High Capacity, Small World Associative Memory Models

NEIL DAVEY, LEE CALCRAFT and ROD ADAMS

School of Computer Science, University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK

{N.Davey, L.Calcraft, R.G.Adams}@herts.ac.uk

Abstract

Models of associative memory usually have full connectivity or if diluted, random symmetric connectivity. In contrast, biological neural systems have predominantly local, non-symmetric connectivity. Here we investigate sparse networks of threshold units, trained with the perceptron learning rule. The units are given position and are arranged in a ring. The connectivity graph varies between being local to random via a small world regime, with short path-lengths between any two neurons. The connectivity may be symmetric or non-symmetric. The results show that it is the small-world networks with non-symmetric weights and non-symmetric connectivity that perform best as associative memories. It is also shown that in highly dilute networks small world architectures will produce efficiently wired associative memories, which still exhibit good pattern completion abilities.

1 Introduction

There is a long history of research into the properties of random graphs, graphs in which the connectivity matrix is randomly configured, often with a specific probability of connectivity (Bollobas, 2001). Recently there has been an explosion of interest in networks with non-random connectivity graphs, such as *small world* and *scale free* networks, as described in Section 2.6. Such networks have interesting properties and their patterns of connectivity have also been found to be very common in real networks, both biological and otherwise. There has also been a huge increase in our understanding of the connectivity pattern in real neuronal networks, as summarised in Section 2.7. In this paper we report our own empirical investigation into the relationship between the pattern of connectivity and performance, in sparsely connected associative memory models. The primary motivation for this investigation is to answer the following question: are there general principles that the connectivity pattern in an associative memory should adhere to in order to produce: i) good functionality and ii) parsimonious wiring.

2 Background

2.1 Associative Memory Model

The standard neural network model of associative memory, the Hopfield network (Hopfield, 1982), is one of the most studied of all neural networks. It offers a highly simplified model of how associative memory can function in a collection of units that have asynchronous dynamics and a simple update rule. This property led it to be considered as a very abstract model of some of the functionality in mammalian brains (Amit, 1989). However, the full connectivity of the network implies that the model does not scale well. Any physically realised, large scale Hopfield network will be difficult to build: the number of connections grows with the square of the number of units.

Moreover, many of the assumptions of this model are problematic when it is considered as a model of brain function. Specifically, the required symmetry of connections and the full connectivity do not find parallels in real neuronal systems. More recent work (Davey, and Adams, 2004a) has established that some of the constraints of the Hopfield network can be relaxed, and that the learning rule can be much improved. In the work reported here we use a version of the Hopfield network that maintains some of the appealing aspects of the model, such as simple dynamics, whilst adhering to some more biologically plausible assumptions. In particular our models have some, or all, of the following characteristics:

- Sparse, non-symmetric connectivity
- A relatively well performing learning rule
- Spatial positioning of the artificial neurons
- Structured, non-random connectivity

2.2 Dynamics

Each unit in the network is a simple, bipolar, threshold device, summing its net input and firing deterministically. The net input, or *local field*, of a unit, is given by: $h_i = \sum_{j \neq i} w_{ij} S_j$

where $\boldsymbol{S}$ (± 1) is the current state and w_{ij} is the weight on the connection from unit *j* to unit *i*. The dynamics of the network is given by the standard update:

$$S_i' = \begin{cases} 1 & \text{if } h_i > 0 \\ -1 & \text{if } h_i < 0 \\ S_i & \text{if } h_i = 0 \end{cases} \quad \text{where } S_i' \text{ is the new state of } S_I .$$

Unit states may be updated synchronously or asynchronously. Here we use asynchronous, random order updates. Using a symmetric weight matrix and asynchronous updates ensures that the network will evolve to a fixed point. However, in practice, the symmetric weight constraint can be relaxed without damaging the convergence properties (Chengxiang, Dasgupta, and Singh, 2000), and in many of the

models discussed here the weights are not required to be symmetric. In fact, as we show later, symmetric weights can be damaging to the performance of the network, at low levels of connectivity.

If a training pattern $\boldsymbol{\xi}^{\mu}$ is one of the fixed points of the network then it is successfully stored, and is said to be *a fundamental memory*. A network state is stable if, and only if, all the local fields are of the same sign as their corresponding unit, equivalently the *aligned local fields*, $h_i S_i$, should be non-negative.

2.3 Connectivity

The networks analysed here have sparse connectivity, so that only a small fraction of all the possible N^2 connections in an N unit network are present. Equivalently the connection matrix $\{C_{ij}\}$ where $C_{ij} = 1$ iff w_{ij} is present and 0 otherwise, is sparse. In general we do not require symmetric connectivity, so that C is not required to be a symmetric matrix. We do however make one simplification in the allowed patterns of connectivity: each unit in the network has the same number, k, of afferent (incoming) connections. This is equivalent to requiring that the connection graph is *regular*. The reason for this simplification is that the performance of the units in the network is, to some extent, determined, in a well understood way (see 2.4 below), by the afferent connectivity level. Since we are interested in the collective behaviour of functionally similar units, we give them identical levels of connectivity.

As described earlier, we are interested in the spatial organisation of effective patterns of connectivity, and it is therefore necessary for the units in the network to have position. The simplest approach is taken, and the units are arranged in one dimension, and to avoid edge effects are placed in a ring. We take the minimum path length between any two nodes to be the actual distance between them, giving a simple geometry. Within this configuration there are two extremes of connection pattern. Firstly the connections can be placed as locally as possible, giving a completely *local network* (Figure 1 left). This is clearly the configuration that minimises wiring length. Alternatively the connections can be placed completely randomly giving a *random network* (Figure 1 right). Between these two extremes are other architectures discussed in the Section 2.5.

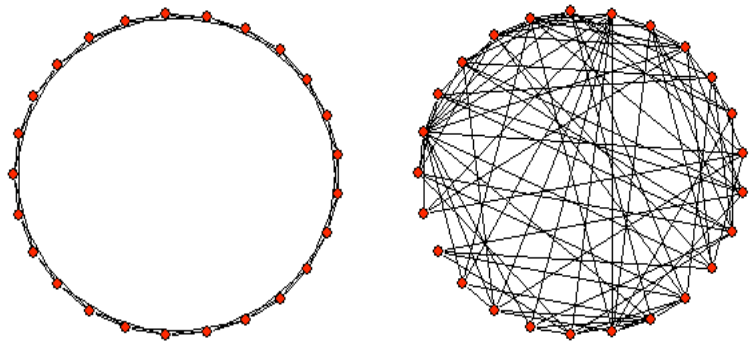


Figure 1: The units arranged in a ring. Left: a locally connected network and right: a random network. In both cases the number of afferent connections is, $k = 4$. Diagrams generated with the Pajek package (de Nooy, Mrvar, and Batagelj, 2005).

2.4 Learning

The standard learning rule in Hopfield networks is one-shot Hebbian. This learning rule is attractive not only for its simplicity, but also because it gives networks that are tractable to analysis by the powerful tools of statistical physics. However, in terms of producing good associative memories it is a rather poor choice, since there is no guarantee that the training patterns are actually learnt (even at low loadings). There are, however, two other classes of learning rule that perform much more effectively: those based around the perceptron learning rule and those based on the pseudo-inverse matrix method. These methods actually produce networks that perform very similarly (Davey, Hunt, and Adams, 2004b), and due to its simplicity we use perceptron based learning in our networks. Given a training set $\{\boldsymbol{\xi}^{\mu}\}$ the training algorithm is designed to drive the aligned local fields of each unit the correct side of the learning threshold, T, for all the training patterns. This is equivalent to requiring that $\forall i, \mu \quad h_i^{\mu}\xi_i^{\mu} \geq T$.

So the learning rule is given by:

Begin with a zero weight matrix

Repeat until all local fields are correct

Set the state of the network to one of the $\boldsymbol{\xi}^{\mu}$

For each unit, i, in turn

Calculate $h_i^p \xi_i^p$.

If this is less than T then change the weights on connections into unit i according to:

$$\forall j \neq i \quad w'_{ij} = w_{ij} + C_{ij}\frac{\xi_i^p \xi_j^p}{k} \qquad (1)$$

The form of the update in (1) is such that changes are only made on the weights that are actually present in the connectivity matrix C, and that the learning rate is inversely

proportional to the number of connections k. Earlier work has established that a learning threshold $T = 10$ gives good results (Davey *et al.*, 2004b).

When the connectivity matrix is symmetric the learning rule can be modified to produce symmetric weights (Gardner, 1988). This is simply achieved by modifying w_{ji} whenever w_{ij} is changed. The learning rule is then given by:

Begin with a zero weight matrix

Repeat until all local fields are correct

Set the state of network to one of the $\boldsymbol{\xi}^{\mu}$

For each unit, i, in turn

Calculate $h_i^p \xi_i^p$.

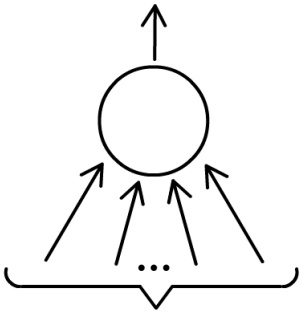
If this is less than T then change the weights between unit i and all other units, j, according to:

$$\forall j \neq i \quad w'_{ij} = w_{ij} + C_{ij} \frac{\xi_i^p \xi_j^p}{N} \ , \quad w'_{ji} = w'_{ij}$$

We denote the symmetric learning rule *SL* and the non-symmetric rule *NSL*.

As is well known, training a perceptron using *NSL* is a convergent process if a set of weights actually exists that embeds the training set (Hertz, Krogh, and Palmer, 1991). If there are P patterns in the training set then the loading of the network, the number of patterns stored per connection, for each unit, is $\alpha = \frac{P}{k}$. We use random training sets, so the task of each perceptron is to learn the correct output for each of its P input vectors. A perceptron can learn such a mapping if the two classes (those for which the desired output is +1 and those for which it is -1) to be learnt are linearly separable in $\mathcal{R}^k$. In 1965 Cover (Cover, 1965) showed that this is likely to be the case for up to $2k$ random patterns (a loading of $\alpha = 2$), with convergence as k becomes large. The capacity of these networks of perceptrons is therefore determined by the level of connectivity and not the pattern of connectivity, and is thus not a subject of our experiments. Surprisingly the capacity of the symmetric learning rule, for fully connected networks, can be shown by a theoretical argument, to be the same as that of the non symmetric network (Nardulli, and Pasquariello, 1991) (surprising because the symmetric network has only half the number of independent weights as its non-symmetric counterpart). Empirical work has also shown similar capacity in the two learning rules for dilute, sparsely connected, networks (Davey *et al.*, 2004a).

However, the ability to store patterns is not the only functional requirement of an associative memory: the fundamental memories should also act as *attractors* in the state space of the dynamic system resulting from the recurrent connectivity of the network, so that pattern correction can take place. And it is the case that the pattern of connectivity has a major influence on this capability (Komlos, and Paturi, 1993). The position is summarised in Figure 2. For example, if the network graph is disconnected, then information cannot pass between these subgraphs. Moreover if the network has only local connections, local domains of errors tend not to be corrected (Noest, 1989).



k inputs $\Rightarrow$ Capacity = $2k$

Figure 2: A perceptron with k inputs has a capacity of $2k$. A collection of recurrently connected perceptrons, each with k inputs will also have capacity $2k$. However the actual pattern correction performance of the network is influenced by the origin of these connections.

2.5 Performance Measures

We are interested in the performance of our networks as effective associative memories, so it is necessary to measure the pattern completion ability of the network. We use two measures. Once a network has been trained on a set of random patterns it is relatively straightforward to evaluate its pattern correction capabilities. The well established method is to estimate *the mean normalised radius of the basins of attraction* of the fundamental memories, R. In essence this is the maximum proportion of bits in a fundamental memory that can be randomised and then fully corrected by the network dynamics.

Since the attractor basins cannot be expected to be Hamming hyperspherical (Storkey, and Valabregue, 1999), it is usual to take the minimum Hamming radius:

$$R\left(\boldsymbol{\xi}^p\right) = \inf\left\{\left|\mathbf{q} - \boldsymbol{\xi}^p\right| : \mathbf{q} \in Basin\left(\boldsymbol{\xi}^p\right)\right\}$$

The mean radius of attraction over the patterns, R, can act as a measure of the quality of a particular associative memory. It is also common for R to be normalised with respect to the size of the network, so that it lies between zero and one.

For very small networks it is possible to exhaustively explore the state space (see, for example Personnaz, Guyon, and Dreyfus, 1986), in order to calculate R exactly, but for more realistic sizes the nature of the attractors is very hard to compute (Floréan, and Orponen, 1993; Kepler, and Abbott, 1988) and only empirical methods are available.

A sample of states at a fixed distance, r, from a trained pattern, $\boldsymbol{\xi}^p$, is made, and if all of them relax to $\boldsymbol{\xi}^p$, it is concluded that $R(\boldsymbol{\xi}^p)$ is at least as big as r. Clearly, the larger the sample size the higher the quality of the estimate; in all of our experiments the sample

size is 50. An analysis of the effect of sample size on the estimate of R can be found in (Davey *et al.*, 2004b).

In our implementation we have slightly adapted the method of Kanter and Sompolinsky (Kanter, and Sompolinsky, 1987) in the calculation of R. For each of the sample states chosen, a fixed fraction, m_0, of the state is identical to the corresponding part of one of the stored patterns, ξ^p , and the rest of the state is random. Initially a low value is taken for m_0 and consequently it needs to be incrementally increased until all of the sample states relax to ξ^p. Averaging m_0 over different stored patterns yields:

$$R = 1 - \langle m_0 \rangle$$

As is pointed out in (Kanter *et al.*, 1987), for finite size associative memories, another factor needs to be considered. The initial states used in this calculation may overlap one of the other stored patterns more closely than ξ^p, and to compensate for this the definition of R is modified to:

$$R = \left\langle\left\langle \frac{1 - m_0}{1 - m_1} \right\rangle\right\rangle$$

where m_1 is the overlap with closest of the remaining patterns. This is a double average over both different sets of stored patterns and different sample states.

So in our implementation, a fixed number of random starting points are chosen, each of which has a low overlap with a member of the training set (low average m_0). If, as is likely, the start state does not relax to that training pattern in one or more of the random cases, the value of m_0 is increased (by $1/N$), and the search is repeated. This continues until all random start states relax to the closest stored pattern. This whole procedure of training the network with the same number of random patterns is repeated 50 times and the average value of R is reported. The perfect attractor network has $R = 1$, which means that it is possible to move away from any stored pattern, and stay within its basin of attraction up to the point at which another stored pattern becomes nearer (see Figure 3). Note that the calculation of average attractor basin size for the trained patterns can only be undertaken when these patterns are themselves stable.

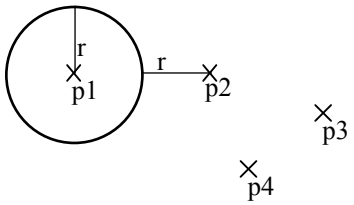


Figure 3. Calculating R. In this figure p1, p2, p3 and p4 are fundamental memories. The closest pattern in the training set to p1 is p2, at a distance of 2r. Optimal performance occurs when all vectors within the hypersphere centred on p1 and radius r, are attracted to p1. If all patterns stored in a network exhibit this performance, its normalised average radius basin of attraction, R, is 1.

The second measure that we use is the *Effective Capacity* of the network, *EC* (Calcraft, 2005). The *R* value measures the performance of the network at a particular loading α, but it is also useful to measure the intrinsic capabilities of a specific pattern of connectivity, independently of the actual loading. The Effective Capacity of a network is a measure of the maximum number of patterns that can be stored in the network with *reasonable* pattern correction still taking place. We take a fairly arbitrary definition of *reasonable* as correcting the addition of 60% noise to within an overlap of 95% with the original fundamental memory. Varying these figures gives differing values for *EC* but the values with these settings are robust for comparison purposes. For large fully connected networks the *EC* value is about 0.1 of the conventional capacity of the network, but for networks with sparse, structured connectivity *EC* is dependent upon the actual connection matrix *C*.

The *EC* of a particular network is determined as follows:

Initialise the number of patterns, P, to 0
Repeat
    Increment P
    Create a training set of P random patterns
    Train the network
    For each pattern in the training set
        Degrade the pattern randomly by adding 60% of noise
        With this noisy pattern as start state allow the network to converge
        Calculate the overlap of the final network state
        with the original pattern
    EndFor
    Calculate the mean pattern overlap over all final states
Until the mean pattern overlap is less than 95%
The Effective Capacity is P-1

As with the *R* calculation, described above, the degraded fundamental memory may become closer to one of the other fundamental memories. If this is the case then the degraded pattern is rejected, and another generated. For implementation purposes, a binary search algorithm is used to search for the loading resulting in 95% or better recall, rather than simply increasing the loading from unity upwards (Calcraft, 2005). Similar measures have often been used elsewhere, for example: *partial capacity* in (Komlos *et al.*, 1993), *error correcting ratio* in (Gorodnichy, 1999) and *effective retrieval* (Kosinski, and Sinolecka, 1999).

As symmetry of the weights is one of the issues that we investigate we make use of the standard symmetry measure for a matrix, defined as: $\sigma = \frac{\sum_{i,j} w_{ij} w_{ji}}{\sum_{i,j} w_{ij}^2}$. For a symmetric matrix this has value +1, for a random matrix it will be roughly 0, and for an anti-symmetric matrix it will be -1.

2.6 Non-Random Graphs

The seminal paper of Watts and Strogatz (Watts, and Strogatz, 1998) formalised the notion of a *Small World* network. The idea was inspired by work in the Social Sciences showing that there appeared to be only roughly 6 degrees of separation (by acquaintance) between any two people in North America (Milgram, 1967); this despite the fact that most people have a cliquish group of acquaintances, in the sense that any two of their acquaintances are also likely to be acquaintances. The *Small World Effect* is therefore characterised as a network with short path lengths (the minimum number of arc traversals to get from one node to another), between any pair of nodes. The simplest sort of network that displays this characteristic is a random network. In a regular random network of N nodes, with each node having k connections, the number of first order acquaintances is k, second order is about k^2, third order k^3 and so on. So in general the number of degrees of separation, D, to reach all N nodes in the network is given by setting $k^D = N$, which gives $D = \frac{\ln N}{\ln k}$, so that D increases logarithmically with the size of the network – the small world effect. However, random networks are not cliquish and require a relatively large amount of wiring. Watts and Strogatz gave a mechanism for constructing networks that showed the small world effect, from local networks. Their idea was to begin with a local network and then to *rewire* a small proportion, p, of these connections to random targets. At even very low levels of rewiring, the mean path length between any pair of nodes drops to a value comparable to that of a random network: the rewired connections act as shortcuts through the network. Figure 3 shows the earlier local network of Figure 1 rewired with a probability $p = 0.1$.

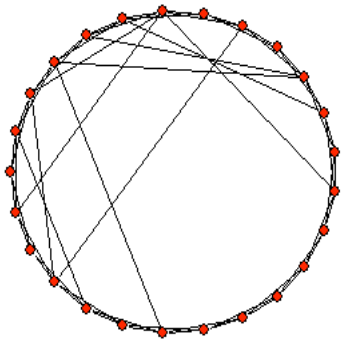


Figure 4: A small world network with rewiring $p = 0.1$.

In fact it is now apparent that a slightly different version of the Watts and Strogatz network construction method leads to networks that are more amenable to theoretical analysis. In this version, additional random connections are added, leaving all the local connections intact (Newman, 2000). It is to these networks that the label *Small-World Network* is normally used. However, we use the original formulation as it maintains the regularity of the connection graph.

The cliquishness of a network can be formalised by its *clustering coefficient*, the average fraction of pairs of neighbours of a node, which are also neighbours. Networks that show the small world effect, but which also have high clustering coefficients have been shown to be remarkably common. Some examples include: networks of movie actors, where neighbours are defined by having been in the same movie, power grid networks, the Internet and from our point of view most interestingly, real neuronal networks, to which we turn to in the next Section.

Other interesting networks that show the small world effect are so called *Scale-Free* networks (Barabasi, Albert, and Leong, 1999; Keller, 2005). These are network models where the distribution of connections follows a power law (that is the frequency of nodes with connectivity k falls off as $k^{-\alpha}$). This degree distribution is surprisingly close to that of the distribution of links in the World Wide Web. Some nodes end up with very high levels of connectivity, and act as network hubs, that facilitate short path lengths. Such networks can arise due to a preferential growth process in which nodes that are already well connected are favoured by new connections.

2.7 Connectivity in Real Neuronal Networks

The neuronal network of the nematode worm *C. Elegans* has been completely mapped. It consists of 302 neurons and around 1000 connections A recent analysis (Cherniak, 1994) of the optimality of the positioning of the neurons (for the given connectivity and physical position of actuators and sensors in the worm) with respect to the total length of

wiring (the sum of the length of neuronal fibre) has shown that no better positioning can be found by exhaustive search; a remarkable triumph for evolutionary optimisation. The network also displays short path lengths, 2.65 steps between any two neurons, and a relatively high clustering coefficient of 0.28 (as against 0.05 in an equivalent random network). In (Shefi, Golding, Segev, Ben-Jacob, and Ayali2, 2002) cultured in-vitro neuronal networks are studied. They vary in size from $N = 104$ to $N = 240$. Once again the networks show the small world effect and are relatively highly clustered.

Larger neuronal networks found in more sophisticated animals are not as well understood. Nonetheless several studies have been undertaken into the positioning and connectivity of the neuronal systems. Analysis of the mammalian cortex has been undertaken at two levels of granularity, firstly at the level of the positioning and connectivity of distinct functional areas such as V1 or V2 in the visual cortex. And secondly at the level of individual neurons. In the first case it has been shown that once again positioning is highly optimised (Cherniak, Mokhtarzada, Rodriguez-Esteban, and Changizi, 2004; Hilgetag, and Kaiser, 2004; Laughlin, and Sejnowski, 2003) to minimise connection length. It has also been shown that the connectivity gives both a small world effect and a high clustering coefficient (Sporns, and Zwi, 2004). The question of whether these neuronal systems show the characteristics of scale-free networks is still open, with opinions differing. (Eguiluz, Chialvo, Cecchi, Baliki, and Apkarian, 2005; Sporns *et al.*, 2004).

At the level of individual neurons the connectivity pattern is so complex that only generalised statistics can be produced, see (Braitenberg, and Schüz, 1998) for a fascinating discussion.

2.8 Eigenvalues of the Connection Matrix

In the standard, fully connected Hopfield network the performance of the network can be theoretically predicted from the loading, α. The larger α then the poorer will be performance. So, for example, the maximum loading at which the training patterns are likely to be stable is known to be, approximately, $\alpha = 0.138$ (Amit, Gutfreund, and Sompolinsky, 1985). In a very interesting, but little known paper (Komlos *et al.*, 1993) it is proved that in regular, sparsely connected Hopfield networks there is an analogous parameter that determines performance. If λ_1 and λ_2 are respectively the first and second eigenvalues of the connection matrix *C*, then the quantity $\alpha + \frac{\lambda_2}{\lambda_1}$ determines the performance of the network, in an direct analogy with α, in a fully connected network. The term $\frac{\lambda_2}{\lambda_1}$ acts as extra loading on the network, and in a regular, symmetric network λ_1 will be equal to the degree of the graph, which in our case is *k*, so that λ_2 is the key determiner of performance. If the network has two or more disconnected subgraphs then λ_2 will also be equal to *k*. This is the worst case, where information in one sub-graph cannot pass to the other subgraph(s). The better the connectivity properties of the graph, the lower will be λ_2 (for a fully connected network all the eigenvalues except the

first are -1), and the better will be the performance of the network, at a given loading. Although the networks used here are not equivalent to sparse Hopfield networks (for reasons explained earlier in 2.1), it is highly likely that connectivity properties of the connection graph will also be important in determining performance. Figure 5 gives the result of computing λ_2 for 1000 unit networks with $k = 60$ at varying levels of rewiring. For the local network ($p = 0$) λ_2 is only just less than 60 and so is very close to λ_1 ($\lambda_1 = k = 60$). The network is almost disconnected and will perform very poorly. From this point on it can be seen that, as expected, the connection properties of the network improve in a linear fashion with rewiring. The randomly connected network ($p = 1$) does not have $\lambda_2 = -1$ (as would a fully connected network) as it is only sparsely connected.

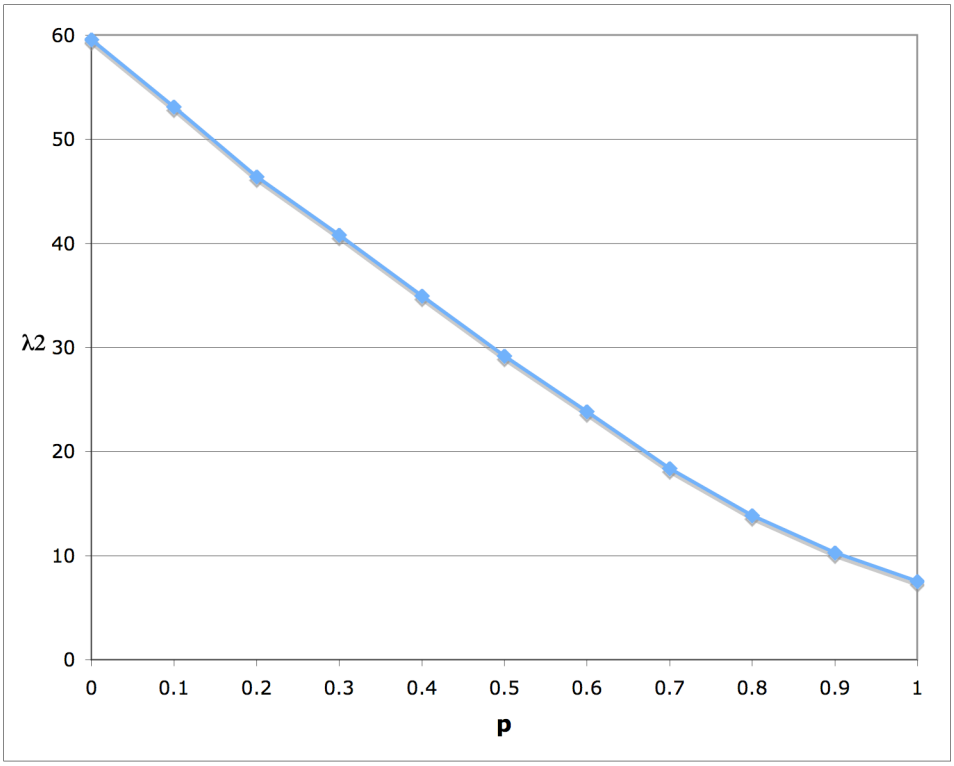


Figure 5: 1000 unit small world networks with 60 input connections. Results are averages over 5 runs at each value of the rewiring parameter, p. The second eigenvalue of the connection matrix is shown.

3 Results

The main result from the investigations presented here is that small world networks produce excellent performance in terms of both Effective Capacity and pattern completion ability, as measured by R, even for moderately low amounts of rewiring. Bohland and Minai (Bohland and Minai, 2001) and others (see Section 4) have looked at small world architecture versions of the standard Hopfield network. Here we are

investigating small world models of Hopfield style networks but trained using the Perceptron learning rule. Figure 6 shows the value of *R* plotted against loading for both types of network and illustrates the significant difference in pattern completion performance that there is between the standard Hopfield Network (trained with one-shot Hebbian learning) and the version using the perceptron learning rule. (In all our results we do not show confidence limits as they are typically very small, order ±1% of the mean, and are hardly visible. In fact the confidence intervals do not overlap in areas where the graphs are not asymptotically close to 1.)

However, before we present our main small world results we need to decide the type of network connectivity and learning rule to use (see Section 2.4 for the two learning rules). In other words, should we be keeping both symmetric connectivity and symmetric learning, or are either of the non-symmetric versions better. These results are given in the next section and the main results are then presented in section 3.2.

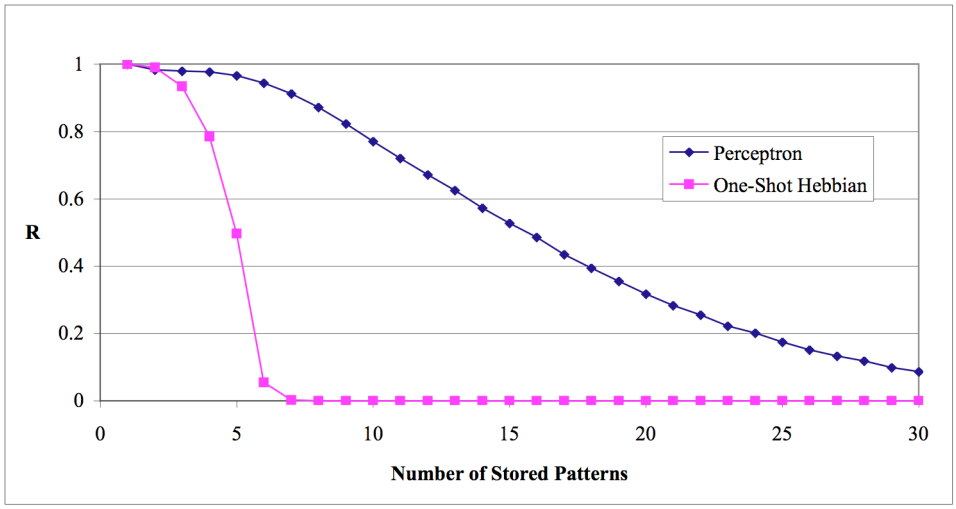


Figure 6: 1000 unit networks with 60 input connections per node. Results are averages over 50 runs at each loading. The attractor performances, *R*, for both the standard Hopfield Network (lower line) and the better performing Perceptron trained networks (upper line).

3.1 Symmetric or non-symmetric learning and connectivity

Here we have taken both the symmetric (*SL*) and non-symmetric (*NSL*) versions of the perceptron learning rule (Section 2.4) and combined them with symmetric and non-symmetric connectivity (denoted *SC* and *NSC* respectively) to give three different types of network. There are only three models since networks with non-symmetric connectivity cannot be trained using the symmetric learning rule. Table 1 summarises these network types.

TABLE 1:
THE TYPES OF NETWORK CONNECTIVITY AND LEARNING RULE

Name	Connection Matrix	Learning Rule
NSC-NSL	Non-Symmetric	Non-Symmetric
SC-NSL	Symmetric	Non-Symmetric
SC-SL	Symmetric	Symmetric

3.1.1 Small world models

For each of the network types in Table 1, a 1000 unit network, configured as a ring with an initial arrangement of 60 nearest neighbour connections (30 either side) for each node, is constructed. These were trained with sets of 18 random, unbiased patterns. Each network type was then rewired with a varying probability, p, in steps of 0.1 up to full rewiring with $p = 1$. The type of connectivity (symmetric or non-symmetric) was maintained during rewiring. Figure 7 shows the R measure for each network type at each level of rewiring. Figure 8 shows the values of Effective Capacity (EC) for each network type at each level of rewiring, but here the number of patterns is not pre-determined, since the number of patterns effectively stored is precisely what the EC value is measuring.

For all graphs, the poorest performance is for local connectivity ($p = 0$) since such graphs are poorly connected with high path lengths between any pair of nodes. As the amount of rewiring increases, the R value and the EC value increase for all networks. The *NSL* networks peak at $R = 1$ earlier than the *SL* network, with the fully non-symmetric one performing best, indicating that less rewiring is needed to perfect the pattern completion performance for non-symmetric networks than for the fully symmetric network. The EC values show the same relative performance, with the fully non-symmetric network having the highest Effective Capacity and the fully symmetric network the lowest.

It is interesting to note the effect of the rewiring on the symmetry (σ) of the weights in the networks (see section 2.5 for the definition of σ). This is shown in Figure 9. Here the *NSC-NSL* network rapidly loses its weight symmetry, reducing to an extremely low level by full rewiring. The other two networks maintain better symmetry, despite the non-symmetric learning used in the *SC-NSL* network. It is significant that the *NSC-NSL* network still performs so well despite its low level of weight symmetry: it has an R value of 1 and an almost maximal value of EC from a rewiring of 0.4 onwards while having a symmetry of ~0.3 or lower.

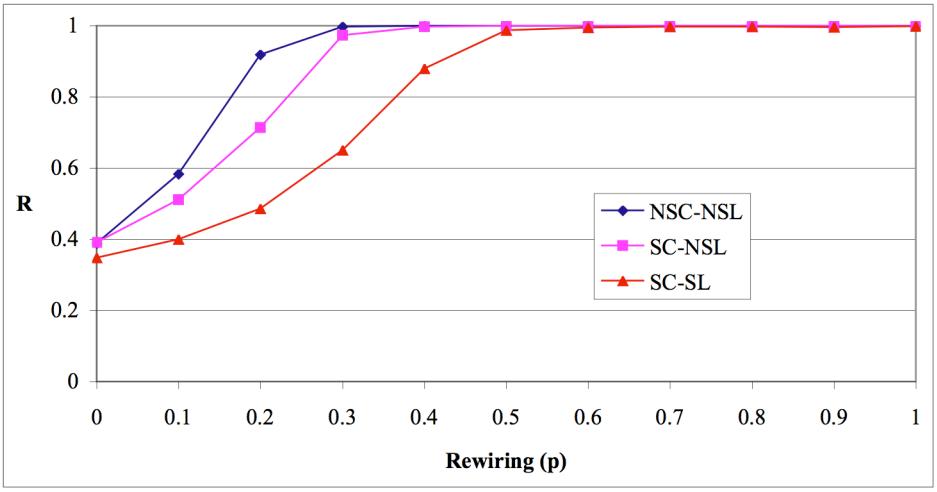


Figure 7: 1000 unit, sparsely connected, small world networks with 60 input connections per node and a training set of 18 random patterns. Results are averages over 50 runs at each value of the rewiring parameter, *p*. The attractor performances, *R*, for each of the three types of network is shown. See Table 1 for an explanation of the legend.

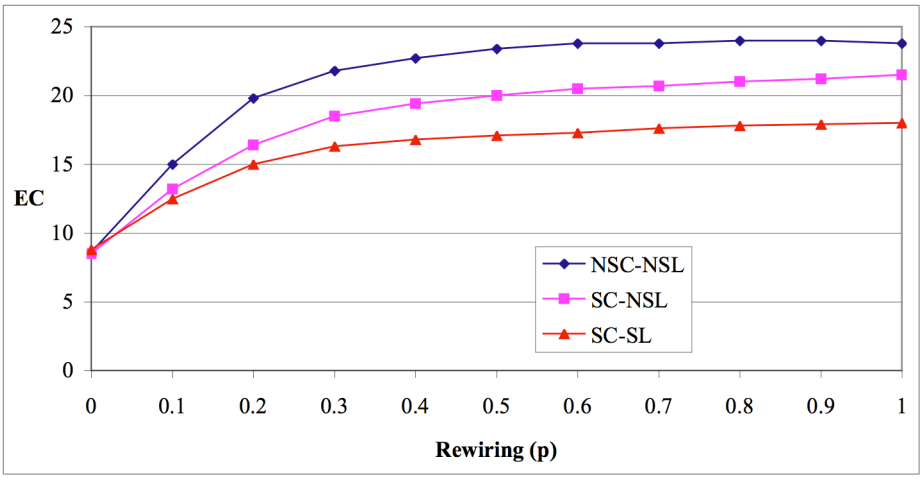


Figure 8: 1000 unit small world networks with 60 input connections per node. Results are averages over 50 runs at each value of the rewiring parameter, *p*. *EC* values for each of the three types of network are shown.

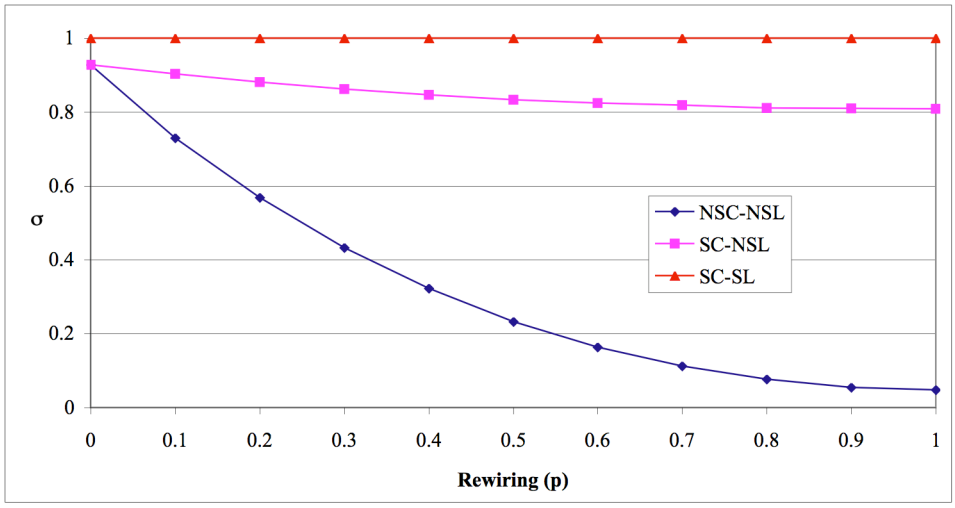


Figure 9: 1000 unit small world networks with 60 input connections per node and a training set of 18 random patterns. Results are averages over 50 runs at each value of the rewiring parameter, *p*. Symmetry values, σ, for each of the three types of network are shown.

3.1.2 Random models

The results from the previous section show that with small world connectivity the fully symmetric model exhibits the poorest performance and the non-symmetric version the strongest. The networks examined here have quite low connectivity, since we have only 60 connections per unit in a 1000 unit network. In order to investigate whether the relative performance applies to random networks and to explore what happens as the networks become more fully connected we measured *R* as we progressively increased the connectivity rate for each of the three network models. Figure 10 shows the results.

In Figure 10 we used 100 patterns and started each model at 100 connections per unit in a 1000 unit network. Initially with this large number of patterns to store, all three network models failed to pattern-complete. As the number of random connections was increased, all three network models increased their *R* value, with the fully symmetric model lagging behind (at 200 connections *SC-SL* was only at $R = 0.25$, whereas the others have *R* values between 0.5 and 0.6). Eventually all three networks attained $R = 1$ by 400 connections, which still represents a connectivity rate of 0.4.

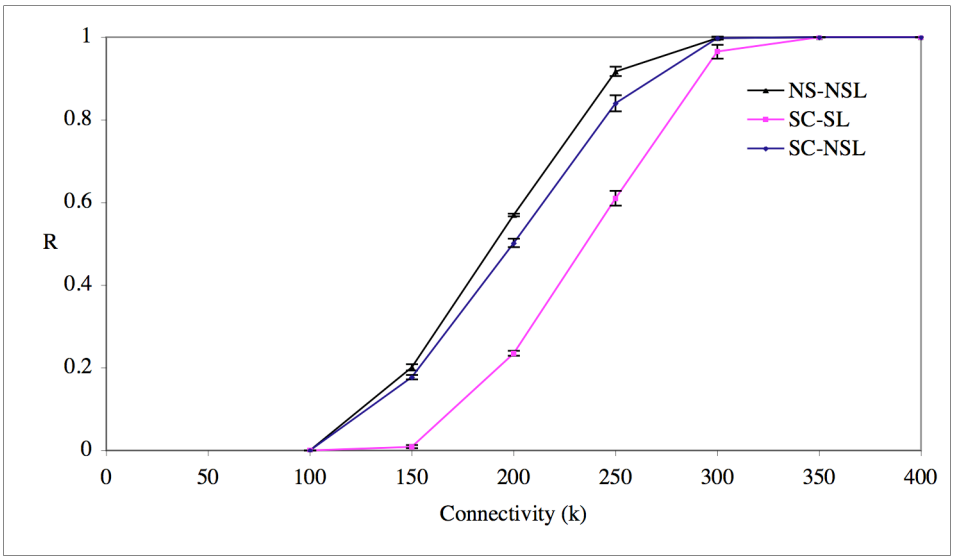


Figure 10: 1000 unit randomly connected networks with increasing input connections and a training set of 100 random patterns. Results are averages over 10 runs at each value of the connectivity parameter, *k*. The attractor performances, *R*, for each of the three types of network are shown. 95% confidence limits are shown.

Both of these subsections have demonstrated that the fully non-symmetric model, *NSC-NSL*, performs better than its symmetric counterparts. Hence we now focus exclusively on this model for all of our remaining results on small world networks.

3.2 Performance of Small world Networks

This section contains our principal results on small world networks, where we investigate how the values of network size, *N*, connectivity, *k*, and rewiring parameter, *p*, determine the performance of the network. In the first subsection we vary *k* and *p* and look at the resultant values of pattern completion ability, *R*, and Effective Capacity, *EC*. In the second subsection we perform a subset of this analysis with much larger networks.

3.2.1 Effects of different connectivity and rewiring

The results in this subsection involve investigating the loading differences in small world networks. Figures 11 and 12 show the results obtained for *R* and *EC* respectively, when the models are rewired from a locally connected initial configuration. Graphs are shown for 20, 40 and 60 connections per unit. In Figure 11 the loading is the same for all three graphs, α, is set to 0.3. This means that different numbers of patterns are used in each case. For $k = 20$, there are 6 patterns, for $k = 40$ there are 12 while $k = 60$ has 18 patterns. The Effective Capcity results presented in Figure 12 do not have this complication: the results show the number of patterns that are effectively stored. Both Figures show the expected results, with the higher connectivity graph ($k = 60$) displaying the strongest

performance. As the rewiring is increased performance improves. More rewiring is needed to produce perfect pattern completion for the lower connectivity set-up (k = 20). The effective Capacity plot shows that more patterns can be effectively stored with higher connectivity, but all of the graphs have basically flattened off by the same amount of rewiring, about $p = 0.5$.

A close look at the comparison between Figures 11 and 12, shows that if perfect performance is desired, as measured by the pattern completion ability measure, R (see Figure 11), much more rewiring is needed for k = 20 than for k = 60 to get values of R approaching 1. This is probably due to an artefact of the low connectivity and the way R is defined: a set of perceptrons will be likely to have similar performance if each unit has a relatively large number of inputs. At the lower connectivity value (k = 20) the performance of some of the units in the network may be significantly below the average, and R requires correction by *every* unit.

If, however, a few incorrect bits are tolerated, as with Effective Capacity (see Figure 12), then the amount of rewiring needed to get more or less the top value is the same for all connectivity values at about 40-50% rewiring. This should be taken as the more reliable result.

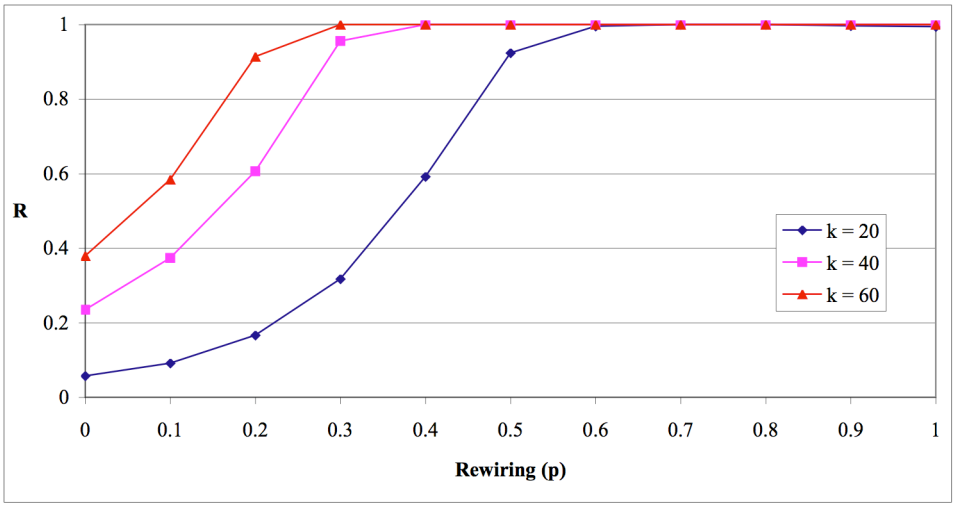


Figure 11: 1000 unit small world networks with 20, 40 and 60 input connections per node and a training set of 6, 12 and 18 random patterns respectively (giving a loading of 0.3 in each case). Results are averages over 10 runs at each value of the rewiring parameter, p. The attractor performance, R, for each of the connectivities are shown.

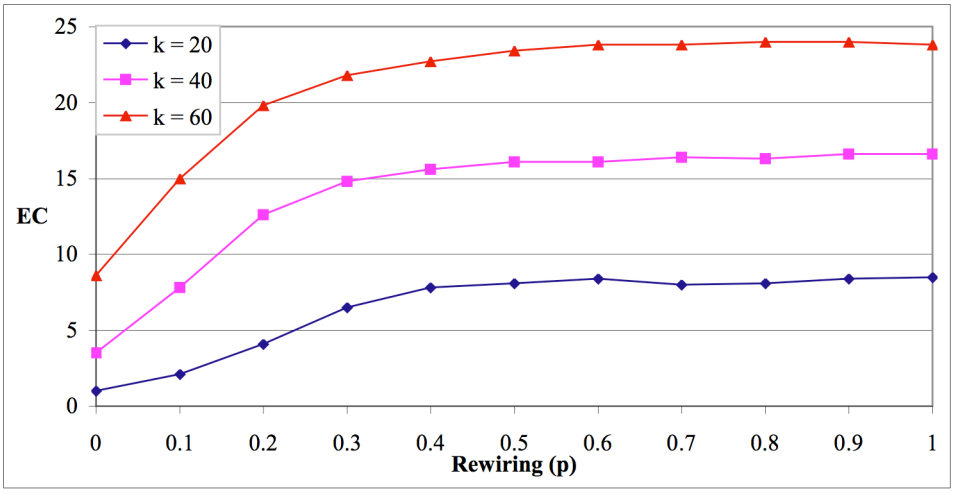


Figure 12: 1000 unit small world networks with 20, 40 and 60 input connections per node. Results are averages over 50 runs at each value of the rewiring parameter, *p*. *EC* values for each of the connectivities are shown.

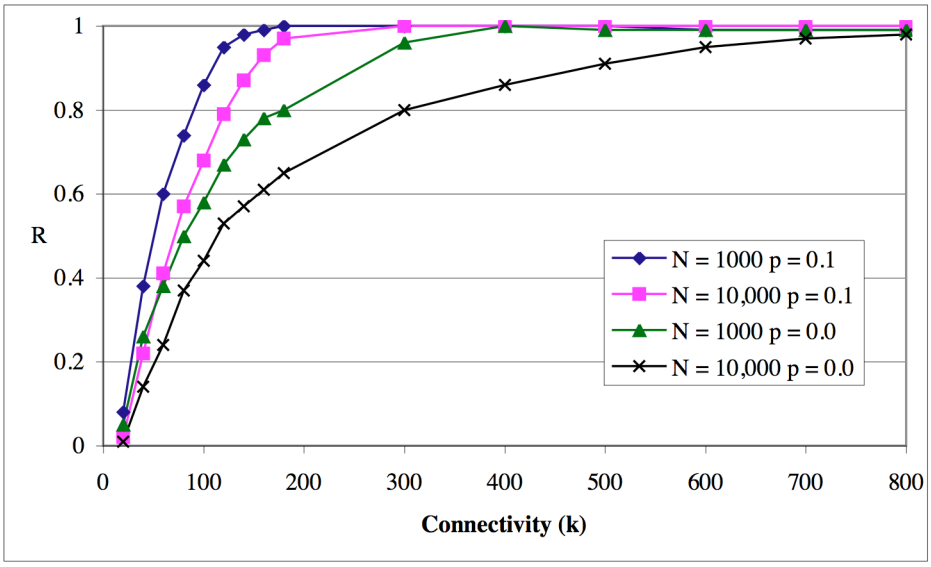


Figure. 13. Networks of size $N = 1000$ and $N = 10{,}000$, either with local connectivity only $p = 0$ (lower two plots), or with $p = 0.1$, (upper two plots). The attractor performance, *R*, is reported for different values of connectivity *k*. In all cases the loading of the network is $\alpha = 0.3$. Results are averages over 10 runs.

3.2.2 Network size effects

In the final experiments we investigated the effect of changing the size of the network. Figure 13 shows the results. Here the usual 1000 unit network is shown alongside a

network one order of magnitude larger, at 10,000 units. The two lower plots are for the networks with purely local connectivity, while the upper plots have just 10% rewiring. As can be seen, the small amount of rewiring has a dramatic effect on the pattern completion performance, in that both the upper graphs reach perfect pattern completion ($R = 1$) faster and earlier than the graphs of the purely locally connected networks.

Comparing the 1000 unit network against the 10,000 unit network it is seen that for both values of p the 1000 unit network performs the best at each value of connectivity, k. Even so, all networks have reached perfect pattern completion ($R = 1$) by 800 connections per unit. The reason that the larger network appears to be worse is that, for this extremely large network size, the number of connections relative to network size is extremely small. This means that it is hard for information to propagate through the network at such a low connectivity relative to network size. In fact if you consider that the larger, 10,000 unit, network, with only 10% rewiring, still manages to perform almost perfectly with only 180 connections, it represents quite an achievement. In this configuration the network is storing 54 10,000-ary vectors, with each unit having only 180, mostly local, connections and it is still able to perform almost perfect pattern completion. A randomly connected network at this connectivity level of would perform just as well. However, here we have almost perfect pattern completion with only a wiring length of about 12% of that of a random graph with this same level of connectivity.

4 Related Work

As described earlier in Section 2.8, the performance of the standard, sparse Hopfield network with a specific connection matrix, can be predicted from the relationship between the first two eigenvalues of this matrix. Surprisingly this technique has not been applied to the subsequently proposed sparsely connected Hopfield networks.

The first suggestion for using small world architectures with associative memory neural networks is in (Bohland, and Minai, 2001). They empirically evaluate standard Hopfield networks of size $N = 1000$, with $k = 150$ and a loading of 0.15 (25 training patterns). They confirmed that the local network has poor performance and that with a rewiring of about $p = 0.5$ good performance is restored (on both contiguous and random correction). Similar results, with sparser networks, are presented in (McGraw, and Menzinger, 2003), where $N = 5000$ and $k = 50$. They also give results for networks with a scale-free pattern of connectivity, which do a little worse than the random networks. Similar networks are also used in (Morrelli, Abramson, and Kuperman, 2004) with $N = 5000$ and $k = 100$. Interestingly, the results suggest the network performance peeking at $p = 0.4$ and then falling a little, with the particular measure, *efficacy*, that they use. Another interesting piece of work described in (Kim, 2004) takes the connectivity matrix of *C. Elegans* (discussed earlier in 2.7) and builds a Hopfield network with this connectivity. Since the average degree of this network is roughly $k = 14$, it was also possible to build small world and scale free Hopfield networks with roughly similar connectivity. It is found that the *C. Elegans* Hopfield network performs quite well: in fact better than a $p = 0.1$ small world network, but not as well as a random network. It is also shown that if the clustering coefficient of the network graph is modified (by swapping edges between two pairs of

connected vertices), then performance can be either improved, by decreasing clustering, or worsened by increasing clustering. Small world networks, with biologically inspired, integrate and fire neurons have also been investigated, and here also, fairly high levels of rewiring are needed for good performance (Anishchenko, Bienenstock, and Treves, 2005).

Hopfield networks with scale free patterns of connectivity have also been investigated in: (Perez Castillo, Wemmenhove, Hatchett, Coolen, Skantzos, and Nikoletopoulos, 2004; Stauffer, Aharony, da Fontoura Costa, and Adler, 2003; Torres, Munoz, Marro, and Garrido, 2004)

5 Discussion

Finding good patterns of connectivity in sparse recurrent networks is an interesting problem. Two competing factors need to be balanced: firstly the need for information to propagate globally in the network implies the need for distal connectivity, but secondly the desire for economical wiring promotes the reverse objective. In this paper we have shown that it is possible to find a workable balance by having the majority of the connectivity local but also with a significant fraction between random locations. Our results correspond with those reported for the standard Hopfield model, in that we generally require about 40% random connectivity for reasonably optimal pattern correction. It is worth noting that these levels of rewiring give networks that have many more random connections than the original examples of small world networks. Nonetheless there is still a considerable benefit in having more than half the connections between close neighbours. The efficacy of rewiring is not, apparently, lost when the network becomes very large, as can be seen in the 10,000 unit network (Figure 13) when the local and $p = 0.1$ networks are compared.

One of the interesting features of these large, but sparsely connected networks is that they show a clear performance difference between symmetric and non-symmetric weight conditions. This is notable because in the normal, fully connected network there is no difference in performance between the two types of weight conditions, with symmetrical weights being considered preferable due to the clean dynamics. It is thought that, in general, real neuronal systems do not have symmetric connectivity (Braitenberg *et al.*, 1998).

We have, to some extent, answered our original research question: it is possible to produce efficiently wired associative memory networks with good functionality, using the Watts and Strogatz inspired rewiring approach. Our results, however, raise another more fundamental question: what is the *most* parsimonious network configuration, that is, what sort of network connectivity has both minimal wiring length and good performance. Early results suggest that configurations other than the small world networks may work even better. In (Adams, Calcraft, and Davey, 2005) we report on the connectivity found by a genetic algorithm (GA) attempting to minimise wiring length and optimise

performance. The result was a pattern of connectivity in which the probability of a connection between two units fell in a roughly linear way with distance. However the networks that the GA evaluated were much smaller than those discussed in this paper and direct comparisons cannot be made. We are currently undertaking further work to throw more light on this fundamental question.

References

Adams, R., Calcraft, L., and Davey, N. (2005). "Evolving High Capacity Associative Memories with Efficient Wiring." Paper presented at the IJCNN, Montreal, 2005.

Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985). Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks. *Physical Review Letters* 55, 1530-1533.

Amit, D. J. (1989). *Modeling Brain Function: The world of attractor neural networks*. Cambridge University Press, Cambridge.

Anishchenko, A., Bienenstock, E., and Treves, A. (2005). Autoassociative Memroy Retrieval and Spontaneous Activity Bumps in Small-World Networks of Integrate-and-Fire Neurons. *Submitted to Neural Computation*

Barabasi, A., Albert, R., and Leong, H. (1999). Scale-free characteristics of random networks: the topology of the world wide web. *Physica A: Statistical Mechanics and its Applications* 272, 173-187.

Bohland, J., and Minai, A. (2001). Efficient Associative Memory Using Small-World Architecture. *Neurocomputing* 38-40, 489-496.

Bollobas, B. (2001). *Random Graphs*. Cambridge University Press.

Braitenberg, V., and Schüz, A. (1998). *Cortex: Statistics and Geometry of Neuronal Connectivity*. Springer-Verlag, Berlin.

Calcraft, L. (2005). *Measuring the Performance of Associative Memories* (Report Number 420). University of Hertfordshire.

Chengxiang, Z., Dasgupta, C., and Singh, M. P. (2000). Retrieval Properties of a Hopfield Model with Random Asymmetric Interactions. *Neural Computation* 12, 865-880.

Cherniak, C. (1994). Component placement optimization in the brain. *J. Neurosci.* 14, 2418-2427.

Cherniak, C., Mokhtarzada, Z., Rodriguez-Esteban, R., and Changizi, K. (2004). Global optimization of cerebral cortex layout. *PNAS* 101, 1081-1086.

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers* EC-14, 326-334.

Davey, N., and Adams, R. (2004a). High Capacity Associative Memories and Connection Constraints. *Connection Science* 16, 47-66.

Davey, N., Hunt, S. P., and Adams, R. G. (2004b). High capacity recurrent associative memories. *Neurocomputing* 62, 459-491.

de Nooy, W., Mrvar, A., and Batagelj, V. (2005). Exploratory Social Network Analysis with Pajek. In *Structural Analysis in the Social Sciences*. Cambridge University Press.

Eguiluz, V. M., Chialvo, D. R., Cecchi, G. A., Baliki, M., and Apkarian, A. V. (2005). Scale-Free Brain Functional Networks. *Physical Review Letters* 94,

Floréan, P., and Orponen, P. (1993). Attraction radii in binary Hopfield nets are hard to compute. *Neural Computation* 5, 812-821.

Gardner, E. (1988). The space of interactions in neural network models. *Journal of Physics A* 21, 257-270.

Gorodnichy, D. O. (Year). "The optimal value of self-connection." Paper presented at the International Joint Conference on Neural Networks (IJCNN'99), Washington, DC, 1999, 1999.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, Redwood City, CA.

Hilgetag, C., and Kaiser, M. (2004). Clusetered Organization of Cortical Connectivity. *Neuroinformatics* 2, 353-360.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America - Biological Sciences* 79, 2554-2558.

Kanter, I., and Sompolinsky, H. (1987). Associative Recall of Memory Without Errors. *Physical Review A* 35, 380-392.

Keller, E. F. (2005). Revisiting "scale-free" networks. *BioEssays* 27, 1060-1068.

Kepler, T. B., and Abbott, L. F. (1988). Domains of attraction in neural networks. *Journal Physique de France* 49, 1657-1662.

Kim, B. J. (2004). Performance of networks of artificial neurons: The role of clustering. *Physical Review E* 69,

Komlos, J., and Paturi, R. (1993). Effect of Connectivity in an Associative Memory Model. *Journal of Computer and System Sciences* 47, 350-373.

Kosinski, R. A., and Sinolecka, M. M. (1999). Memory Properties of Artificial Neural Networks with different types of dilutions and damages. *Acta Phsica Polonica B* 30, 2589-2594.

Laughlin, S. B., and Sejnowski, T. J. (2003). Communication in Neuronal Networks. *Science* 301, 1870-1874.

McGraw, P., and Menzinger, M. (2003). Topology and computational performance of attractor neural networks. *Physical Review E* 68, 047102.

Milgram, S. (1967). The Small World Problem. *Psychology Today* 60-67.

Morrelli, L. G., Abramson, G., and Kuperman, M. N. (2004). Associative Memory on a small-world neural network. *The European Physical Journal B* 38, 495-500.

Nardulli, G., and Pasquariello, G. (1991). Domains of attraction of neural networks at finite temperature. *Journal of Physics A: Mathematical and General* 24, 1103.

Newman, M. E. J. (2000). Models of the Small World. *Journal of Statistical Physics* 101, 819.

Noest, A. J. (1989). Domains in Neural Networks with Restricted-Range Interactions. *Physical Review Letters* 63,

Perez Castillo, I., Wemmenhove, B., Hatchett, J. P. L., Coolen, A. C. C., Skantzos, N. S., and Nikoletopoulos, T. (2004). Analytic solution of attractor neural networks on scale-free graphs. *Journal of Physics A: Mathematical and General* 37, 8789.

Personnaz, L., Guyon, I., and Dreyfus, G. (1986). Collective Computational Properties of Neural Networks: New Learning Mechanisms. *Physical Review A* 34, 4217-4228.

Shefi, O., Golding, I., Segev, R., Ben-Jacob, E., and Ayali2, A. (2002). Morphological characterization of in vitro neuronal networks. *Physical Review E* 66,

Sporns, O., and Zwi, J. D. (2004). The small world of the cerebral cortex. *Neuroinformatics* 2, 145-62.

Stauffer, D., Aharony, A., da Fontoura Costa, L., and Adler, J. (2003). Efficient Hopfield pattern recognition on a scale-free neural network. *European Physical Journal B* 32, 395-399.

Storkey, A., and Valabregue, R. (1999). The basins of attraction of a new Hopfield learning rule. *Neural Networks* 12, 869 - 876.

Torres, J. J., Munoz, M. A., Marro, J., and Garrido, P. L. (2004). Influence of topology on the performance of a neural network. *Neurocomputing* 58-60, 229-234.

Watts, D., and Strogatz, S. (1998). Collective Dynamics of 'small-world' networks. *Nature* 393, 440-442.