Understanding hierarchical neural network behaviour: a renormalization group approach

1991 J. Phys. A: Math. Gen. 24 2655

(http://iopscience.iop.org/0305-4470/24/11/030)

J. Phys. A: Math. Gen. 24 (1991) 2655-2664. Printed in the UK

Understanding hierarchical neural network behaviour: a renormalization group approach

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 14 January 1991

Abstract. A hierarchical neural network model, presented in an earlier paper, is analysed using a renormalization group (RG) approach. The RG method puts many of the previously found empirical results on a firm theoretical foundation. The functional dependence of the propagation of errors from one level of the hierarchical tree to the next is derived and is shown to exhibit a phase transition. When the probability of entering errors at a given level exceeds some critical value, the error propagation is unbounded and will extend throughout the entire network, whereas below the critical value, the errors remain localized. This result along with individual cluster updating data is used to explain the content-addressability properties of the model.

1. Introduction

In an earlier paper [1], a hierarchical model was presented that was capable of storing and retrieving an exponential number of stored states. The model, with origins from an earlier one proposed by Dotsenko [2], is formulated on a spin-glass analogy, with its spins organized into a multi-tier cluster hierarchy such that for an N-spin network, the number of stored states is exponential in N. The network's ability to store an exponential number of states does not violate information capacity theorems [3] because the states are highly correlated, implying a low information content per state. Because of the large number of stored states, it is important to understand not only the network's ability to content-address its stored memory patterns but how these states are retrieved. Empirical results from simulation experiments gave partial clues to the recall process but a firm theoretical foundation remained to be found. The purpose of this paper is to take steps toward establishing a theoretical base through the application of renormalization group (RG) methods [4, 5].

The RG approach can be used because of the hierarchical model's inherent scale invariant properties. In general, systems possessing scale invariance are often found to exhibit critical behaviour [4, 5]. For the hierarchical model, this is indeed the case when we consider how errors propagate throughout the network. As will be shown, when the probability of entering errors at the base of the hierarchical tree exceeds some critical value, the errors no longer remain locally contained, but instead propagate up and invade the entire network. The static problem of the propagation of errors, when coupled with the recall probability of stored patterns for individual clusters, can be used to predict the content-addressability of network patterns involving all N spins. The method can satisfactorily explain the empirical content-addressability simulation results presented previously for the hierarchical model [1].

0305-4470/91/122655+10$03.50

The paper begins with a brief review of the hierarchical model with emphasis on its construction using a renormalization approach. Next, we establish a method for determining how far errors, when entered at the base of the tree, will propagate upwards. For the static case, which does not involve updating, the spread of errors displays critical behaviour having a critical exponent which is then calculated. After explicitly including updating results from a single cluster, we then show how the RG approach allows one to predict, assuming a nearest-neighbour approximation, the global pattern recall properties of the network. We conclude by comparing simulation data with theoretical predictions and then discuss the results.

2. The model

The model presented here is based on the hierarchical model introduced in [1]. In the present model the branching size k, however, is kept fixed to simplify the discussion, although the extension to arbitrary branching at each level is straightforward. The construction of the tree begins with a starting cluster or replication cell having k binary-state neurons, which we shall call spins, s_i where, $i = 1, 2, \ldots, k$ and can only assume the values ± 1. Coupling the k spins is a square storage matrix J_{ij} storing p, k-spin vector states $\{s_i^{(r)}\}$, $r = 1, 2, \ldots, p$ according to the usual Hebbian rule [6],

$$J_{ij} \equiv \sum_{r=1}^{p} s_i^{(r)} s_j^{(r)} \qquad i \neq j \tag{2.1}$$

with J_{ij} defined to be zero when $i = j$. Through (2.1) the matrix J_{ij} contains a total of $k(k-1)$ programmed entries, although only half of these are independent because of the symmetry under interchange of i and j. In addition, each of the stored cluster states are *constrained* to having a specified magnetization M given by

$$M \equiv \sum_{i=1}^{k} s_i. \tag{2.2}$$

Spins within a cluster are chosen at random and updated according to

$$s_i(t+1) = \operatorname{sgn}\left(\sum_{j=1}^{k} J_{ij} s_j(t) \right). \tag{2.3}$$

If $s_i(t+1)$ evaluates to zero, our convention is to leave the spin unchanged. The cluster, therefore functions like a Hopfield [7] net comprised of k spins and storing p, k-dimensional vector states having magnetization M.

Now consider a hierarchical tree network, of which the first two levels are shown in figure 1. The nth topmost node and its branches, represent a cluster having k effective spins, also denoted by s_i which are computed from the normalized magnetizations belonging to the states of k clusters that form the set of nodes and branches at the next lower $n-1$ level of the tree. That is

$$s_i = \frac{M_i}{|M_i|}$$

defines the effective spins, where the i index above, now refers to one of the k clusters at the $n-1$ level. Therefore, at each subsequent level one simply treats the spins

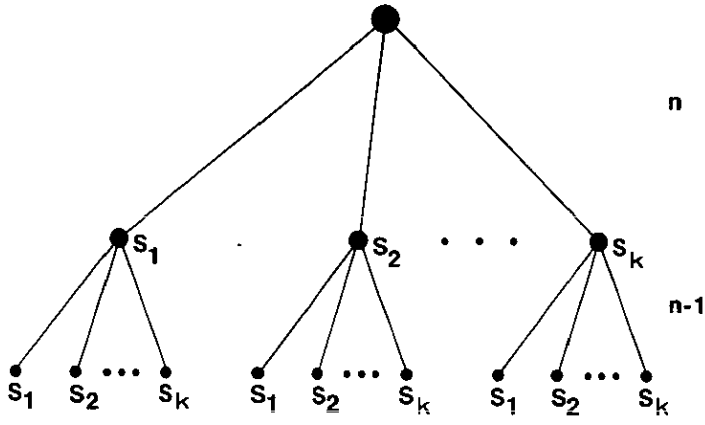

Figure 1. Graphical depiction of the first two levels of a hierarchical tree showing the nth topmost cluster and k, $n-1$ level clusters. Ellipses denote spins or clusters not explicitly shown.

belonging to a given cluster as effective spins which are computed from the normalized magnetizations belonging to the set of clusters occupying the next level down. The tree ends after reaching the first level or bottom, where the spins are now the actual network spins of the model.

All clusters regardless of their level in the tree are updated according to (2.3), with the cluster coupling matrices defined using (2.1), keeping in mind that the s_i represent effective spins unless we are at the bottom branch end points of the tree, where they represent network spins. To avoid notational clutter, we will not add special cluster identifying labels to each of the J_{ij} matrices or effective spins unless necessary.

Note that even though the spins are renormalized at each level as we advance from one level up to the next, the coupling matrices, however, are not rescaled proportionately to those occupying the previous level. Instead, each coupling matrix is programmed independently using (2.1). Updating any given cluster can be performed independent of the clusters, from which its effective spins are computed, because the symmetry (under $i \leftrightarrow j$) of the cluster coupling matrix allows for *either* sign of the cluster state magnetization, hence, the cluster effective spins can take on any value desired.

We should emphasize that the RG formalism has not been introduced for the purpose of reducing the dimensionality of the network. The intention is to show how the levels are inter-dependent on one another and as will be clear later, explain how errors propagate throughout the network.

To compute the number of stored network states, observe that for a two-level tree, there are a total of $N = k^2$ spins comprising a given network state. These spins form the bottom branch end points (first level) of the tree. There are k clusters at the first level and one cluster at the second and therefore two levels capable of storing information, so if each cluster stores p states, the total number of *network* states stored by a two-level model is $p \cdot p^k = p^{k+1}$. Generalizing to an n-level tree having $N = k^n$ spins and using the geometric progression, the total number of stored states, denoted by N_t is then

$$N_t = p^{(N-1)/(k-1)}. \tag{2.4}$$

Note the distinction between network states and cluster states: the former involves all

N spins of the network, whereas the latter involves only k spins belonging to a particular cluster.

The superscript factor $(N-1)/(k-1)$ in (2.4) is the total number of clusters present in the network. Multiplying this factor by the number of programmed J_{ij} matrix elements stored per cluster gives $k\cdot(N-1)$ for the total number in the hierarchical network which is *less* than that of a standard Hopfield model with N spins whose J_{ij} contains $N\times(N-1)$ programmed entries. This suggests that in spite of the greater number of stored states, the hierarchical model actually stores *less* information than a standard Hopfield model with an equivalent number of spins. The information content per stored state is low because the states are highly correlated. Moreover, these correlated states can be embedded within an ultrametric topology. This aspect as well as a rigorous calculation of the information capacity has already been covered in reference [1].

Starting from an arbitrary initial configuration of spins, updating the network can proceed several ways, of which only one will be mentioned here. The following method is somewhat easier to implement than the one used in our earlier paper [1]. The procedure used for all our numerical simulations began by randomly selecting and then updating from t to $t+1$ each of the effective spins $s_{i_n}(t)$ belonging to the topmost (nth-level) cluster of the tree using (2.3). Since each effective spin is the sign of the magnetization of the next lower level cluster state, the sign of all effective spins belonging to this lower level cluster must change according to

$$s_{i_{n-1}}(t)\rightarrow s_{i_{n-1}}(t)\left[\frac{M_{i_n}(t)}{|M_{i_n}(t)|}\,s_{i_n}(t+1)\right] \tag{2.5}$$

where the subscript on the i index† is used to distinguish effective spins at different levels and the quantity in square brackets is minus 1 if the $n-1$ level magnetization (denoted by M_{i_n}) must change sign and plus 1 if not. Of course a cluster index should also be present which we do not show. The next step is to update the set of clusters (at the $n-1$ level) and their effective spins $s_{i_{n-1}}(t)$ again using (2.3) followed by application of (2.5) only with $n\rightarrow n-1$ and $n-1\rightarrow n-2$. This updating process is continued down the tree and terminated after updating the individual spins belonging to the lowest (first) level clusters in the network. The above process is then repeated successively until the entire network becomes stable.

3. Error propagation (static case)

The renormalization method can be used to analyse how errors (in the form of random spin flips) introduced at one level, induce errors at subsequently higher levels. To be more specific, the problem we address is the following. Given an n-level network comprised of clusters storing $p=2$ states each, we ask what is the collective effect on the network of starting with a particular stored *network* state, of which there are N_t choices and then flipping each spin ($s_i\rightarrow -s_i$) with some given probability. At the moment, we only consider the static problem which does not include updating. In response to the probabilistic flipping of first level spins, there will be some probability P_M that the *sign* of the magnetizations of the first level cluster states will change. According to our renormalization approach, P_M becomes the probability of flipping

† Here the first level spins are labelled as s_{i_1} because the level index starts at $m=1$. When referring to reference [1], beware of a change in notation where the level index starts with $m=0$ rather than 1.

the effective spins for clusters at the *next* higher level. For the general case we label these probabilities at the mth level by $P_M^{(m)}$.

The probability that the sign of a cluster state magnetization will change depends on k and on the *magnitude* of the magnetization M of the stored states. The derivation of this probability is given by equation (A3) in the appendix. As a specific example, we plot in figure 2, with a full curve, $P_M^{(m+1)}$ against $P_M^{(m)}$ for the case $k = 15$ and $M = 7$. The curve has an 'S' shape with three fixed point solutions to the equation $P_M^{(m+1)} = P_M^{(m)}$, namely 0, 1 and P_c. The critical point at $P_c = 0.5$ marks the onset of a second-order phase transition.

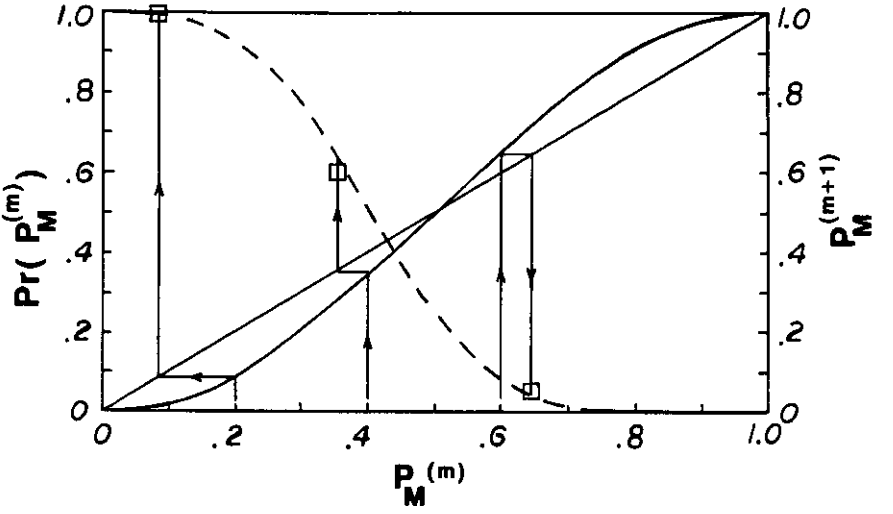

Figure 2. The full curve indicates the dependence of the probability of spin flips for the $m+1$ level cluster $P_M^{(m+1)}$ on the spin flip probability of the m level cluster $P_M^{(m)}$ for clusters having $k = 15$ spins and states constrained to $M = 7$. The broken curve shows simulation results for the probability of a perfect recall $\Pr(P_M^{(m)})$ from a $k = 15$ spin single level cluster network storing $p = 2$ states as a function of $P_M^{(m)}$. The square boxes are second level percentage recalled perfectly $P_s^{(2)}$ simulation results from a two level, $N = 225$ spin network.

Values of $P_M^{(m)}$ below P_c will give $P_M^{(m+1)}$ values less than $P_M^{(m)}$ and will lead, therefore, to the containment of errors. In contrast, when $P_M^{(m)}$ is above P_c, the error propagation is unbounded, extending all the way up the tree. This is a catastrophic failure and is similar to that found for loaded fractal trees [8]. As an example, successive iterations of equation (A3) using $k = 15$ and $M = 7$ takes $P_M^{(m)} = 0.2$ into $P_M^{(m+1)} = 0.088$, $P_M^{(m+2)} = 0.008$, $P_M^{(m+3)} = 0.000\,0018$ and rapidly approaches zero (i.e. the errors are contained) as m increases. On the other hand, starting with $P_M^{(m)} = 0.6$ the sequence becomes $P_M^{(m+1)} = 0.646$, $P_M^{(m+2)} = 0.712$, $P_M^{(m+3)} = 0.805$, $P_M^{(m+4)} = 0.916$ which tends to a probability of one (i.e. the error propagates up the tree) as m increases.

For starting values of P_M less than P_c, the maximum number of levels invaded by errors can be characterized by a propagation length L. The propagation length remains finite for P_M less than P_c but increases rapidly as P_M approaches P_c from below and can be described by a power law,

$$L \propto (P_c - P_M)^{-\nu} \tag{3.1}$$

having a critical exponent ν. For the particular case with $k = 15$ and $M = 7$ the critical exponent ν has the value 7.075 28. Varying the values of k and M yield different values of ν. The details of calculating ν are relegated to appendix B.

4. Error propagation (dynamic case)

So far, we have only considered the *static* problem of how the introduction of spin flip perturbations at one level will change the configuration of spins at subsequently higher levels. The *dynamic* problem of how the network responds during updating to this static perturbation of spins is considered next. What we now address is the process of pattern retrieval in the hierarchical model. The case considered below deals with the probabilistic flipping of spins belonging to some nominal target pattern and should be contrasted with the situation where a prescribed number of spins are flipped to form a desired overlap with the target pattern. The former case is preferred here because it preserves the scale invariance of the analysis. The latter is compatible with traditional treatments of the content-addressability of stored patterns and is covered in the discussion.

For the dynamic case we can again take advantage of the scale invariant properties of the network. From the static analysis, we know how the cluster states at each level will be affected by an initial perturbation of first level cluster spins. If we know how well an individual cluster can recall nominal target patterns as a function of the probability that spins in the target pattern initially are flipped, then in principle, we should be able to predict how the entire network recalls its stored patterns. The success of this prediction rests on the degree to which a nearest-neighbour interaction approximation (NNIA) is valid. In this context, interactions are called nearest neighbour if spins only interact with other spins belonging to the same cluster.

When the updating approach, such as the one described in section 2, makes multiple passes through the network before reaching a stable configuration of spins, then the nearest-neighbour interaction assumption is only approximate. A weak long-range interaction can arise because of the multiple sweeps through the network. As lower level cluster states are updated, the sign of their magnetization can change. This implies that the state that an upper level cluster begins with on its next updating epoch is not simply a function of its initial spin configuration and individual cluster recall probability, but also depends on the recall performance of the lower level clusters. As we will see, in spite of these long range interactions, predictions assuming the NNIA are surprisingly accurate.

The first step is to gather data on how well an individual cluster can recall nominal target patterns as a function of the probability that spins in the target pattern initially are flipped. The broken curve in figure 2 shows the simulation results for the probability $\Pr(P_M^{(m)})$ of realizing the perfect recall of a target pattern after perturbing each of the k spins with probability equal to $P_M^{(m)}$. In this simulation, the cluster had $k = 15$ spins and stored $p = 2$ randomly selected state vectors, each with $M = 7$. Including the two conjugate states (i.e. $\{s_i^{(r)}\} \rightarrow -\{s_i^{(r)}\}$) there are then a total of four possible states that can be recalled by this cluster. Because only two states are stored per cluster, the problem of spurious states does not arise, therefore *only* stored states, or their conjugates, will occur.

Both curves in figure 2 can now be used to predict, in the NNIA, how the entire network should behave dynamically in response to an initial static perturbation of

network spins. If we let $P_s^{(m)}$ be the percentage of mth level states recalled perfectly, where an mth level state involves *all* effective spins at the mth level, then it is easy to see that the general expression for $P_s^{(m)}$ is given by

$$P_s^{(m)} = \prod_{l=m}^{n} (\Pr(P_M^{(l)}))^{k^{(n-l)}}. \tag{4.1}$$

We can use (4.1) to estimate the probability of perfectly recalling mth level states for arbitrary hierarchical networks. As our first example, numerical simulations were carried out on a two-level network having $N = 225$ spins with a branching scale factor of $k = 15$. Each cluster stored $p = 2$ states with M fixed at 7. This network can store $N_t = 2^{16}$ states. In figure 2 we show with arrow traces, the predicted *second* level $P_s^{(2)} = \Pr(P_M^{(2)})$ recall probability results for a two-level network assuming three different starting spin flip probabilities $P_M^{(1)} = 0.2$, 0.4 and 0.6. These map into $P_M^{(2)} = 0.09$, 0.35 and 0.65 probabilities respectively and when projected onto the broken curve, predict the probabilities $P_s^{(2)} = 0.99$, 0.63, 0.04 respectively, of perfectly recalling second-level states. The small square boxes mark the corresponding $P_s^{(2)} = 0.99$, 0.59, 0.05 results found during the numerical simulations of the above two-level network and show good agreement.

Using (4.1), we can also predict the probability $P_s^{(1)}$ of a perfect recall in the first-level state (i.e. network state). Explicitly, $P_s^{(1)}$ is equal to $(\Pr(P_M^{(1)}))^k \cdot (\Pr(P_M^{(2)}))^1$ which evaluates to $(0.93)^{15} \cdot (0.99) = 0.33$, $(0.51)^{15} \cdot (0.63) = 0.0$ and $(0.09)^{15} \cdot (0.04) = 0.0$ when $P_M^{(1)} = 0.2$, 0.4 and 0.6 respectively. Again, satisfactory agreement was found with our simulation results which gave perfect recall values of 0.23, 0.0 and 0.0 respectively. The same two-level network only with $M = 3$ instead of 7 was also examined as well as a three-level network with $k = 5$ and $M = 1$ and 3. The predicted and simulated results for these other networks are also in satisfactory agreement and

Table 1. Tabulated values of the percentage of stored states recalled perfectly for two level ($k = 15$ with $M = 7$, 3) and three level ($k = 5$ with $M = 3$, 1) networks as a function of the first level spin flip probability $P_M^{(1)}$. The predicted results, assuming a nearest-neighbour approximation (NNIA), and actual simulation recall data are shown. The $P_s^{(m)}$ values are the percentage of mth level states recalled perfectly as defined in equation (4.1).

				Percentage recalled perfectly					
				Predicted (NNIA)			Simulated		
Levels	k	M	$P_M^{(1)}$	$P_s^{(1)}$	$P_s^{(2)}$	$P_s^{(3)}$	$P_s^{(1)}$	$P_s^{(2)}$	$P_s^{(3)}$
2	15	7	0.2	0.33	0.99		0.23	0.99	
			0.4	0.0	0.63		0.0	0.59	
			0.6	0.0	0.04		0.0	0.05	
		3	0.2	0.25	0.81		0.27	0.89	
			0.4	0.0	0.40		0.0	0.45	
			0.6	0.0	0.13		0.0	0.10	
3	5	3	0.05	0.25	0.94	1.0	0.13	0.90	1.0
			0.1	0.04	0.77	0.99	0.03	0.70	0.99
			0.2	0.0	0.35	0.90	0.0	0.33	0.91
		1	0.05	0.07	0.26	0.62	0.07	0.39	0.63
			0.1	0.0	0.07	0.46	0.02	0.17	0.47
			0.2	0.0	0.01	0.35	0.0	0.06	0.40

have been tabulated in table 1. All simulations predicted in table 1 were carried out using the top-down updating procedure described in section 2 and involved roughly 200 simulations per data point.

5. *Discussion*

Through appropriate choices of k and M values in equation (A3), the spin flip probability at the $m+1$ level can be made to be always less than the spin flip probability at the mth level, provided the spin flip probability at the first level remains below its critical value. When this condition is satisfied, the probability of recalling higher level states improves with increasing m (i.e. $P_s^{(m)} < P_s^{(m+1)}$, $m = 1, 2, \ldots, n-1$). The significance of this in content-addressing applications is that even though some of the bits have changed (i.e. 'details' are lost) in the recalled network state, the 'rough image' of the original unperturbed state is preserved.

This ties in with what was called the ultametric rule in [1] which stated that higher level states (or magnetization states as referred to in [1]) are more likely to be recalled correctly over those at lower levels. In addition, referring to (4.1) it can be seen that $P_s^{(m)}$ has a tendency to increase anyway, for increasing m, simply from the fact that it depends on fewer clusters as m increases. Hence, even though we may be in a region where $P_M^{(m+1)}$ is greater than $P_M^{(m)}$, the upper levels may still recall patterns better.

As a further check on the utility of the RG approach in predicting network recall performance, we compared theoretical expectations with empirical data taken from simulations presented in [1]. These simulations were intended to demonstrate how the network's associative recall ability, starting from perturbed nominal target patterns, was dependent on the first level cluster size, second level cluster size and the magnitude of the first level cluster state magnetization. For these particular network simulations, the analysis had to take into account the different method of initially perturbing the target patterns. In [1], the percentage of states recalled successfully was determined as a function of the overlap between the starting state and some nominal target pattern, whereas, in the simulations studied in section 4, the first level spins were each flipped with some probability $P_M^{(1)}$.

When the overlap is given, the number of first level spins that are flipped is also known. This changes the expression for determining the probability that the sign of the magnetization of the *first* level cluster state will change. In this case, the probability is determined by summing $P_f(f^+)$, as given by (A1), over f^+, where the summation range for f^+ is the same as used in (A3). Expression (A3) is not used for the first level clusters in this case since f is known from the initial overlap.

With the exception of the above change, the theoretical analysis of the hierarchies in [1] followed the same procedures as outlined in the last two sections. The resulting predictions of the content-addressability of stored patterns were in general accord with the simulation results. We could satisfactorily predict observed recall behaviour as the cluster branching size and magnetization were independently varied.

We have already mentioned that the updating approach described in section 2 and used in all of the simulations presented in table 1 can lead to long range interactions making the assumption of nearest-neighbour interactions only approximate. This is also true for the method used in [1]. This need not, however, always be the case. By updating from the top-down each cluster sequentially such that all clusters at a given

level are stabilized *before* (2.5) is applied and then moving down to the next level, the problem of long range interactions can be avoided.

In conclusion, we find that the globally complex hiearchical model is easy to analyse because of its inherent scale invariant properties. For the examples studied here and in [1], the NNIA appears to work well even when different updating methods are used. The renormalization approach allows results obtained on individual Hopfield network simulations to be used to predict global properties of the model and can be utilized when designing effective hierarchical neural network systems.

Appendix A

We wish to find the probability that a cluster of size k and initial magnetization M will go to a new magnetization M' after flipping f spins at random, but never the same spin twice. To evaluate this probability, we suppose that the cluster state contains $n^+ \equiv (k+M)/2$ plus 1 spins and $n^- \equiv (k-M)/2$ minus 1 spins (so that $n^+ - n^- = M$) and ask for the probability of finding f^+ plus 1 spins and $f^- = (f - f^+)$ minus 1 spins out of f random selections. This probability has been explicitly derived in [1], and works out to be simply the hypergeometric distribution

$$P_f(f^+) = \binom{n^+}{f^+}\binom{k-n^+}{f-f^+}\binom{k}{f}^{-1} \tag{A1}$$

where

$$\binom{a}{b} \equiv \frac{a!}{b!(a-b)!}$$

and it is assumed that $b \leqslant a$ otherwise the binomial coefficient is defined to be zero.

Note that $P_f(f^+)$ is also the probability that given f^+ spins and f selections, the magnetization goes from $M = n^+ - n^-$ to M' where,

$$M' = (n^+ - f^+ + f^-) - (n^- - f^- + f^+) = M - 2(2f^+ - f). \tag{A2}$$

If we sum $P_f(f^+)$ over all f^+ greater than $(f/2 + M/4)$, but less than f, and then multiply by the probability of getting exactly f spins flipped out of k, given that the probability of flipping one spin is $P_M^{(m)}$, and finally summing over all spins within the cluster, we will get the probability of changing the sign of the cluster magnetization $P_M^{(m+1)}$. Explicitly written out this becomes,

$$P_M^{(m+1)} = \sum_{f=0}^{k} \sum_{f^+}^{f} \binom{n^+}{f^+}\binom{k-n^+}{f-f^+} (P_M^{(m)})^f (1 - P_M^{(m)})^{(k-f)} \tag{A3}$$

where the sum over f^+ begins at the next integer greater than $(f/2 + M/4)$. In the event that this evaluates to greater than f, then the contribution to the sum for this value of f is zero.

Appendix B

The calculation of ν depends on the renormalization scale factor and on the probability of changing the cluster state magnetization. For our case, the renormalized propagation

length obeys,

$$L(P_M^{(m+1)}) = \frac{1}{k} L(P_M^{(m)}) \tag{B1}$$

where k is the scaling factor and $P_M^{(m+1)}$ is the probability of changing the cluster magnetization at the $m+1$ level defined in (A3). Once k and $P_M^{(m+1)}$ are established, the computation of ν is straightforward. Following Kogut and Wilson [4], when P_M is close to P_c we can expand about P_c with a linear approximation,

$$P_M^{(m+1)} = P_c - \lambda(P_c - P_M^{(m)}) \tag{B2}$$

from which λ is then given by

$$\lambda = \frac{P_c - P_M^{(m+1)}}{P_c - P_M^{(m)}} \tag{B3}$$

and ν follows from $\nu = \ln(k)/\ln(\lambda)$. For $k = 15$ and $M = 7$, we obtain using (B3) and (A3) $\lambda = 1.466\,30$ from which $\nu = 7.075\,28$.

References

[1] Willcox C R 1989 *J. Phys. A: Math. Gen.* **22** 4707

[2] Dotsenko Vik S 1985 *J. Phys. C: Condens. Matter* **10** L1017

[3] Abu-Mostafa Y S and St Jacques J M 1985 *IEEE Trans.* **IT-31** No. 4 461

[4] Wilson K G and Kogut J 1974 *Phys. Rep.* C **12** 75

[5] Pfeuty P and Toulouse G 1977 *Introduction to the Renormalization Group and to Critical Phenomena* (New York: Wiley)

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

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

[8] Solla S A 1986 *Collapse of Loaded Fractal Trees, Fractals in Physics* ed L Pietronero and E Tosatti (Amsterdam: Elsevier) p 185

Smalley R F Jr, Turcotte D L and Solla S A 1985 *J. Geophys. Res.* **90** 1894