present hierarchical model is a modified version of a finite-temperature model originally proposed by Dotsenko [9] but altered to function at zero temperature. In this model, the intention is to construct an 'ordered' spin glass which permits in pinciple the memorisation of an exponential number of state vectors each correlated with a metastable state. Starting with N Ising spins, we partition the system into clusters such that the clusters at the mth level contain k_m subclusters of the next lowest level. This procedure produces a hierarchy of clusters terminating with the lowest level clusters, each of which contain k_0 spins. For an n-level hierarchy the total number of spins is then $N = k_0 k_1 \ldots k_{n-1}$.

At any given level m, there is then a *family* of clusters denoted by $\{\Omega_{a_m, a_{m+1}, \ldots, a_n}\}$ where the series of subscripts indicate the genealogy, tracing the chain of ancestor clusters starting with the nth topmost level, followed by $n-1$ and ending at the mth level cluster. That is, the subscript $a_m = 1, 2, \ldots, k_m$ labels the k_m clusters belonging to the mth level in the hierarchy. Figure 1 shows an example of an $n = 3$ level hierarchy of spin clusters containing a total of $k_0 k_1 k_2$ spins.

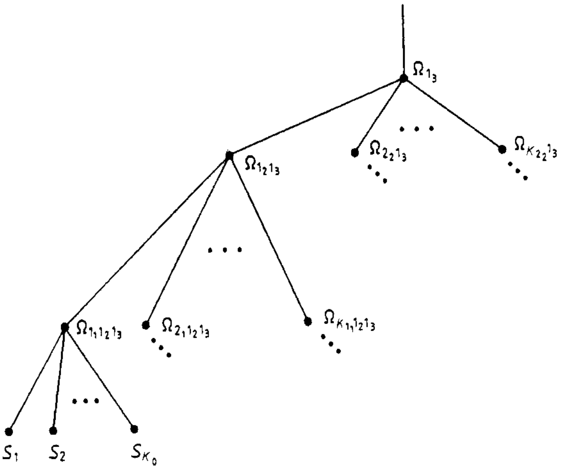

Figure 1. An $n = 3$ level hierarchical tree of spin clusters. The lowest ($m = 1$) level clusters each contain k_0 spins and the higher ($m > 1$) level clusters each contain k_{m-1} subclusters from the $m - 1$ level. Ellipses denote subclusters or spins not explicitly shown.

For each cluster, $\Omega_{a_m, \ldots, a_n}$ we introduce a corresponding mth level magnetisation defined by

$$M_{a_m, \ldots, a_n} \equiv \sum_{a_{m-1} \in \Omega_{a_m, \ldots, a_n}} M_{a_{m-1}, a_m, \ldots, a_n}. \tag{3.1}$$

When $m = 1$ the sum on the right-hand side of (3.1) is then simply over the *spins* within the cluster $\Omega_{a_1, \ldots, a_n}$. Finally, for each of these cluster magnetisations ($\neq 0$) we can also define an mth level ($m \geq 1$) effective Ising spin variable $\tilde{s}_{i_m}$ defined by

$$\tilde{s}_{i_m} \equiv \frac{M_{a_m, \ldots, a_n}}{|M_{a_m, \ldots, a_n}|} \tag{3.2}$$

where i_m is a short-hand notation for the cluster indices $a_m, \ldots, a_n$. We will often write $M_{a_m,\ldots,a_n}$ as simply M_m whenever the distinction between the individual mth level clusters is unimportant.

3.1. The storage prescription

In the hierarchical model, the interaction between spins is dependent upon the particular cluster that each spin happens to reside. Within *each* cluster, the storage of information is accomplished with an Hebbian algorithm as in (2.3). For instance, in each of the first level clusters, spins s_{i_0} within the clusters interact with each other through the following zero-diagonal connection matrix:

$$\begin{aligned} &J^{(0)}_{i_0 j_0} \equiv \sum_{r_0=1}^{p_0} s^{(r_0)}_{i_0} s^{(r_0)}_{j_0} \qquad i_0 \neq j_0 \\ &(i_0, j_0) \in \Omega_{a_1,\ldots,a_n} \end{aligned} \tag{3.3}$$

where the superscript r_0 identifies one of p_0 possible states to be stored within the cluster $\Omega_{a_1,\ldots,a_n}$. Note that in (3.3) $J^{(0)}_{i_0 j_0}$ represents one of N/k_0 *different* interaction matrices, one for each of the N/k_0 first-level clusters.

In general, the interaction of spins residing in different clusters, say clusters $\Omega_{a_m,a_{m+1},\ldots,a_n}$ and $\Omega_{b_m,a_{m+1},\ldots,a_n}$ will depend on the interaction of the corresponding mth level $(m \geqslant 1)$ effective Ising spins $\tilde{s}_{i_m}$ and $\tilde{s}_{j_m}$, respectively. Analogous to (3.3), the effective Ising spin interaction is mediated by an mth level zero-diagonal connection matrix defined by

$$\begin{aligned} &J^{(m)}_{i_m j_m} \equiv \sum_{r_m=1}^{p_m} \tilde{s}^{(r_m)}_{i_m} \tilde{s}^{(r_m)}_{j_m} \qquad i_m \neq j_m \\ &(i_m, j_m) \in \Omega_{a_{m+1},\ldots,a_n} \end{aligned} \tag{3.4}$$

where the superscript $r_m = 1, 2, \ldots, p_m$ identifies a particular mth level cluster state to be memorised (in terms of the effective spins $\tilde{s}^{(r_m)}_{i_m}$) and where p_m is the maximum number of different configurations of the mth level effective spins that we wish to store. Note that the total number of entries in all of the $J^{(m)}_{i_m j_m}$ is not greater than the total number of entries found in J_{ij} for the Hopfield model with an equivalent number of spins.

With the individual connection matrices defined, a corresponding cluster energy can be identified analogous to expression (2.4). By replacing J_{ij} and s_i in equation (2.4) with the appropriate values of $J^{(0)}_{i_0 j_0}$ $(J^{(m)}_{i_m j_m})$ and $s_{i_0}(\tilde{s}_{i_m})$, respectively, one obtains the energy expressions for the individual clusters.

3.2. An updating algorithm

Having established a method for storing states, we now describe a procedure for updating the network. Unless it is clear from context, we shall refer to states belonging to individual clusters as *cluster states* and states involving all N spins as *network states*. At zero temperature, Dotsenko's proposed model occasionally gets stuck in cluster states having magnetisations that do not necessarily place the upper level clusters into their lowest energy states. This problem limits the model's usefulness as a content-addressable machine†. In the present model we overcome this limitation by introducing what is effectively a tunnelling process.

When a cluster state is stored within the network via (3.3) or (3.4) its conjugate state defined by $\{s_{i_0}^{(r_m)}\} \to \{\bar{s}_{i_0}^{(r_m)}\} \equiv -\{s_{i_0}^{(r_m)}\}$ or $\{\tilde{s}_{i_m}^{(r_m)}\} \to \{\bar{\tilde{s}}_{i_m}^{(r_m)}\} \equiv -\{\tilde{s}_{i_m}^{(r_m)}\}$ is also stored, owing to the invariançe of $J_{i_m j_m}^{(m)}$ under interchange of i_m and j_m. Hence, for each cluster, either sign of the stored state magnetisation is possible. This flexibility allows one to arbitrarily specify the stored state of the next highest level parent cluster. As will be shown, when the network is relaxed, cluster states will be provided with the opportunity to selectively invert, or 'tunnel', to their conjugate form in order to locally minimise their energy, regardless of their level within the hierarchy.

Beginning with an arbitrary configuration of spins, the system can be relaxed using either a bottom-up or top-down updating sequence. In general, the mth level cluster states ($m \geq 1$) are determined by the sign of the mth level potential, viz

$$\tilde{s}_{i_m}(t+1) \equiv \operatorname{sgn}(X_{i_m}^{(m)}) \tag{3.5}$$

where

$$X_{i_m}^{(m)} \equiv \sum_{j_m} J_{i_m j_m}^{(m)} \tilde{s}_{j_m}(t) \qquad (i_m, j_m) \in \Omega_{a_{m+1},\ldots,a_n} \tag{3.6}$$

with $i_m (j_m)$ labelling the effective Ising spins corresponding to the clusters $\Omega_{a_m, a_{m+1},\ldots,a_n}$ $(\Omega_{b_m, a_{m+1},\ldots,a_n})$, respectively. Using (3.5), if the sign of the *effective* spin $\tilde{s}_{i_m}(t+1)$ changes, then all *spins* which are elements of the corresponding cluster under consideration also have their signs inverted.

To illustrate this procedure, consider an $n=2$ level hierarchy having $N=9$ spins and a fixed cluster size $k_0 = k_1 = 3$:

$$\begin{array}{ccc} (-1\ -1\ 1) & & (1\ -1\ 1) \qquad m=1 \\ (1\ -1\ -1)(-1\ -1\ 1)(-1\ 1\ 1) & \to & (-1\ 1\ 1)(-1\ -1\ 1)(-1\ 1\ 1) \qquad m=0 \end{array}.$$

Here we show the first, second level ($m=1$) effective spin being updated from -1 to 1 which in turn causes the first, first level ($m=0$) cluster state to invert to its conjugate form i.e. $(1\ -1\ -1) \to (-1\ 1\ 1)$. In the above, only the first level (bottom row) actually represents the state of the spins (neurons) themselves, whereas the upper level ($m=1$) carries information about the signs of the magnetisations belonging to each of the lower level cluster states.

In a similar way, the individual spins within the first level ($m=0$) clusters are updated according to

$$s_{i_0}(t+1) \equiv \operatorname{sgn}(X_{i_0}^{(0)}) \tag{3.7}$$

where

$$X_{i_0}^{(0)} \equiv \sum_{j_0} J_{i_0 j_0}^{(0)} s_{j_0}(t) \qquad (i_0, j_0) \in \Omega_{a_1,\ldots,a_n}. \tag{3.8}$$

† Unpublished results by the author. Going to finite temperatures should ameliorate this problem. However, whether this would be sufficient for useful memory recall operation remains to be demonstrated. Poor storage capacity was also reported by several researchers according to private communications cited by Gutfreund [10].

Our convention will be to leave the spins belong to an *m*th level cluster unchanged whenever the cluster potential is zero.

In practice, for top-down updating, a spin is selected at random, then beginning at the top of the hierarchy, the effective spin $\tilde{s}_{i_{n-1}}$ that belongs to the $n-1$ level cluster which contains the selected *spin* is updated using (3.5) and (3.6). If this effective spin changes sign then *all* spins which are members of this particular cluster have their signs inverted. This procedure is repeated as we move down to the next lower level until, at the $m=0$ level, we determine the sign of s_{i_0} through (3.7) and (3.8). For bottom-up updating, the above procedure is reversed.

3.3. *Storage capacity*

Using the Hebb algorithm (2.3) to store states in the Hopfield model necessarily favours uncorrelated states and limits the number of stable stored states to be linear in *N*. In contrast, the number of possible memories in the hierarchical model is exponential in *N*. To see how this comes about we observe that, whenever the state of a *cluster* is changed, an entirely new state of the *network* results. Hence, the total number of different *network* states N_t is simply the product of the number of different cluster states for each cluster in the hierarchy, that is,

$$N_t = p_0^{N/k_0} p_1^{N/k_0 k_1} \dots p_{n-1}. \tag{3.9}$$

The total in (3.9) neglects the conjugate network states which can also occur as stable states of the network and, if included, would double N_t. If we let $p \equiv p_0 = p_1 = \dots = p_{n-1}$ and maintain a fixed branching ratio by keeping the cluster size fixed at $k \equiv k_0 = k_1 = \dots = k_{n-1}$ then (3.9) simplifies to

$$N_t = p^{(N-1)/(k-1)}. \tag{3.10}$$

By setting $p = \alpha_k k$, where α_k is a fixed constant, the total number of states (3.10) can be maximised for appropriate values of the cluster size *k*. The result is

$$k = a_k^{-1} \exp(1 - k^{-1}) \tag{3.11}$$

which is approximately equal to $\alpha_k^{-1} e$ for *k* greater than 1. Hence, for a given value of α_k and *N*, the closest integer values of *k* determined from (3.11) and *n* consistent with $k^n = N$, will give the largest number of states. Once *k* has been determined, the closest practical integer value of $p(=\alpha_k k)$ can be identified. Computer simulations of hierarchies with various *p* and *k* values will be presented in § 4.

In the next section we provide an estimate of the fraction of errors expected in the network states following a signal-to-noise analysis. For now, we note that expression (3.9) reflects the total number of different spin configurations that can be stored within the network. Whether these patterns can be retrieved as stable network states, however, will depend on whether the individual cluster states are stable. In the hierarchical model, each *cluster* behaves like an independent Hopfield network, storing approximately orthogonal cluster states according to (3.3) and (3.4). Therefore, the stability of the cluster states, and hence the network states, will be determined by the same factors that constrain the Hopfield model. That is, the stability of the stored cluster patterns will depend on the ratio of p_m to k_m, the magnetisation value of the stored cluster states [22] and on finite-size effects.

Finite-size effects, for example, can be seen when only two vectors are stored in a three-neuron Hopfield network. In this simple case, even though $\alpha = \frac{2}{3}$, the network not only perfectly stores any two vectors but will perfectly retrieve these states as well. Recall that we use the convention that, whenever the potential evaluates to zero, the state of the corresponding spin is left unchanged.

Although a three-spin network will exactly store and recall any two vectors, if a random spin selection updating procedure is used, it will not always recall the *same* stored vector starting from the same initial spin configuration. This is because, in some cases, the initial state is equidistant, as measured by the number of spin flips, from either of the two stored states. Hence, the particular vector it relaxes to may depend upon which sequence of spins is selected. This indeterminacy can be minimised by increasing k_m while keeping p_m fixed.

3.4. *Perfect memory fraction*

A signal-to-noise analysis can be used to reveal qualitative estimates of the errors expected in the recalled network state vectors. Per cluster, both the signal and the noise term that occurs in the local spin field (3.6) or (3.8) will be the same as in the Hopfield model (2.5) only with $N \rightarrow k_m$ and $P_H \rightarrow p_m$. To begin, we ask what is the probability P_{c_m}, that, out of the $(p_m-1)(k_m-1)$ numbers, each ±1, arising in the non-signal contribution to $X_{i_m}^{(m)}$, their sum M_m, will add up to a number greater than k_m-1, the magnitude of the signal contribution? The probability of getting, say, q, $+1$ spins out of $(p_m-1)(k_m-1)$ is denoted by $B^{(q)}_{(k_m-1)(p_m-1),1/2}$ where

$$B^{(q)}_{Q,1/2} \equiv (\tfrac{1}{2})^Q \binom{Q}{q} \tag{3.12}$$

and where $\binom{Q}{q}$ is the binomial coefficient $= Q!(q!(Q-q)!)^{-1}$. If we have q, $+1$ spins then the sum M_m is simply $M_m = q - [(p_m-1)(k_m-1)-q] = 2q-(p_m-1)(k_m-1)$. Therefore, when $M_m = k_m - 1$, q takes the value $\frac{1}{2}p_m(k_m-1)$, so that P_{c_m} is then given by

$$P_{c_m} = \sum_{q=\frac{1}{2}p_m(k_m-1)}^{(k_m-1)(p_m-1)} B^{(q)}_{(k_m-1)(p_m-1),1/2}. \tag{3.13}$$

Since $(1-P_{c_m})^{k_m}$ represents the probability that a bit error will not occur within an mth level cluster state, then the probability that no error occurs within the entire network state is simply the product of these factors for each cluster in the hierarchy. We denote this by F_h for the perfect storage fraction:

$$F_h = \prod_{m=0}^{n-1} (1-P_{c_m})^{k_m N_{c_m}} \tag{3.14}$$

where N_{c_m} is the number of clusters in the mth level. The validity of (3.14) is checked for various numerical simulations in § 4.

3.5. *Information capacity*

In contrast with the large number of states, the total information I_h stored by the hierarchical network is actually less than in the standard Hopfield model. Letting $p \equiv p_0 = \ldots = p_{n-1}$ and $k \equiv k_0 = \ldots = k_{n-1}$ the information content of the hierarchical model is derived in appendix 1 (equation (A1.6)) with the result

$$I_h = \left(\frac{N-1}{k-1}\right) p \log_2(N_p) \tag{3.15}$$

where it is assumed that each cluster can store and retrieve without error, p states out of a choice of N_p per cluster. In general N_p should be simply 2^k. However, in practice N_p equals

$$k!\left(\left(\frac{k+M}{2}\right)!\left(\frac{k-M}{2}\right)!\right)^{-1}$$

and is less than 2^k because the choice of cluster states is constrained to non-zero values of the cluster magnetisations (see appendix 1, equation (A1.7)).

We can see that I_h will be less than I_H since $p < P_H$ and $N_p < 2^k$. Although each ultrametric state vector is distinct, its capacity to store information is reduced because it is comprised of blocks which are common to the other vectors whereas, in the Hopfield model, every bit comprising the state vectors potentially carries information. Finally, we note that in the limit $N_p \to 2^k$ and $k \to N$ (i.e. $n \to 1$) we have $I_h \to pN$, which is the same as found for the Hopfield model with $p = P_H$.

3.6. *The ultrametric structure of the stored states*

In the hierarchical model, the stored network states are correlated in such a way that they can be embedded within an ultrametric topology (see appendix 2 for the definition of an ultrametric space). We will show, however, that these states exhibit a different hierarchical structure than that of the original clusters.

This difference is best illustrated by a specific example. In figure 2(a) we show a simple two-level hierarchy with $N = 9$, $k = 3$ and $p = 2$. For this example, each of the first level clusters assume two states, each with magnetisation $+1$ and are labelled by the letters $A^{(r)}$ through $C^{(r)}$ with superscripts $r = 1$ or 2 identifying either of the two cluster states. The corresponding conjugate states are indicated by $\bar{A}^{(r)}$ through $\bar{C}^{(r)}$ and all have magnetisations equal to -1 (e.g. if $A^{(1)} = \{1-11\}$ then $\bar{A}^{(1)} = \{-11-1\}$). The two second level cluster states are labelled by $M^{(r)}$. For exactness we choose $M^{(1)} = \{1-11\}$, $M^{(2)} = \{-111\}$, so that, for example, the *network* states $\{A^{(1)}\bar{B}^{(1)}C^{(1)}\}$ and $\{\bar{A}^{(1)}B^{(1)}C^{(1)}\}$ yield the second level magnetisation states $M^{(1)}$ and $M^{(2)}$ respectively.

Figure 2(b) shows the resultant hierarchy of *network* states corresponding to the cluster hierarchy of figure 2(a). For each magnetisation state $M^{(1)}$ and $M^{(2)}$ there are eight corresponding network states. These states can be visualised, as shown in figure 2(c), as residing at the corners of a three-dimensional cube, arranged according to their cluster state coordinates. For general n-level hierarchies, this simple picture becomes increasingly more complex. In general, the cubes at each level would be hypercubes whose dimensionality would depend on the level (increasing as m decreases), the cluster size and the number of assigned memories per cluster.

On the cube, a measure of the relative separation between any two network states is given by their cluster state Hamming distance. For instance, when two network states are separated by a cube edge, they differ by only one cluster state; if they lie across from each other along a diagonal lying on one of the cube faces, they differ by two cluster states; and if the two states lie on a diagonal through the cube then all three cluster states are different. Moreover, network states lying on different cubes differ by at least two cluster states. Also, for every network state there is a corresponding conjugate state not explicitly shown in figures 2(a–c).

One can show using this simple example that, in this model, attempts to arrange the network states according to their cluster (or spin) state Hamming distances will

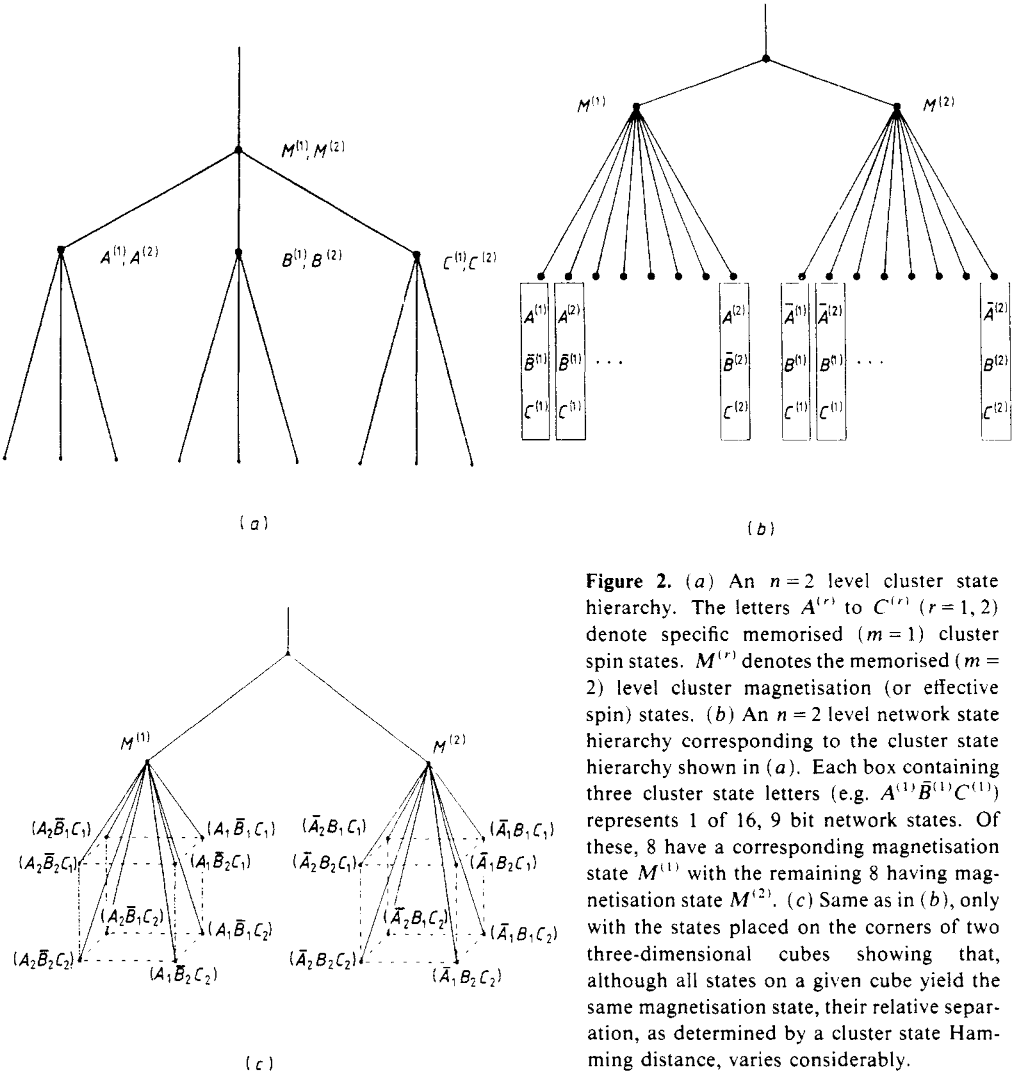

Figure 2. (*a*) An $n = 2$ level cluster state hierarchy. The letters $A^{(r)}$ to $C^{(r)}$ $(r = 1, 2)$ denote specific memorised $(m = 1)$ cluster spin states. $M^{(r)}$ denotes the memorised $(m = 2)$ level cluster magnetisation (or effective spin) states. (*b*) An $n = 2$ level network state hierarchy corresponding to the cluster state hierarchy shown in (*a*). Each box containing three cluster state letters (e.g. $A^{(1)}\bar{B}^{(1)}C^{(1)}$) represents 1 of 16, 9 bit network states. Of these, 8 have a corresponding magnetisation state $M^{(1)}$ with the remaining 8 having magnetisation state $M^{(2)}$. (*c*) Same as in (*b*), only with the states placed on the corners of two three-dimensional cubes showing that, although all states on a given cube yield the same magnetisation state, their relative separation, as determined by a cluster state Hamming distance, varies considerably.

fail to reveal an ultrametric structure. However, the network states are correlated according to their respective magnetisation states. The following distance function between any two network states $\{s^{(a)}\}$ and $\{s^{(b)}\}$ reveals this ultrametric property and is valid for an n-level hierarchy:

$$d(\{s^{(a)}\}, \{s^{(b)}\}) \equiv \sum_{m=0}^{n-1} d_m \tag{3.16a}$$

with

$$d_0 \equiv 1 - \delta\left(\sum_{i_0=0}^{N} (s_{i_0}^{(a)} - s_{i_0}^{(b)})\right) \tag{3.16b}$$

and for $m \geq 1$,

$$d_m \equiv 1 - \delta\left(\sum_{i_m} (\tilde{s}_{i_m}^{(a)} - \tilde{s}_{i_m}^{(b)})\right)$$

where $i_m \in \Omega_{a_{m+1},\ldots,a_n}$ for each of the $a_{m+1} = 1, 2, \ldots, k_{m+1}$ clusters and $\tilde{s}_{i_m}^{(a)}$ ($\tilde{s}_{i_m}^{(b)}$) represents the mth level effective spin for the state a (b) respectively. The function $\delta(x)$ is the usual delta function.

Clearly $d_0(d_m)$ is 0 whenever the spin (effective mth level spin) states are the same and 1 when they are different. For example, in the two-level hierarchy, different states under the same magnetisation state $M^{(1)}$ (or $M^{(2)}$) are all a distance $d = 1$ apart, while two states, one with magnetisation state $M^{(1)}$ and the other with $M^{(2)}$ are a distance $d = 2$ apart. It is also clear from the above definition that the total separation distance is zero when the two network states are identical.

As a final note, we comment that this embedding is hidden in the sense that one needs to know over which clusters to perform the sums. That is, looking only at the network states, without any knowledge of how the clusters were partitioned, it would be difficult to identify their ultrametric form.

3.7. Content-addressability

At the individual cluster level, the content-addressability of stored patterns should depend on the same factors that govern the content-addressability of stored patterns within equivalent-sized Hopfield networks. The likelihood of a successful cluster state retrieval, therefore, should be determined largely by k_m,† p_m and the overlap between the intended target pattern and initial configuration. The mth level cluster state overlap between a stored spin (or effective spin) state $\{\tilde{s}_i^{(r_m)}\}$ ($r_m = 1, 2, \ldots, p_m$) and a state $\{\tilde{s}_i^{(r_{\Gamma_m})}\}$ is defined by

$$\Gamma_m \equiv \frac{1}{k_m} \sum_{i_m=0}^{k_m} \tilde{s}_{i_m}^{(r_m)} \tilde{s}_{i_m}^{(r_{\Gamma_m})} \qquad i_m \in \Omega_{a_{m+1},\ldots,a_n} \tag{3.17}$$

where $\{\tilde{s}_i^{(r_{\Gamma m})}\}$ differs from $\{\tilde{s}_i^{(r_m)}\}$ in only $f_m = \frac{1}{2}k_m(1 - \Gamma_m)$ spins (or effective spins).

In addition to the cluster state overlaps, the overlap Γ between a stored *network* spin configuration $\{s_i^{(r)}\}$ ($r = 1, 2, \ldots, N_t$) and a spin configuration $\{s_i^{(r_\Gamma)}\}$ (differing in only $f = \frac{1}{2}N(1 - \Gamma)$ spins) is also important and is given by

$$\Gamma \equiv \frac{1}{N} \sum_{i=1}^{N} s_i^{(r)} s_i^{(r_\Gamma)}. \tag{3.18}$$

The crucial issue in the hierarchical model is how the collective behaviour of the clusters influences the global content-addressing properties of the network. From the last section, the stored network states were shown to be correlated according to an ultrametric rule, depending on both the magnetisation and individual spin states. This suggests that their content-addressability may likewise be dependent on a similar rule.

Numerical simulations of two-level hierarchies presented in the next section show that the content-addressability indeed depends on both the magnetisation and network state overlaps. The network shows a strong tendency to converge to final states having,

† Finite-size effects may complicate the situation when a stored cluster state is corrupted and k_m is very small, because there is an increased probability that the other stored cluster state(s) may share an equal or higher overlap with the corrupted state. For numerical simulations of large Hopfield networks see, for example, Forrest cited in [10].

in order of importance, the largest, or shared largest, second level ($m = 1$) magnetisation state overlap Γ_1, followed by the largest, or shared largest, network state overlap Γ with the initial patterns.

Simulations also show that the dependence on second level magnetisation states can be controlled, in part, by the ***magnitude*** of the first level ($m = 0$) stored state magnetisations M_0. The sensitivity to the magnetisation magnitude can be understood by considering what happens to a nominated pattern in a two-level hierarchy after randomly flipping f spins. This is equivalent to flipping, on average $f_0 = f/k_1$ spins in each first level cluster. (We place a bar over a quantity to denote its average value.) As a result, the first level cluster state overlaps Γ_0 will have an average value $\bar{\Gamma}_0$ equal to the network state overlap Γ. The second level ($m = 1$) overlap Γ_1, however, does not necessarily equal Γ. In fact, depending on f_0 and the magnitude of the stored cluster state magnetisations M_0, the value of Γ_1 can range from 0 to 1.

Specifically, when the number of first level cluster spin flips f_0 is less than $\frac{1}{2}|M_0|$, it is easy to show that the sign of M_0, and hence the effective spin, cannot change. If this holds for all of the first level clusters, then the second level magnetisation state is also not changed and the value of Γ_1 is 1. On the other hand, if the number of first level cluster spin flips exceeds $\frac{1}{2}|M_0|$, there is a chance that the sign of M_0 may change and thereby reverse the sign of the effective spin for that cluster. This only occurs if a disproportionate number of same-sign spins happen to be selected and reversed in sign. The greater the number of effective spin sign changes, the more corrupted the second level magnetisation state becomes and the lower the resultant overlap Γ_1 with the original pattern. Although we have only discussed what happens in a two-level hierarchy, similar arguments can be made for networks having an arbitrary number of levels.

Table 1. Tabulated values of $\bar{M}'_0$, $\sigma_{M'_0}$ and M'_0(max/min) for various Γ_0 values assuming $k_0 = 30$. The last column displays simulation results for a two level hierarchy having $p_0 = p_1 = 2$, $k_0 = 30$, $k_1 = 4$ and Γ_{initial} set equal to Γ_0, and shows qualitatively how the percentage recalled successfully depends on M_0.

Γ_0	f_0	M_0	$\bar{M}'_0$	$\sigma_{M'_0}$	M'_0 (max/min)	% recalled 'successfully'
$\frac{1}{5}$	12	2	0.4	5.45	2 ± 24	24
		8	1.6	5.26	8 ± 24	36
		16	3.2	4.62	16 ± 24	41
$\frac{1}{3}$	10	2	0.67	5.24	2 ± 20	27
		8	2.67	5.06	8 ± 20	46
		16	5.32	4.45	16 ± 20	62
$\frac{7}{15}$	8	2	0.93	4.92	2 ± 16	38
		8	3.73	4.75	8 ± 16	55
		16	7.47	4.17	16 ± 16	92
$\frac{2}{3}$	5	2	1.33	4.14	2 ± 10	48
		8	5.33	4.0	8 ± 10	79
		16	10.67	3.51	16 ± 10	100
$\frac{4}{5}$	3	2	1.6	3.34	2 ± 6	59
		8	6.4	3.22	8 ± 6	88
		16	12.8	2.83	16 ± 6	100
1	0	2	2	0	2 ± 0	100
		8	8	0	8 ± 0	100
		16	16	0	16 ± 0	100

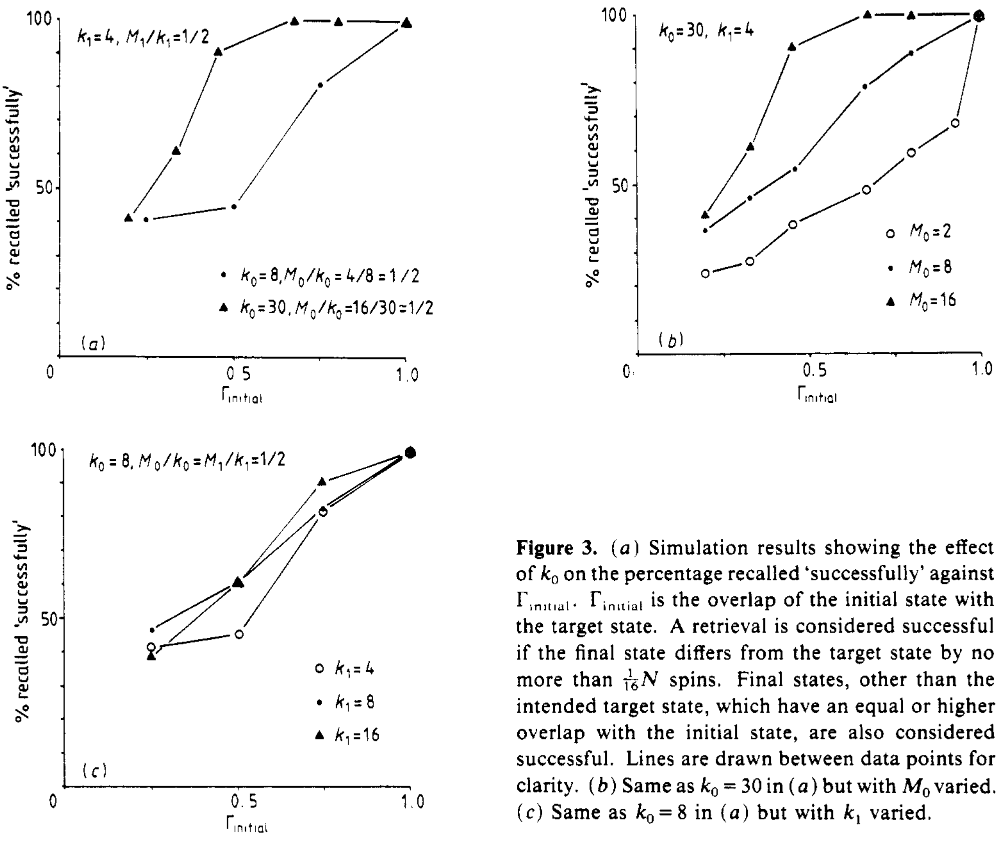

Figure 3. (*a*) Simulation results showing the effect of k_0 on the percentage recalled 'successfully' against $\Gamma_{\rm initial}$. $\Gamma_{\rm initial}$ is the overlap of the initial state with the target state. A retrieval is considered successful if the final state differs from the target state by no more than $\frac{1}{16}N$ spins. Final states, other than the intended target state, which have an equal or higher overlap with the initial state, are also considered successful. Lines are drawn between data points for clarity. (*b*) Same as $k_0=30$ in (*a*) but with M_0 varied. (*c*) Same as $k_0=8$ in (*a*) but with k_1 varied.

In appendix 3 we derive an expression for the probability of obtaining a cluster magnetisation, M'_m after f_m random spin flips (but never the same spin twice). Using this expression the average magnetisation $\bar{M}'_m$ and standard deviation $\sigma_{M'_m}$ after f_m spin flips can be evaluated. Table 1 gives the calculated results for f_0, $\bar{M}'_0$, $\sigma_{M'_0}$ and M'_0(max/min) for various Γ_0 and M_0 values assuming $k_0=30$. The last column gives numerical simulation results for the percentage recalled 'successfully' using top-down updating on a two-level hierarchy with $k_0=30$, $k_1=4$ and Γ initialised at the value Γ_0. These data only serve to illustrate qualitatively how M_0 affects the recall performance. At the time of pattern storage, the magnitude of the first level cluster state magnetisations were held to the values $M_0=2, 8$ and 16 while M_1 was fixed at 2. A recall is labelled successful if the final state differs from the intended target state in no more than $\frac{1}{16}N$ spins. These simulation data are also presented in figure 3(*b*) and reviewed in the next section.

In general, for a two-level hierarchy with top-down updating and small α_k, when $\sigma_{M'_0}$ is large compared to $\bar{M}'_0$, (e.g. when $\Gamma_0=\frac{1}{3}$ and $M_0=2$ in table 1), we expect the percentage of those recalled successfully to be low because of the unpredictability in the resultant magnetisation state overlap Γ_1 with the original pattern. When $\sigma_{M'_0}$ is less than or equal to $\bar{M}'_0$ (e.g. when $\Gamma_0=\frac{1}{3}$ and $M_0=8$ in table 1), we expect the probability of a sign change occurring in M_0 to be much less, resulting in a higher Γ_1 overlap value and a corresponding increase in the percentage recalled correctly. If f_0 is less than $\frac{1}{2}|M_0|$ (e.g. when $\Gamma_0=\frac{2}{3}$ and $M_0=16$ in table 1), then no sign change in M_0

will occur. In this case we would anticipate a high recall success rate. Of course, these expectations are only approximate because the network dynamics also depend on other factors such as k_m and α_k.

So far, the assumption has been that the updating is top-down. However, bottom-up updating could also have been used. The two implementations will differ somewhat in their content-addressing performance. When the updating is bottom-up, the first level cluster states have the first opportunity to correct some of their corrupted bits and possibly restore the sign of the cluster state magnetisation to its original value. At the next level up, there is then a higher probability that the second level magnetisation state will be less degraded, and therefore have an increased chance of relaxing to its uncorrupted form. This improvement is then carried into the next higher level and so on.

4. Numerical simulations

The effectiveness of the storage prescription and tunnelling relaxation algorithm was investigated for a total of ten different hierarchical networks. Table 2 summarises the results of these simulations. M_m is the mth level cluster magnetisation which was constrained to 1 (2) for k odd (even). We show in table 2 twice N_t, since the presence of the conjugate network states effectively doubles the number of different stable states that can occur during the simulations. For each of the 10 hierarchies, nominal cluster vectors were selected at random and stored using the storage prescriptions (3.3) and (3.4). Relaxation of the network always began with a randomly selected state vector and proceeded top-down according to algorithms (3.5)-(3.8). The updating ceased after a stable network state was found. This state was then checked to determine whether it correlated with any known member of the set of possible stored network states.

The number of independent simulations completed for each hierarchy was continued until either the total number of new stable states found reached the maximum possible for that given hierarchy (e.g. 2^5 and 2^6 for hierarchies 1, 2 and 7 respectively), or until it reached 2^7. This arbitrary upper limit was set because, for some of the hierarchies, the maximum number of different possible states approaches intractably high numbers, in one case as high as 2^{22} states.

Table 2. Summary of calculated and simulation results for various hierarchies.

H	k_m	M_m	p_m	n	N	$2N_t$	N_{ms}/N_{ss}	F_{h_c}	F_{h_a}
1	3	1	2	2	9	2^5	1	1	1
2	4	2	2	2	16	2^6	1	1	1
3	7	1	2	2	49	2^9	1	1	1
4	8	2	2	2	64	2^{10}	1	1	1
5	3	1	2	3	27	2^{14}	1	1	1
6	4	2	2	3	64	2^{22}	1	1	1
7	$k_0=30,\ k_1=4$	$M_0=M_1=2$	$p_0=p_1=2$	2	120	$2^2 2^4$	1	1	1
8	$k_0=30,\ k_1=4$	$M_0=M_1=2$	$p_0=3,\ p_1=2$	2	120	$2^2 3^4$	0.51	0.99	1
9	$k_0=30,\ k_1=4$	$M_0=M_1=2$	$p_0=6,\ p_1=2$	2	120	$2^2 6^4$	0.01	0.30	0.31
10	$k_0=30,\ k_1=4$	$M_0=M_1=2$	$p_0=15,\ p_1=2$	2	120	$2^2 15^4$	0	0	0

The quantity $N_{\mathrm{ms}}/N_{\mathrm{ss}}$ is the ratio of the number of recalled *memorised states* (starting from a randomly generated initial state), to the number of recalled *stable states*. Any deviation of $N_{\mathrm{ms}}/N_{\mathrm{ss}}$ from 1 implies that stable states were found which were not members of the set of possible stored states. As apparent from table 2, and as expected, when the number of stored cluster states was kept at 2 the hierarchical model *always* relaxed to one of the stored network states (i.e. $N_{\mathrm{ms}}/N_{\mathrm{ss}}=1$). That is, hierarchies 1-7 exhibted both perfect storage and perfect recall. Because N_t was relatively small for hierarchies 1, 2 and 7 each having 9, 16 and 120 spins respectively, we were able to run a sufficient number of simulations to completely exhaust the set of possible stored states, retrieving all of the stored states and *no* others, thus verifying the ability of the model to store and retrieve an exponential number of states.

In simulations 8, 9 and 10, the number of cluster state vectors was set at 3, 6 and 15 respectively. The first level cluster size was fixed at $k_0=30$ so that the corresponding effective cluster alphas were 0.1, 0.2 and 0.5 respectively. These three simulations allowed an analysis of how cluster state errors affect the overall final network state errors. In table 2, F_{h_c} (F_{h_a}) represents the calculated (actual) perfect memory fraction errors. All evaluated quantities shown are rounded to the second decimal place. The determination of F_{h_a} was based on simulations whose initial states were randomly selected from the known set of assigned states. In contrast, $N_{\mathrm{ms}}/N_{\mathrm{ss}}$ was determined from simulations that began with a randomly generated state and provides a measure of the network's ability to function as a content-addressable machine. As evident from table 2, good agreement was found between the expected perfect memory fractions calculated using (3.14) and the actual observed perfect memory fractions.

Hierarchy 10 shows that, when p_0 was set equal to 15 (i.e. $\alpha_0=0.5$), both $N_{\mathrm{ms}}/N_{\mathrm{ss}}$ and F_{h_a} were found to be zero. When p_0 was set to 6, (i.e. $\alpha_0=0.2$) as in hierarchy 9, $N_{\mathrm{ms}}/N_{\mathrm{ss}}$ was still essentially zero; however, F_{h_a} increased to 0.31. In other words, after being placed into one of its assigned memory configurations there remained a finite probability of the network persisting in that state following successive updates. Hierarchy 8 with p_0 set at 3 (i.e. $\alpha_0=0.1$), not only had a perfect memory fraction close to one but found to have a non-zero $N_{\mathrm{ms}}/N_{\mathrm{ss}}$ factor, thus indicating the content-addressability of some of its stored patterns.

To more carefully examine the content-addressability, several different two-level hierarchies were selected for study. For this analysis, the number of stored cluster states was always kept at 2 (i.e. $p_0=p_1=2$) to avoid any recall errors. However, the cluster size k_m and magnetisation magnitudes M_m were allowed to take on various values. For each simulation trial, a stored target network state $\{s_i^{(r)}\}$ was selected at random and a sufficient number of spins (each chosen at random) were flipped to achieve an initial overlap $\Gamma_{\mathrm{initial}}$ with the target state.

Because of the model's high storage capacity,. there may exist other stored states which share an equal or higher overlap with the corrupted state $\{s_i^{(r_I)}\}$. Relaxation to any of these equivalent stored states, or the target state, up to a maximum deviation of $\frac{1}{16}N$ spins would be considered a successful recall. This degeneracy of possible final states increases for small k_m and tends to bias the data, especially for small $\Gamma_{\mathrm{initial}}$.

The data shown represent the results of between 100 and 300 runs per each $\Gamma_{\mathrm{initial}}$ with the updating sequence always top-down. Although we do not show any explicit results, we found that bottom-up updating generally gave equivalent or better content-addressing performance. This was anticipated from the analysis provided in the last section. With only 100-300 runs per data point, the results presented can only be considered approximate, although the trends displayed are believed to be correct.

Figures 3(a–c) show how the percentage recalled successfully versus the initial overlap Γ_{initial} depends on the values of k_0, M_0 and k_1 respectively. In figure 3(a), k_0 took the values 8 and 30 while k_1 was held constant at 4 and the ratio M_m/k_m ($m=0, 1$) was maintained at approximately $\frac{1}{2}$. In both cases, the total number of network states, $2N_t$, was 64. There is a clear improvement in the percentage recalled successfully as k_0 is increased. The levelling off of the $k_0=8$ curve at small Γ_{initial} is most likely an artefact of the degeneracy alluded to earlier.

With k_0 and k_1 fixed at 30 and 4 respectively, figure 3(b) reveals the strong dependence on the first level ($m=0$) magnetisation magnitude. The performance improvement as M_0 is increased is in agreement with the analysis given in the last section. In contrast, with $k_0=8$ and $M_m/k_m=\frac{1}{2}$, ($m=0, 1$), figure 3(c) reveals no obvious trend as k_1 is varied. The percentage recalled successfully does not seem particularly sensitive to the size of the upper level cluster k_1 and perhaps more significantly, to the large differences in the number of stored states, with $2N_t$ having the values 262 144, 1024 and 64 for $k_1=16$, 8 and 4 respectively. More extensive simulations using larger k_1 values are needed to determine if this lack of sensitivity is general, or whether an optimum k_1 value exists.

A great deal can be learned about the role the magnetisation states play in the content-addressing process without deliberately setting the magnetisation state overlaps between initial and target patterns. From the simulations already performed, a comparison can be made of the magnetisation state overlap between both the initial and final states and the initial and intended target patterns. Figure 4(a) shows the percentage of cases that were found to have second level ($m=1$) initial/final magnetisation overlaps $(\Gamma_1)_{\text{final}}$ greater than or equal to the initial/target magnetisation overlaps $(\Gamma_1)_{\text{target}}$. The percentage is quite high (85% or better) and not particularly sensitive to the initial network spin state overlap Γ_{initial}. This indicates that the network has a strong tendency to iterate towards configurations having the largest magnetisation state overlap with the initial patterns.

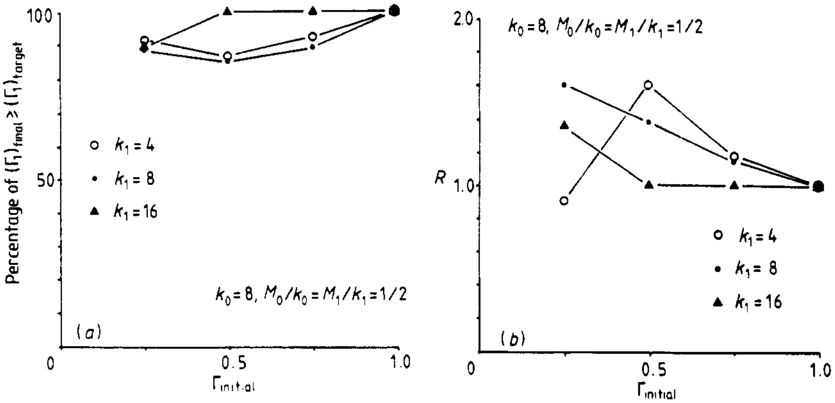

Figure 4. (a) Percentage of trials which had ($m=1$) level initial/final magnetisation state overlaps $(\Gamma_1)_{\text{final}}$ greater than or equal to the initial/(intended target) magnetisation state overlaps $(\Gamma_1)_{\text{target}}$. ($b$) Plot of R against Γ_{initial} as a function of k_1. See text for a complete definition of R. In conjunction with the results from (a), a value of R greater than one essentially means that the network prefers to content-address patterns according to an ultrametric rule involving both the magnetisation state and network state overlaps.

To assess the recall efficiency of stored network states belonging to the dominant magnetisation state, we form the ratio R of the percentage of successful recalls for states having $(\Gamma_1)_{\text{final}} = (\Gamma_1)_{\text{target}}$ (i.e. states remaining within a given magnetisation state), as a function of Γ_{initial}, to the percentage of successful recalls having no such restrictions as in figures 3(*a*–*c*). Figure 4(*b*) plots the ratio R as a function of Γ_{initial} for various k_1 values with k_0 at 8 and a constant ratio of M_m to k_m of $\frac{1}{2}$. In all cases (except at $\Gamma_{\text{initial}} = \frac{1}{4}$ for $k_1 = 4$) the value of R is greater than or equal to 1. Finite-size effects may be responsible for the R roll-off at $\Gamma_{\text{initial}} = \frac{1}{4}$ and $k_1 = 4$.

The combined results from figures 4(*a*) and 4(*b*) indicate that, at least in two-level hierarchies, the model prefers to content-address patterns by associations which respect the ultrametric embedding of the states. Although not demonstrated, we believe this property will carry over into general n-level hierarchies.

5. Discussion

We have provided numerical examples which show that a fully connected hierarchical neural network can both store and successfully recall an exponential number of ultrametrically correlated state vectors. Because of the large multiplicative effects, the individual cluster recall error probabilities must be kept low for useful operation. Implementation of the hierarchical model is straightforward and, if rendered in microcircuit form, would be useful in many associate recall applications.

The network is able to store an exponential number of states vectors because whenever any cluster changes its state, a new network state results. Each cluster operates like an independent Hopfield network, relaxing independently, except for possible state (conjugate state) inversions. The magnetisation state requirements of the upper level antecedent clusters determine whether or not a lower level cluster will invert. One can view the upper level parent cluster states as dynamically breaking the lower level offspring cluster state (conjugate state) symmetry to achieve a global reduction in energy. This global reduction in energy is different from that found in constraint satisfaction models† which seek a true single global minimum in the network energy. In the present hierarchical model, the network is 'globally' minimised if each cluster at every level has its energy locally minimised.

In spite of its ability to store a large number of states, the information capacity of the hierarchical model was shown to be actually less than that of the standard Hopfield model with an equivalent number of spins (neurons). However, in certain applications the ability to distinguish between a large number of different states may be of greater importance than the storage of a limited number of states, each having a high information content.

The content-addressability of stored input patterns within two-level hierarchies was investigated as a function of various model parameters. The model showed a strong preference to access states via an ultrametric path, whereby stored states having both the largest magnetisation and network state overlaps with the initial state had the highest probability of being retrieved. The attraction region surrounding each state could be increased by increasing the first level cluster size k_0 or increasing the magnitude of the first level cluster state magnetisation M_0. Increasing M_0 reduces the information capacity, but does not limit the storage capacity unless the desired number of stored patterns per cluster p is greater than N_p as given by equation (A1.7).

† Boltzmann machines are models of this type (see, e.g., [23]).

The hierarchical organisation of memorised patterns may have significance for neural networks belonging to multicellular animals. In human memory, the organisation of information is by classification according to recognised patterns and associations with previously stored memories. This superficially seems consistent with some form of hierarchical organisation. There is an interesting similarity between hierarchical cluster configurations and the size and organisation of dendritic bundles [24] (a general review can be found in [25]) located within the cortical column of mammals. From physiological observations, the number of neurons per bundle is suggested to be of the order of 50.

For the present hierarchical model, following (3.11), the most efficient utilisation of a given number of neurons is achieved when only two vectors are stored per cluster and the cluster size is kept close to 3. However, if we increase the number of stored vectors per cluster to 3, then the optimum cluster size having an approximately equal low recall error probability increases to about 50. In biological systems, a three-neuron cluster arrangement would probably be unacceptably sensitive to the random neuron failures which occur regularly throughout the lifespan of the organism. In addition, when a random neuron updating procedure is used, the relaxation process is not always deterministic and is exacerbated in clusters of small size. At the expense of increasing the total number of neurons, a 50-neuron cluster, on the other hand, would provide not only a more robust defense against random neuron failures, but also a more deterministic response to external stimuli.

Regarding hierarchical structure, one interpretation of bundle organisation might be to assume that within each bundle (cluster) a dedicated neuron serves as a summing node for all remaining neurons within the bundle. This neuron would then communicate its tailed information to other neurons, which are themselves located within different bundles. Contained within each of these new bundles is a specialised neuron which polls all its members relaying the result to yet another neuron in a different bundle and so on up the pyramid. This construction suggests that the basic wiring (bundle to bundle) is responsible, at least in part, for the memory function most likely related to long term memory, particularly that associated with information learned during the animal's infancy. in other words, the 'older' neuron connections serve predominantly as summing junctions carrying the oldest most general pieces of information while the 'younger' neuron links provide additional details by making up the remaining connections within each bundle.

In closing, we comment that although the hierarchical model in its present form can readily and usefully be implemented in many applications, several extensions remain to be explored. For instance, the content-addressability of patterns stored within hierarchies with more than two levels should be investigated. The effects of random synaptic potential fluctuations on the network (i.e. finite-temperature effects) should also be examined. Since random fluctuations actually occur in the synaptic emission process in real neurological systems, the effects of finite temperature on the network's performance would be of particular interest.

Appendix 1. Information capacity of the Hopfield and hierarchical models

We define the information capacity of an N-bit word stored in the Hopfield model to be related to the probability of that would occur by the random assignment of states to each of its N bits. Since each bit can take only one of two values, the probability of selecting an N-bit word is simply $P_w = (\frac{1}{2})^N$. The information capacity per word is then defined to be the log base 2 of one over P_w, that is,

$$\begin{aligned} I_w &\equiv \log_2\left(\frac{1}{P_w}\right) \\ &= N. \end{aligned} \tag{A1.1}$$

the use of log base 2 in I_w was to normalise the information content of one binary bit to unity. In (A1.1), I_w is the information content of a single word or pattern. Now, if each of the P_H patterns stored in the Hopfield model is independent and can be retrieved without error, then the total information capacity of the Hopfield model becomes

$$\begin{aligned} I_H &= P_H I_w \\ &= P_H N. \end{aligned} \tag{A1.2}$$

If the patterns are constrained to fixed magnetisations M_H, then the number of different states goes from 2^N to $\binom{N}{(1/2)(M_H+N)}$ (see equation (A1.7)). Hence, I_H reduces to

$$I_H = P_H \log_2\left(\frac{N!}{[\frac{1}{2}(N+M_H)]![\frac{1}{2}(N-M_H)]!}\right). \tag{A1.3}$$

Applying Stirling's approximation reduces (A1.3) to the same expression (6.1) found by Amit *et al* [22] with their $a = M_H/N$. When errors are allowed in the recalled states, the information content is further reduced and equation (A1.3) no longer applies. These effects are considered in [22].

To determine the information capacity of the hierarchical model we first need to determine the amount of information stored within each cluster. For simplicity, assume that $k = k_m$ and $p = p_m$ for all m and only p states out of a choice of N_p are stored per cluster. Since the probability of selecting one of N_p states is $P_{cw} = (1/N_p)$, then using the same convention as before, the information content per cluster per pattern I_{cw} is simply

$$\begin{aligned} I_{cw} &= \log_2\left(\frac{1}{P_{cw}}\right) \\ &= \log_2(N_p). \end{aligned} \tag{A1.4}$$

Because p patterns are stored per cluster and are independent (recall that each cluster behaves like an independent Hopfield network) then the information carried per cluster is

$$I_c = pI_{cw} \tag{A1.5}$$

where we assume that p is small enough that no errors occur in the recalled cluster states.

The total information capacity of the hierarchical network I_h is the sum of the information contained in all of the clusters. Since the total number of clusters in the network is given by $((N-1)/(k-1))$, the total information capacity becomes

$$I_h = \left(\frac{N-1}{k-1}\right) p \log_2(N_p). \tag{A1.6}$$

As indicated for I_H, a similarly more complicated expression results if we allow errors to occur in the retrieved cluster states.

For a given cluster of size k, we can determine the possible number of different k-bit vectors, each with the same magnetisation M. If the number of $+1$ spins in a vector of length k is q then the number of -1 spins is $(k-q)$. Hence, the magnetisation is $M = q-(k-q)$, implying that $q = \frac{1}{2}(M+k)$. Therefore, the number of vectors N_p which have magnetisation M is given by the binomial coefficient

$$N_p = \binom{k}{\frac{1}{2}(M+k)} \tag{A1.7}$$

where M must be even(odd) for k even(odd).

Appendix 2. Ultrametric spaces

An ultrametric space is a simple extension of a metric space. Recall that a metric space is a set X with a distance function d between any two points that obeys the following for any x, y and z in X:

$$\begin{aligned} &d(x,x)=0\\ &d(x,y)=d(y,x)\\ &d(x,z)\leqslant d(x,y)+d(y,z). \end{aligned} \tag{A2.1}$$

An ultrametric space is also a metric space (X, d) which has the following additional property:

$$d(x,z)\leqslant \max(d(x,y), d(y,z)). \tag{A2.2}$$

The ultrametric inequality (A2.2) implies that, for any three points x, y and z residing in an ultrametric space, of the three distances that occur between any two pairs, two of the distances will be equal and the third will always be less than or equal to the other two. A hierarchical tree structure naturally obeys this property where the distance between different end points at the bottom of the tree is measured by the height one has to ascend to find a common ancestor. For further discussion see references cited in [26].

Appendix 3. Cluster magnetisation probabilities

We wish to find the probability that a cluster of size k_m and initial magnetisation M_m will go to a new magnetisation M'_m after flipping f_m spins at random, but never the same spin twice. To evaluate this probability, we suppose that the cluster state contains $n_m^+ \equiv \frac{1}{2}(k_m + M_m)$, $+1$ spins and $n_m^- \equiv \frac{1}{2}(k_m - M_m)$, -1 spins (so that $n_m^+ - n_m^- = M_m$) and ask for the probability of finding f_m^+, $+1$ spins and $f_m^- = (f_m - f_m^+)$, -1 spins out of f_m random selections. This involves the product of two terms. The first term gives the number of arrangements containing exactly f_m^+, $+1$ spin selections and f_m^-, -1 selections out of f_m total selections, i.e. the binomial coefficient

$$\binom{f_m}{f_m^+}.$$

The second factor is the probability of actually realising any one of these different arrangements. Since we select spins at random but never the same spin twice one can show that the denominators (numerators) in this probability go like

$$\frac{k_m\,!}{(k_m-f_m)!}\left(\frac{n_m^+!}{(n_m^+-f_m^+)!}\cdot\frac{n_m^-!}{(n_m^--f_m^-)!}\right)$$

respectively. Therefore, the probability $P_f(f_m^+)$ of finding f_m^+ +1 spins in f_m selections is then

$$P_f(f_m^+)=\binom{f_m}{f_m^+}\left[\frac{(k_m-f_m)!}{k_m\,!}\,\frac{n_m^+!}{(n_m^+-f_m^+)!}\,\frac{n_m^-!}{(n_m^--(f_m-f_m^+))!}\right] \qquad \text{(A3.1)}$$

where it is assumed that $f_m^+\leqslant n_m^+$ and $f_m^-\leqslant n_m^-$.

Note that $P_f(f_m^+)$ is also the probability that the magnetisation goes from $M_m=n_m^+-n_m^-$ to M'_m where

$$\begin{aligned} M'_m &= (n_m^+-f_m^++f_m^-)-(n_m^--f_m^-+f_m^+) \qquad \text{(A3.2)}\\ &= M_m-2(2f_m^+-f_m). \end{aligned}$$

Using $P_f(f_m^+)$ and M'_m above, and summing over f_m^+, the average and standard deviation magnetisations $\bar{M}'_m$ and $\sigma_{M'_m}$ become respecively,

$$\bar{M}'_m=\sum_{f_m^+=0}^{f_m} P_f(f_m^+)[M_m-2(2f_m^+-f_m)] \qquad \text{(A3.3)}$$

$$\sigma_{M'_m}=\left(\sum_{f_m^+=0}^{f_m} P_f(f_m^+)\,.\,(\bar{M}'_m-[M_m-2(2f_m^+-f_m)])^2\right)^{1/2}. \qquad \text{(A3.4)}$$

By a different argument one can show that $\bar{M}'_m$ should equal $\Gamma_m M_m$, where $\Gamma_m=(1-2f_m/k_m)$. This allows a check on the validity of $P_f(f_m^+)$ through (A3.3). Indeed, after carrying out the sum in (A3.3) the two results agree.

References

[1] Hopfield J J 1982 *Proc. Natl. Acad. Sci. USA* **79** 2554; 1984 *Proc. Natl. Acad. Sci. USA* **81** 3088

[2] Little W A 1974 *Math. Biosci.* **19** 101
Little W A and Shaw G L 1978 *Math. Biosci.* **39** 281

[3] McCulloch W W and Pitts W 1943 *Bull. Math. Biophys.* **5** 115

[4] Hinton G E and Anderson J A (ed) 1981 *Parallel Models of Associative Memory* (Hillsdale, NJ: Erlbaum)

[5] Kohonen T 1978 *Associative Memory, A System Theoretical Approach* (Berlin: Springer); 1984 *Self-Organisation and Associative Memory* (Berlin: Springer)

[6] Binder K and Young A P 1986 *Rev. Mod. Phys.* **58** 801

[7] Bray A J and Moore M A 1980 *J. Phys. C: Solid State Phys.* **13** L469; 1981 *J. Phys. C: Solid State Phys.* **14** 1313

[8] Mézard M, Parisi G, Sourlas N, Toulouse G and Virasoro M 1984 *J. Physqiue* **45** 843; 1984 *Phys. Rev. Lett.* **52** 1156
Ioffe L B and Feigelman M V 1984 *J. Physqiue Lett.* **45** 475

[9] Dotsenko V S 1985 *J. Phys. C: Solid State Phys.* **10** L1017

[10] Parga N and Virasoro M 1986 *J. Physique* **47** 1857
Gutfreund H 1988 *Phys. Rev.* **A37** 570

[11] Hebb D O 1949 *The Organisation of Behavior* (New York: Wiley)

[12] Sherrington D and Kirkpatrick S 1975 *Phys. Rev. Lett.* **35** 1792

[13] McEliece R J, Posner E C, Rodemich E R and Venkatesh S S 1987 *IEEE Trans. Info. Theor.* **IT-33** 461
Weisbuch G and Fogelman-Soulie F 1985 *J. Physique Lett.* **46** L623

[14] Amit D J, Gutfreund H and Sompolinsky H 1985 *Phys. Rev. Lett.* **55** 1530; 1985 *Phys. Rev.* A **32** 1007
Gardner E 1986 *J. Phys. A: Math. Gen.* **19** L1047

[15] Kac M 1968 *Trondheim Theoretical Physics Seminar Nordita Publ. No* 286

[16] Wallace D J 1985 *Proc. Workshop on Lattice Gague Theory—A Challenge in Large Scale Computing, Wuppertal, 1986* eds B Bunk and K H Mutter (New York: Plenum)
Bruce A D, Canning A, Forrest B, Gardner E and Wallace D J 1986 *Neural Networks For Computing* (*AIP Conf. Proc.* **151**) eds R G Lerner and J S Denker (New York: AIP)

[17] Moorjani K and Coey J M D 1984 *Magnetic Glasses* (Amsterdam: Elsevier) pp 360-82

[18] Amit D J, Gutfreund H and Sompolinksy H 1986 Statistical mechanics of neural networks near saturation *Preprint* Hebrew University

[19] Newman C M 1988 *Neural Networks* **1** 223

[20] Cover T M 1965 *IEEE Trans. Elec. Comput.* **EC-14** 3, 326
Venkatesh S 1986 *Neural Networks for Computing* (*AIP Conf. Proc.* **151**) ed R G Lerner and J S Denker (New York: AIP) p 440
Komlos J and Paturi R 1988 *Neural Networks* **1** 239

[21] Forrest B M 1988 *J. Phys. A: Math. Gen.* **21** 245

[22] Amit D J, Gutfreund H and Somploinsky H 1987 *Phys. Rev.* A **35** 2293

[23] Ackley D H, Hinton G E and Sejnowski T J 1985 *Cogn. Sci.* **9** 147

[24] Mountcastle V B 1978 *The Mindful Brain* ed G M Edelman and V B Mountcastle (Cambridge, MA: MIT Press)

[25] Roney K J, Scheibel A B and Shaw G L 1979 *Brain Res. Rev.* **1** 225

[26] Baldi P and Baum E B 1986 *Neural Networks For Computing* (*AIP Conf. Proc.* **151**) ed R G Lerner and J S Denker (New York: AIP) p 35
Ramal R, Toulouse G and Virasoro M A 1986 *Rev. Mod. Phys.* **58** 765