Associative Memories with Small World Connectivity

Neil Davey, Lee Calcraft, Bruce Christianson, Rod Adams
Department of Computer Science, University of Hertfordshire, UK
E-mail: {N.Davey, L.Calcraft, B.Christianson, R.G.Adams}@herts.ac.uk

Abstract
In this paper we report experiments designed to find the relationship between the different parameters of sparsely connected networks of perceptrons with small world connectivity patterns, acting as associative memories.

1 Introduction
It is possible to build associative memory models from networks of simple perceptrons. These networks perform much better than the canonical Hopfield model, both in terms of capacity and pattern completion. It is also possible to use such networks of perceptrons with sparse or diluted connectivity, and the performance is relatively robust, even at high rates of dilution. Of course real neural networks have sparse connectivity (for example in the cortex of the mouse each neuron is connected to about 0.1% of the other neurons [1]), which motivates the investigation undertaken here. It is also known that in biological systems the networks have a small world characteristic [2, 3]. That is they exhibit short path lengths (the minimum number of nodes on a path) between any pair of neurons, as in a random network, but also show a cliquish behaviour, with locally clustered connections. The advantage of such clustering is apparent in the mean connection length (the average wiring length), which is far smaller than in an equivalent random network, see Figure 1. A further benefit is that the wiring complexity is much reduced in networks with predominantly local connections. Recent research has shown that a small world Hopfield network could be a relatively effective associative memory [4]. In our earlier paper [5] we showed how the high capacity model could benefit from small world connectivity. Here we investigate the detailed relationship between the nature of the connection graph, the size of the network and the resulting performance. Sections 2 and 3 describe the background and the computational model. After the performance measure is explained in Section 4 the results are given in Section 5. The paper finishes with a discussion.

2 Background
The simple small world model of Watts and Strogatz [2] consists of a regular N-node ring lattice. Each node is connected to k/2 neighbours on either side, where k is typically small compared to N. The mean path length between any pair of random points is therefore high. A fraction, p, of these local connections is then rewired to randomly selected nodes, see right diagram in Figure 1. They showed that at surprisingly low values of p, the mean path length in the network dropped dramatically, resulting in a small world regime: highly clustered but with low path lengths. Many real networks have been shown to have a small world architecture, including the internet, human acquaintance networks and real networks of neurons [2]. Theoretical work [6] has now shown the detailed relationship between the characteristics of such networks.

Much is known about the effect of sparse connectivity on the standard Hopfield neural network. Some work has also been undertaken on diluted Hopfield networks with modular and small world connectivity [4, 7-9]. For the higher capacity version of the Hopfield network, trained using perceptron learning much less is known about the effect of connectivity patterns. It was shown in [10] that capacity falls linearly with dilution and in [11] that structured local connectivity could help in storing locally correlated data.

The sparse network with only local connectivity can be considered as a simple example of a Cellular Neural Network and it has been proposed [12] that such networks can be used as associative memories.

Fig. 1. A ring, with random connectivity on the left and small world connectivity on the right.
3 Network Model

The high capacity models studied here are a modification of the standard Hopfield network. The net input, or local field, of a unit, is given by:

\[ h_i = \sum_{j} w_{ij} S_j \]

where \( S (\pm 1) \) 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: \( S_i = \Theta(h_i) \), where \( \Theta \) is the heaviside function. Unit states may be updated synchronously or asynchronously. Here we use asynchronous, random order updates. If a training pattern \( \xi^{p} \) is a fixed point of the dynamics 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 positive.

We examine sparse networks with small world connectivity. The network topology is similar to the original Watts and Strogatz model. We start with an \( N \)-ring regular lattice, with each unit connected to its \( k \) nearest neighbours, and then rewire with probability \( p \). However as a network of perceptrons is not necessarily constrained to have symmetric connections we can generalize the network to a weighted directed graph (as is the case for real neural networks). The rewiring does not therefore maintain the symmetry of connectivity of the original regular lattice. In fact it has been shown [5] that for sparse networks of this type, symmetric weights give rise to poor performance.

The networks are trained using the normal perceptron training rule:

1. Begin with zero weights
2. Repeat until all local fields are correct
   - Set state of network to one of the \( \xi^{p} \)
   - For each unit, \( i \), in turn:
     - Calculate \( h_i^{p} \). If this is less than \( T \)
       - then change the weights to unit \( i \) according to:
         \[ w'_{ij} = w_{ij} + \frac{\xi^{p}_{i} \xi^{p}_{j}}{k} \]
     - Where \( \xi^{p} \) denotes the training patterns, and \( T \) is the learning threshold which here has the value of 10.

4 Performance Measure

We are interested in how well the small world networks and random networks, trained using the perceptron style learning rule described above, perform as associative memories. The capacity of such networks is determined by the number of incoming connections \( k \) that each perceptron has. For random pattern sets a perceptron can learn up to \( 2k \) patterns [13]. Assuming roughly regular connectivity graphs (as is the case here) the capacity will be determined by the level of dilution and not the specific pattern of connections, and hence is not subject to empirical investigation.

We use \( R \), the normalised mean radius of the basins of attraction, as a measure of attractor performance in these networks. It is defined as:

\[ R = \frac{1 - m_0}{1 - m_l} \]

where \( m_0 \) is the minimum overlap an initial state must have with a fundamental memory for the network to converge on that fundamental memory, and \( m_l \) is the largest overlap of the initial state with the rest of the fundamental memories. The angled braces denote a double average over sets of training patterns and initial states. Details of the algorithm used can be found in [10]. A value of \( R = 1 \) implies perfect performance and a value of \( R = 0 \) implies no pattern correction.

5 Results

In this paper we report experiments designed to find the relationship between the different parameters of the model. We summarise these parameters in Table 1.

| \( N \) | The size of the network
| \( k \) | The number of connections each unit makes
| \( \alpha \) | The loading per connection: size of training set / \( k \)
| \( p \) | The proportion of rewired connections

5.1 Fixed Size Networks

The first set of experiments fixes the network size, \( N \), at 1000 units, arranged in a ring as described earlier. Initially each unit is connected to its \( k = 20, 40 \) or 60 neighbours. Random training sets of 1000-ary vectors are created. The number of vectors in the training set is determined by the specific values of \( \alpha \) and \( k \). For example with \( \alpha = 0.3 \), \( k = 20 \) implies a training set of 6 vectors, \( k = 40 \) 12 vectors and \( k = 60 \) 18 vectors (as used in Figure 3). The attractor performance (\( R \) value) is then measured as the network is progressively rewired, as described above. All results presented are averages over 10 runs.

Figure 2 gives the results. Considering first the overall pattern, it can be seen that, in all cases, as the amount of rewiring is increased the performance of the network is also improved. In fact all the networks reach a point at which pattern correction behavior is