Using a Genetic Algorithm to Investigate Efficient Connectivity in Associative Memories

Rod Adams, Lee Calcraft, Neil Davey,
School of Computer Science
University of Hertfordshire
College Lane, Hatfield
Hertfordshire AL10 9AB
{R.G.Adams, L.Calcraft, N.Davey}@herts.ac.uk

***Abstract*—** We investigate sparse networks of threshold units, trained with the perceptron learning rule to act as associative memories. The units have position and are placed in a ring so that the wiring cost is a meaningful measure. A Genetic Algorithm is used to evolve networks that have efficient wiring, but also good functionality. It is shown that this is possible, and that the connection strategy used by the networks appears to maintain some connectivity at all distances, but with the probability of a connection decreasing rapidly with distance.

***Index Terms*—** Associative Memory, Neural Network, Genetic Algorithm, Small World Network, Connectivity.

I. Introduction

In real neuronal networks the position of the neurons, and the pattern of connectivity, appears to be highly optimized to minimize the total amount of wiring (Sporns, Chialvo et al. 2004). For example in the nematode worm, assuming that the number of neurons is fixed and that the sensors and actuators are in appropriate positions, the neurons are actually placed in almost exactly the right position for the most efficient wiring (Cherniak 1994). Normally in artificial neural networks the neurons are not considered to have a geometric position and so considerations of efficient wiring are not relevant, rather the nature of the resulting computation is of interest. However, when the network is intended to be physically instantiated then, of course, position and wiring costs become important. For example in cellular neural networks the units are normally placed in a grid and are connected locally (Brucoli, Carnimeo et al. 1995).

This paper is concerned with sparsely connected variants of the standard Hopfield neural network, intended to act as associative memories, in that they can pattern-correct efficiently. Most of the original work on Hopfield Networks was done on fully connected networks (Hopfield 1982; Davey, Hunt et al. 2004), here we are only interested in sparsely connected networks, since real neuronal networks are clearly very sparse (Braitenberg and Schüz 1998). Results can be produced for sparsely connected artificial neural networks that indicate the best type of connectivity, when wiring costs are ignored. For instance: when an artificial neural network is to perform as an associative memory then it is known that using only local connectivity leads to poor performance, with local domains of errors not being corrected (Noest 1989; Davey, Christianson et al. 2004); and with sparsely connected networks it is well known that ones with regular random connections will perform best as an associative memory (Davey, Christianson et al. 2004) – see Table 1. However, here we are interested in attempting to determine the best wiring configuration for an associative memory, when wiring costs have a bearing on the selection of the architecture.

In the work we report here we allow the networks to be relatively free to choose the most suitable connectivity matrix and we investigate the efficiency of different connectivity patterns in a high capacity associative memory model by evolving networks using a *genetic algorithm* (GA). Our major finding is that a Gaussian-like distribution of connection lengths gives networks with good performance and minimal wiring. The rest of this paper is organised as follows. The next section looks at connectivity in networks, including real neuronal networks, as reported in the literature. The next three sections look at the associative memory model, the fitness function used in the genetic algorithm and the actual connectivity of the model. The two following sections give details of the results for two different experiments, and the final section offers some conclusions.

II. Connection Strategies

In recent years much has been discovered about the pattern of connectivity of the neurons in real neuronal networks (Braitenberg and Schüz 1998; Hellwig 2000; Karbowski 2002; Shefi, Golding et al. 2002; Laughlin and Sejnowski 2003; Buzsaki, Geisler et al. 2004; Cherniak, Mokhtarzada et al. 2004; Hilgetag and Kaiser 2004; Sporns, Chialvo et al. 2004; Sporns and Zwi 2004). A good deal of this research has explored the way that specific neuronal circuits operate and how functional areas in the brain are connected. Another approach, however, has been to examine the connectivity matrix at the level of individual neurons or of functional areas, and to look for patterns of connectivity familiar in other contexts. This ties in with the large amount of current interest in small-world and scale-free networks. In this paper we look at this second issue in the context of associative memory networks. We summarize some of what is currently known about the connectivity matrix in real neuronal systems and show how the search for optimal patterns of connectivity in artificial associative memory models may shed light on some of the issues that real neuronal systems must deal with.

The connectivity in real neuronal systems, such as the mammalian cortex, is quite different from that found in most artificial neural networks. Real neuronal systems have vast numbers of neurons connected to only a fraction of the other neurons. For example the human cortex has about 10^{11} neurons with each connected to, on average, 10,000 other neurons (Braitenberg and Schüz 1998). In such *sparse* networks the connection strategy employed has to balance two competing goals. Firstly the total amount of neuronal fiber should be minimized, both because it is biologically expensive and because increasing length implies increasing difficulty in finding physical paths for the fiber, in the brain. Secondly, however, information needs to travel efficiently throughout the system for fast global computation to take place. The connectivity satisfying both goals is therefore highly likely to be have been optimized by evolution and will be far from random (Buzsaki, Geisler et al. 2004).

a. Non-Random Graphs

Before looking at some of what is currently known about the actual connectivity strategy employed in real systems, we examine what is known about connection graphs in general.

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, however, there has been an explosion of interest in networks with non-random connectivity graphs, such as *small-world* and *scale-free* networks.

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 of each other. 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, from local networks, that showed the small world effect. Their idea was to begin with a local network and then to *rewire* a small proportion, p, of the connections to random targets. Even at 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. We refer to networks constructed in this way as *small-world networks*.

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 (Newman 2000): 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.

Other interesting networks that show the small world effect are so called *Scale-Free* networks (Barabasi, Albert et al. 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.

b. 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, an average of 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 et al. 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 once again that positioning is highly optimised to minimise connection length (Laughlin and Sejnowski 2003; Cherniak, Mokhtarzada et al. 2004; Hilgetag and Kaiser 2004). 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 (Sporns and Zwi 2004; Eguiluz, Chialvo et al. 2005).

At the level of individual neurons the connectivity pattern is so complex that only generalised statistics can be produced. These show that in the mouse cortex, for example, there are about 1.6 million neurons, with each connected to, on average, about 8000 other neurons (Braitenberg and Schüz 1998). The density of connectivity is impressive, with approximately a billion synapses in each cubic millimetre of cortex. Most of the connections are local, with the probability of any two neurons in the same area being connected falling off in a Gaussian-like manner (Hellwig 2000), see Figure 1. It is thought extremely unlikely that these intra-area connections are highly structured (Braitenberg and Schüz 1998) as they are added at the rate of about 40,000 a second as the cortex matures. Cortical connectivity is of particular interest, as it is likely that one major function of the cortex is to act as a very large associative memory (Braitenberg and Schüz 1998).

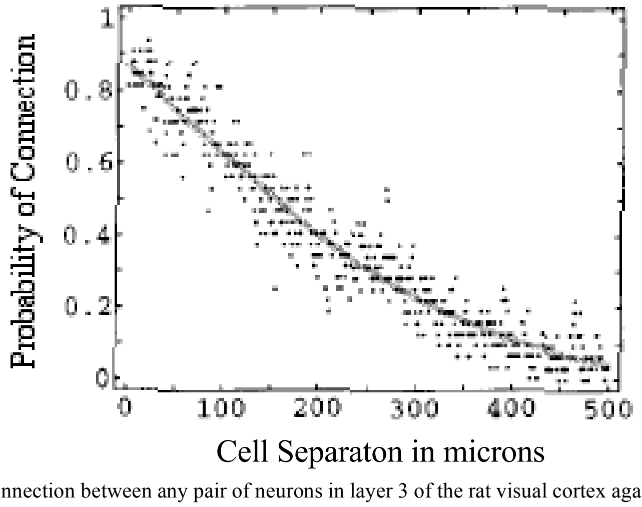

Figure 1. The probability of a connection between any pair of neurons in layer 3 of the rat visual cortex against cell separation. The horizontal axis is marked in μm. Taken from (Hellwig, 2000).

III. The Associative Memory Model

The particular neural network used here is a high performing variant of the canonical Hopfield model (Hopfield 1982). A set of perceptrons is sparsely interconnected with no attempt to ensure symmetry of connectivity. However, the networks are regular – each node has K incoming connections. No self connections are allowed. Training is performed using the standard perceptron learning rule, and the dynamics is governed by asynchronous random-order updates. To be specific:

The net input, or *local field*, of a unit, is given by: $h_i = \sum_{j \neq i} w_{ij} S_j$ where S (± 1) is the current state and w_{ij} is the weight on the connection from unit j to unit i (zero if no connection exists). The dynamics of the network is given by the standard update: $S_i' = \Theta(h_i)$, where Θ is the Heaviside function.

The networks are trained using a variant of the normal perceptron training rule. The learning algorithm (Diederich and Opper 1987) is:

```
Begin with zero weights
Repeat until all local fields are correct
Set state of network to one of the $\xi^p$
For each unit, i, in turn:
        Calculate $h_i^p \xi_i^p$.  If this is less than T
        then change the weights to unit i
        according to:
                $\forall j \neq i \quad w_{ij}' = w_{ij} + \frac{\xi_i^p \xi_j^p}{K}$
```

Where ξ^p denotes the training patterns, and T is the learning threshold, which here has the value of 10. Whilst such a trained network will not satisfy the normal requirements for simple dynamics, in practice these models perform well (Davey and Adams 2004).

In our model we need the units to have a geometry, so that there is a distance defined between any pair of units. We take the simplest approach (as in the original small world model (Watts and Strogatz 1998)) and place the units in a 1-D ring. The distance between any two nodes on the ring is simply the minimal number of steps along the ring to get from one to the other (see Figure 2). Consequently, for any given connectivity, we can calculate the total wiring length by adding up the distances between the connected nodes. Finally the mean wiring length, L, is calculated by dividing by the number of units and the number of connections to each unit. This gives the mean length between connected nodes across the whole network.

As already stated the neural networks used here do not have full connectivity: in fact each unit is always connected to 20 other units – so K = 20. With random training sets, random connectivity is known to give the minimum path length (the mean number of steps from one node to another) and minimum clustering and this helps information to flow across the network. Hence, no connection strategy can better a random one, in terms of pattern completion capability (Davey, Christianson et al. 2004). However, it is known that there are other connection strategies which give a pattern completion performance as good as that of randomly connected networks, but with much reduced wiring length. In particular a so-called *small world* model (one with mostly local connections but also with some distal ones, Figure 2, right) (Watts and Strogatz 1998) can give good performance (Bohland and Minai 2001; Davey, Christianson et al. 2004). However, there may exist alternative connection strategies that are also effective. Here we begin the investigation into what sort of efficient connectivity can be evolved using a GA.

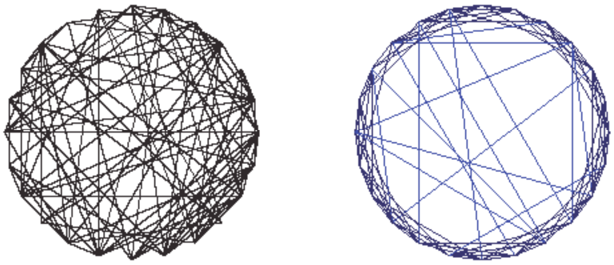

Figure 2: Units arranged in a simple 1-dimensional ring. On the left the units have random connectivity and on the right they have local connectivity and some distal connections – a *small world* model.

IV. The Fitness Function

We are interested in how well the 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 receives. For random pattern sets a perceptron can learn up to 2K patterns (Cover 1965). Assuming 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. These networks are often referred to as high capacity associative memories, since with full connectivity, the capacity is at least 2N (where N is the number of units) as against $0.14N$ for the standard Hopfield model,

We are, however, interested in the pattern correction ability of the network and this is determined not only by loading but also by the nature of connectivity. So we measure, R, the *normalised mean radius of the basins of attraction*, as a measure of attractor performance in these networks. It is defined as:

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

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_1 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 (Davey, Hunt et al. 2004). A value of $R = 1$ implies perfect performance and a value of $R = 0$ implies no pattern correction.

As already described, we also attempt to minimize the mean wire length in the network L. Two different fitness functions are used for the GA. First we attempt to minimize L alone, so $f_L = \frac{1}{L}$. Secondly we try to find networks with low L and high R so we used a fitness function of $f_{RL} = \frac{R}{L^n}$. Initial experimentation indicated that a value for n of 5 gave the appropriate balance between L and R for the networks used here. The reason for the large power of L relative to R needed in this function is that it is the changeability of these quantities that matter and not their actual size. R is calculated statistically and therefore has a fairly large variance in its values. For this reason results using R are usually averages over many different trials. In the GA experiments we only average the R value over 5 different pattern sets in order to reduce the time it takes to measure each network's fitness, consequently R still has a fairly large potential variance in its values. Random fluctuations in R can be compensated by increasing the contribution of L to the fitness function, so reducing the tendency for the GA to find, as its fittest individual in a generation, the best R variation. In order to give a more realistic performance indication at the end of the GA run both an initial random network and the final fittest network are tested using 100 different pattern sets to get a more accurate value for R. It is these final values that are given in the results section. Also, for comparison purposes, a random initial network and the final fittest individual have their *Effective Capacity (EC)* calculated and compared.

The Effective Capacity (Calcraft 2005) 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. In other words, it is a capacity measure that takes into account the dynamic ability of the network to perform pattern correction. We take a fairly arbitrary definition of *reasonable* as the ability to correct the addition of 60% noise to within an overlap of 95% with the original fundamental memory. Varying these two percentage 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 connectivity pattern.

The Effective Capacity 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 then *P-1.* The Effective Capacity of the network is therefore the highest pattern loading for which a 60% corrupted pattern has, after convergence, a mean overlap of 95% or greater with its original value.

V. Connectivity In The Model

In the experiments described here we use networks with 50 units, 100 units and 400 units. The networks are relatively small, as many tens of thousands of them have to be trained and assessed in the process of evolving a fit one. As described previously, for each network size we connect each unit with 20 incoming connections, so *K* is always 20. Any smaller value for *K* leads to unwanted edge effects. The training sets consisted of between 6 and 9 random patterns depending on the network size, the number of patterns being picked so that the *R* value in the trained network had an approximately intermediate value (about 0.4 - 0.6). It is only these values of *R* that can be used for meaningful comparison, since a value of 1 means that the network has no trouble pattern-correcting so few patterns and there is no variability in the experiment, and too low a value can mean that the task is too difficult.

Figure 3 shows a small version of two connectivity patterns on a set of 10 nodes arranged in a ring, with each node having 4 input connections. The first has a random connectivity and the second has local connectivity. The total wiring lengths, *L*, for the first node are shown in the diagram as 10 and 6, giving a mean wiring length of 2.5 and 1.5 for the connections to the first node. Continuing this calculation around all the nodes gives a total mean wiring length, *L*, for the whole network of 2.68 for the randomly connected network, and 1.5 for the locally connected network.

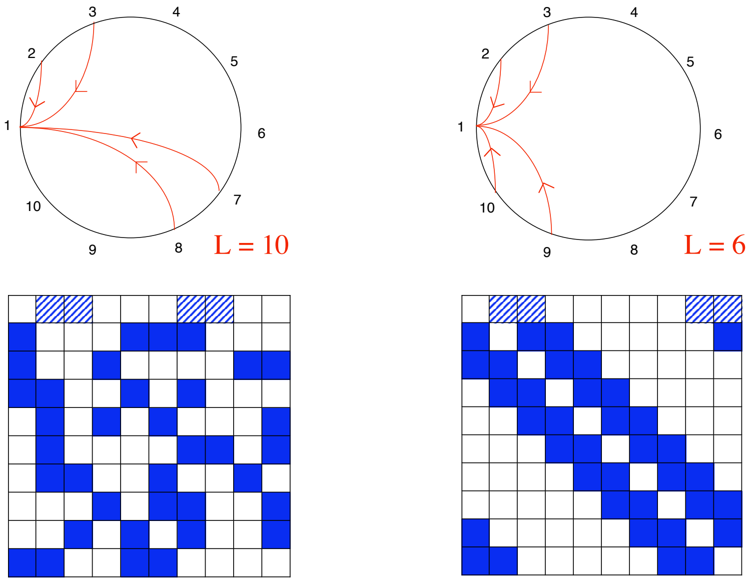

Figure 3: Two possible connectivity patterns: one random and one local. Connections to unit 1 are drawn. The corresponding connection matrices are given under the network. A coloured in square indicates a connection. Note that the top row (hashed) illustrates the input connections for unit 1. In both cases input connections come from units 2 and 3, the randomly connected one then has connections from units 7 and 8, whereas the locally connected one has connections from units 9 and 10. The L value gives the total wiring length for unit 1.

If the total connections to each unit, for each distance away, are counted then the local connectivity configuration will show 20 connections of distance 1 (2 per unit) and 20 connections of distance 2. For a network with 400 units and 20 input connections each, a local connectivity pattern will have 800 connections (2 per unit) at each of the distances out to 10 units away, giving a total of 8000 connections. If plotted as a histogram this gives the 'Local' plot in the left diagram in Figure 4. The random connectivity should give a fairly uniform histogram of connections with all distances being equally likely. This is the other plot in the left diagram in Figure 4. Figure 4 also shows two other connectivity histograms - a linear fall-off of connections with distance and an exponential fall-off of connections with distance. The purpose of this paper is to investigate which sort of connectivity gives both a good pattern completion ability for the associative network and also gives a minimum mean wiring length. As boundary values the Random and Local connectivity patterns for the 400 unit network with 20 inputs per unit are shown in Table 1. This table clearly shows that although a local connectivity pattern has a small mean wiring length, *L*, it comes at the cost of a considerably reduced normalised mean radius of attractor basin, *R*. These large differences are the major motivation for our study, namely can we maintain a relatively large *R* value while reducing the mean wiring length considerably from the maximal random connectivity value.

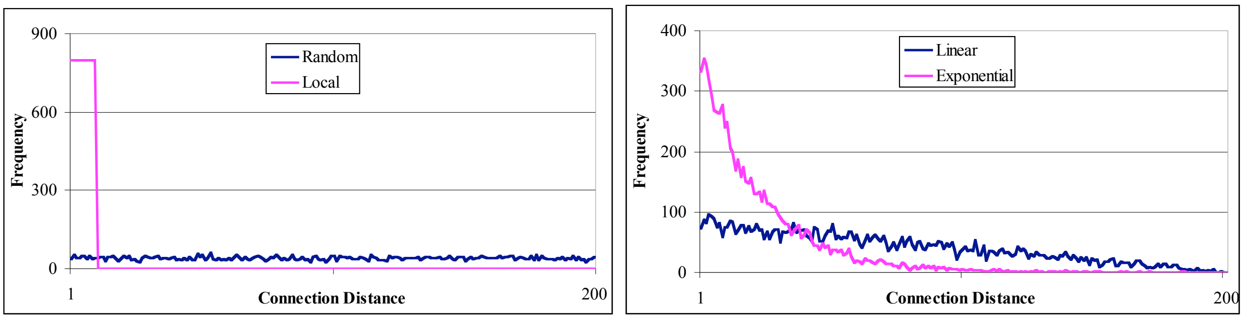

Figure 4: Histograms of the various connection strategies. The left diagram shows a Random connectivity pattern together with the Local connectivity pattern. The right diagram shows a Linear and Exponential fall-off of connections with distance. These are for 400 unit networks each with 20 input connections.

TABLE 1:
THE MEAN (OVER 100 RUNS) OF WIRING LENGTH, *L*, AND *R* VALUE FOR LOCAL AND RANDOM CONNECTIVITY PATTERNS ON A 400 UNIT NETWORK WITH 8 PATTERNS

	Random	Local
L	100.4	5.5
R	0.93	0.02

VI. THE FIRST EXPERIMENT

The first experiment investigates the evolution of an associative memory with effective pattern completion ability and minimum mean wiring length when allowing complete variability of connections between each unit of the network, subject to the 20 incoming connections constraint and no self-connections.

a. Genetic Algorithm

In this experiment the genome for each network consisted of a vector of size $N * N$ where N is the network size. Each element of the vector is a '1' or '0' representing the presence or absence of a connection. Essentially, the vector is a row by row representation of a connection matrix like that shown in Figure 3. So for the 400 unit networks the input was a 160,000 element representation of the connection matrix. Since each row of the connection matrix represents the incoming connections to the node given by that row then there will be 20 '1's in each row. Initially a population of 50 randomly configured networks is created, so that each network has a different random connectivity graph (subject to the 20 incoming connections constraint and no self-connections). Each network is then trained on a different randomly created training set, so that the GA does not learn to exploit particular training patterns, and the *R* and *L* values are calculated. The generation of training patterns is repeated five times and the *L* value and the average value of *R* are reported for each network. The fittest networks are then selected as the basis for the next generation. Crossover in the GA is constrained so that each unit in the offspring will still have 20 incoming connections – this is maintained by restricting crossover to occur only at boundaries representing complete sets of input connections to a unit, that is at row boundaries from the original connection matrix (see Figure 5). Any mutation that takes place is also constrained to maintain the same overall pattern of connections (see Figure 5).

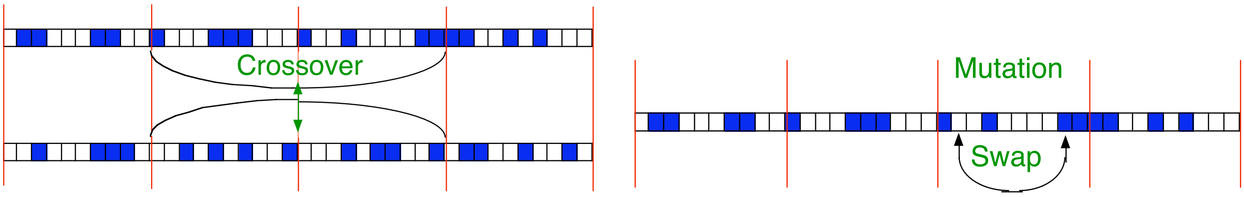

Figure 5: Left: a possible crossover is shown between two genomes representing part of the connection matrices for networks like those illustrated in Figure 2 with 10 nodes. Crossover is constrained to take place only at the boundaries of complete unit connections as shown. Again coloured squares represent connections, so that in each set of 10 possible connections there are just 4 connections. This diagram represents the first 4 sets of input connections (ie into nodes 1-4) so that it also illustrates that there are no self connections. Right: an illustration of a legal mutation. Two positions with the boundary of a complete unit's connection are swapped – this maintains the number of input connections to that unit (to 4 in this case).

The details of the GA used are as follows. Rank-based selection is used, with a structure length of $N * N$ bits, where N is 50, 100, 200 or 400. The crossover rate is 0.6 and the mutation rate is 0.001, dropping by a factor of 0.95 every

1000 generations. Replacement uses single element elitism. Typical runs of the GA showed the fitness level stabilizing after about 12,000 generations for N = 50 and 50000 generations for N = 400. The process is summarized as:

1. Create a population of 50 randomly connected networks.
2. Train each network 5 times with random training sets.
3. Evaluate *L* and the mean *R* for each network.
4. Select, crossover and mutate to form a new population.
5. Repeat from 2.

It is important to note here that only the pattern of connectivity is being evolved. A successful network will have a pattern of connectivity that can function well with any random training set. The networks are thus not evolving to perform well with a single, specific, set of training patterns.

b. Results

As indicated earlier, the GA assesses network performance relatively crudely, averaging *R* over just 5 training sets. In order to get a more reliable indication of final performance, all the results presented in this section are the averages of assessing the final, evolved networks with 100 new randomly generated training sets.

Initially we used networks with just 50 units in order to test the ability of the genetic algorithm to successfully optimise a network's connectivity given the large search space (the network matrix contains 50*50 elements). Our first experiment was undertaken as a simple test to see if good connectivity could be evolved for the relatively simple goal of having short wiring length – with fitness function f_L. Obviously the solution to this problem is simply for each unit to connect to its closest neighbours. The GA successfully evolved a network with a short wiring length – most of the connections being local ones. Obviously, because the fitness function ignores any network performance factors the optimized network had a significantly poorer performance (as measured by R) than that of the original random network. This test indicates that the GA was capable of quite easily optimisating the network connectivity even with such a large search space. This ability to optimise for a smaller mean wiring length is no doubt due to the fact that the fitness surface is convex and it is consequently easy to locate the minimum. Optimising for both minimum mean wiring length and *R* would be a far harder challenge. The success of the GA here encouraged us to continue with the f_{RL} fitness function and then to progress onto larger networks.

So next we used fitness function f_{RL} to optimize *R* and minimize *L*. In this case the GA was run five times for 12,000 generations. Table 2 gives the averaged results.

Table 2:
The result of trying to find a network with low *L* and high *R* - 50 Unit Network, using 6 patterns

	Initial Random	Best at Generation 12,000
L	12.78	9.36
R	0.50	0.48

The networks were evolved using 6 patterns, but it is useful to try out both the initial random network and the final evolved network on different numbers of patterns to make sure that there was no specific fit to a particular number of patterns. The results for between 2 and 8 patterns are shown in Table 3. Table 3 also shows the Effective Capacity, *EC*, of the networks at the start and at the end of the run. This result confirms the stability of the results shown in Table 2.

Table 3:
The result of trying to find a network with low *L* and high *R*. Network size 50 units. The last two columns show the effective capacity result of trying to find a network with low *L* and high *R*

Patterns	*L*-Start	*L*-End	*R*-Start	*R*-End	*EC*-Start	*EC*-Start
2	12.78	9.36	0.91	0.90	2.88	2.74
3	12.78	9.36	0.87	0.85		
4	12.78	9.36	0.76	0.75		
5	12.78	9.36	0.65	0.64		
6	12.78	9.36	0.50	0.48		
7	12.78	9.36	0.34	0.30		
8	12.78	9.36	0.18	0.15		

It can be seen that although *R* is maintained at about its original value, *L* is significantly reduced. The final connectivity matrix for the best individual is shown in Figure 6. Whilst there is a predominance of local connections, a fraction of distal connections are maintained.

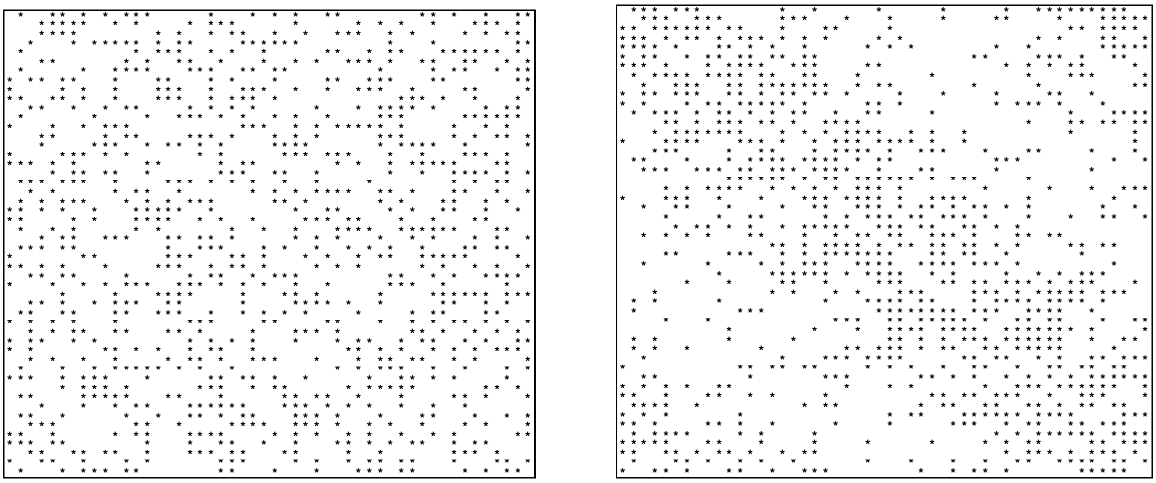

Figure 6: The disordered random connectivity matrix (left) is transformed into one with almost mainly local connections (right), when *L* is minimised and *R* is maximised with fitness function f_{RL}. Both diagrams are for a 50 unit network.

In order to further investigate the evolved connectivity pattern, a histogram of the connection lengths for the network optimized with f_{RL}, was produced, and is shown in Figure 7.

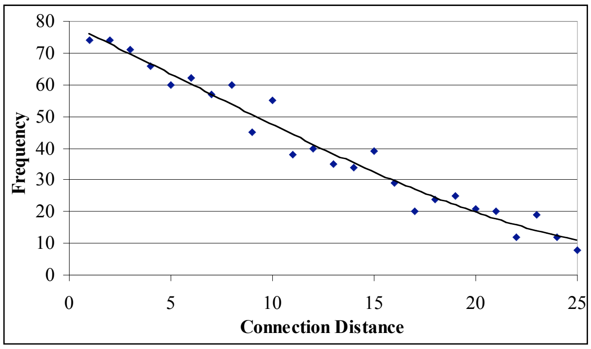

Figure 7: A histogram of the final connection lengths in a network with 50 units optimized for both low *L* and high *R*. The GA was evolved using 6 patterns. A trend line has been added for the f_{RL} plot to guide the eye.

The f_{RL} network has some connections of all lengths. Interestingly, the histogram shows a fairly linear decrease in frequency with length, as shown by the trend line. To find out if this was a general result we repeated the experiment to optimize *R* and minimize *L* using the fitness function f_{RL} using larger networks of size 100, 200 and 400 units, with *K* kept at 20 and a suitable number of random training patterns to give an evolvable *R* value as indicated earlier.

TABLE 4:
THE RESULT OF TRYING TO FIND A NETWORK WITH LOW L AND HIGH R IN THE 100 UNIT NETWORK - EVOLVED WITH 7 PATTERNS

	Initial Random	Best at Generation 40,000
L	25.28	19.70
R	0.65	0.65

As before, it can be seen in Table 4 that, although *R* is maintained at about its original value, L is again significantly reduced. Figure 8 gives the histogram of connection length in the evolved 100 unit and 200 unit networks, and figure 9 gives the histogram of connection length in the evolved 400 unit network. Here it is clear that the fall off in connection lengths is closer to an exponential decay, especially as we get to larger and larger networks. Basically, for the small network there is not enough 'space' between the size of the network (50 units) and the number of input connections (20) to show the real trend.

As before the final evolved network was tested for different numbers of patterns. Tables 5-7 give these results for the 100, 200 and 400 unit networks. In all cases the evolved network performed equally well for all the different numbers of patterns, maintaining nearly the performance level, as measured by *R*, as the original randomly connected network. Tables 5-7 also give the results for Effective Capacity, which confirm the previous results.

TABLE 5:
THE RESULT OF TRYING TO FIND A NETWORK WITH LOW L AND HIGH R IN THE 100 UNIT NETWORK. THE LAST TWO COLUMNS SHOW THE EFFECTIVE CAPACITY RESULT OF TRYING TO FIND A NETWORK WITH LOW *L* AND HIGH *R*

Patterns	*L*-Start	*L*-End	*R*-Start	*R*-End	*EC*-Start	*EC*-End
4	25.28	19.70	0.97	0.97	3.98	3.88
5	25.28	19.70	0.94	0.94		
6	25.28	19.70	0.85	0.84		
7	25.28	19.70	0.65	0.65		
8	25.28	19.70	0.42	0.39		
9	25.28	19.70	0.19	0.18		

TABLE 6:
THE RESULT OF TRYING TO FIND A NETWORK WITH LOW L AND HIGH R IN THE 200 UNIT NETWORK. THE LAST TWO COLUMNS SHOW THE EFFECTIVE CAPACITY RESULT OF TRYING TO FIND A NETWORK WITH LOW *L* AND HIGH *R*

Patterns	*L*-Start	*L*-End	*R*-Start	*R*-End	*EC*-Start	*EC*-End
6	50.25	30.6	0.98	0.98	5.44	5.38
7	50.25	30.6	0.94	0.92		
8	50.25	30.6	0.74	0.71		
9	50.25	30.6	0.42	0.39		
10	50.25	30.6	0.12	0.10		

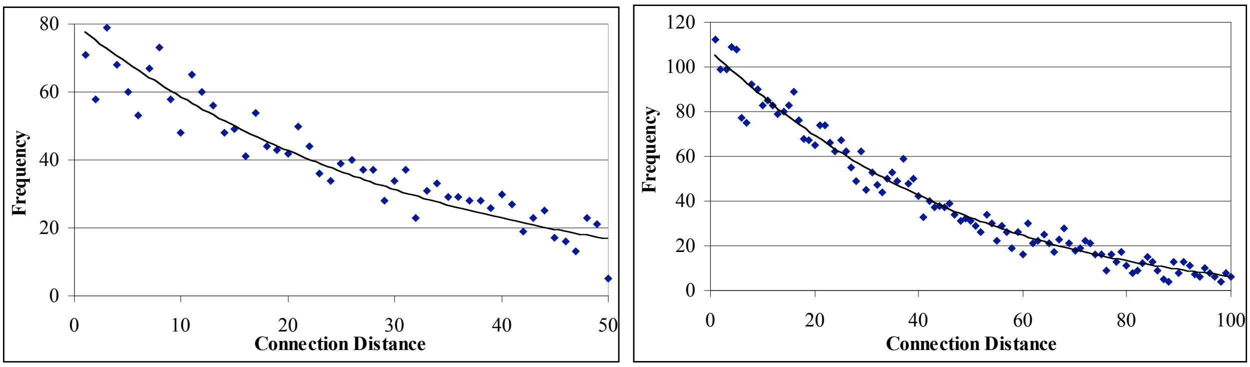

Figure 8: A histogram of the final connection lengths of a 100 and 200 unit network optimized for both low *L* and high *R*. The GA was evolved using 7 patterns for the 100 unit network and 9 patterns for the 200 unit network.

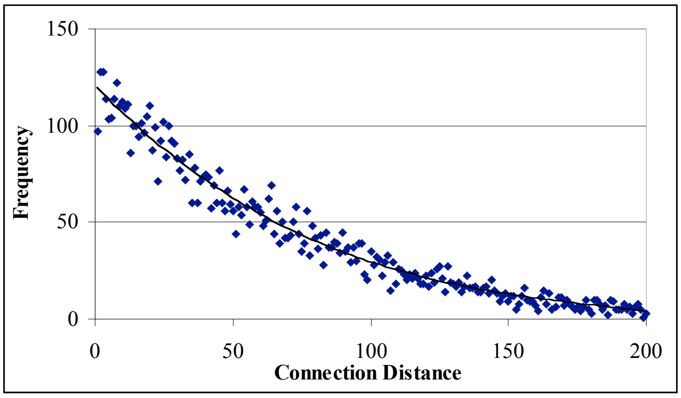

Figure 9: A histogram of the final connection lengths of a 400 unit network optimized for both low *L* and high *R*. The network was evolved using 9 patterns.

TABLE 7:
THE RESULT OF TRYING TO FIND A NETWORK WITH LOW L AND HIGH R IN THE 400 UNIT NETWORK. THE LAST TWO COLUMNS SHOW THE EFFECTIVE CAPACITY RESULT OF TRYING TO FIND A NETWORK WITH LOW *L* AND HIGH *R*

Patterns	*L*-Start	*L*-End	*R*-Start	*R*-End	*EC*-Start	*EC*-End
6	100.4	55.9	0.99	1.0	6.96	7.06
7	100.4	55.9	0.98	0.97		
8	100.4	55.9	0.93	0.92		
9	100.4	55.9	0.67	0.62		
10	100.4	55.9	0.27	0.21		

VII. THE SECOND EXPERIMENT

a. Genetic Algorithm

With the knowledge gained from the first GA experiment, we investigated various different types of hand-constructed connectivity patterns. Each gave a probability of connections that fell with the distance from the unit that was being connected to. The exponential distribution indicated in the first experiment here was used and others such as a Gaussian distribution, a restricted-linear distribution and a restricted-uniform distribution. Results of these experiments were reported in (Calcraft, Adams et al. 2006). This second GA experiment was motivated by the desire to see if the GA would give preference to a particular type of connectivity distribution and what sort of connectivity would result.

This experiment investigated evolving a network which could use any mix of four different connectivity strategies: *local, random, exponential* and *restricted-linear*. Random connectivity is known to give good pattern completion ability to the network, but has high mean wiring length. Local connectivity has low mean wiring length but poor pattern completion ability. The exponential connectivity distribution is illustrated in Figure 4, and the restricted linear is similar to the linear distribution illustrated in Figure 4, but with an intercept on the horizontal axis before the maximum possible length as shown in the Figure 4. A particular version, with intercept at 92 (46% of the maximum) is illustrated in Figure 12. In the exponential distribution the probability of a connection at distance *x* is given by:

$$p(x) = e^{-\lambda x} / \nu$$

where λ controls the tightness of the distribution and ν normalizes it.

In the restricted linear distribution the probability of a connection of length *x* is given by:

$$p(x) = \max\left(\nu - \frac{\nu}{\mu} x , 0 \right)$$

where μ is the maximum connection length and ν normalizes it.

Both the exponential and the restricted linear distributions perform well (Calcraft, Adams et al. 2007) - having a good balance of pattern completion ability and low mean wiring length.

In this experiment we took 400 unit networks with each unit having connections from 20 other units as before. Each unit had one of the four different connection patterns for **all** of its incoming connections, that is each unit was designated as having a distribution of its input connections that was either local, random, exponential or restricted linear. Hence there were four different kind of unit, designated by the sort of input distribution it had, and represented in the genome using the numbers 0-3. Each of the initial population of 50 networks was generated so that it had an equal number of each of these types of unit. So in a 400 unit network there were 100 of each type. However, the arrangement around the 1-D ring of these types of unit was random, so that the initial population had different random arrangements of the four types of unit. The genome was therefore just a list of 400 numbers giving the random arrangement of the four types of unit.

In the case of the exponential and restricted linear distributions a randomly generated exponent, λ, or restriction distance, μ, was also picked. For a given network, in the population, all of the linear type units used the same restriction distance μ, and similarly, all the exponential units used the same gradient multiplier λ. So each genome also had two further numbers specifying the exponential gradient multiplier λ and linear restriction μ to use, giving a total genome size of 402 numbers. The first 400 numbers were coded into 2 bits each (representing the 4 different connectivities) and the last 2 were coded into 5 bits each (giving 32 different values for each) and giving a total genome size of 810 bits. During evolution all aspects of the genome could evolve, therefore the numbers representing the type of each unit could evolve, so that the total number of each type of unit could vary and the arrangement round the 1-D ring of each type of unit would also vary as time progressed. The exponential slope and linear restriction values could also evolve. The GA was evolved using the same fitness function as before, namely f_{RL}. So we were attempting to minimise the mean wiring length and maximise *R*, to see how many of each type of connectivity prevailed in the final fittest genome and to see if the type of unit congregated in regions or not. It should be noted that the genomes only give the type of connectivity (random, exponential etc.) for each unit in the network; the actual connections of the network represented by the genome has to be generated afresh each time as the genome does not determine the specific connectivity matrix.

The details of the GA used are mainly as before. Rank-based selection is used, with a structure length of 810 bits, as described above. The crossover rate is 0.6 and the mutation rate is 0.001, dropping by a factor of 0.95 every 500 generations. Replacement uses single element elitism. Typical runs of the GA were for about 10,000 generations. The process is summarized as:

1. Generate a population of 50 genomes that are random arrangements of equal numbers of each of the four types of connectivity patterns (plus two numbers for the exponential slope and the linear restriction).
2. Generate the actual connections for each network from the genome.
3. Train each network 5 times with random training sets.
4. Evaluate *L* and the mean *R* for each network.
5. Select, crossover and mutate to form a new population.
6. Repeat from 2.

b. Results

The results can be seen in Figure 10. Here it is clear that the random distribution type of unit diminished in numbers fairly rapidly down to about 50% of their original number, meaning that only 50 units (out of 400) still had this type of connections. The local distribution units remained about constant and the other two distributions both increased equally strongly to about a 25% more than they were initially. After completing the GA run, the final best network was trained with one hundred different sets of patterns and compared with the values obtained at the start of the GA run on a random network chosen from the initial population - Table 8 gives the results and gives the equivalent Effective Capacity results. In this experiment, since we were not starting from a random connectivity, it was possible to reduce *L* and sometimes increase *R*, however the Effective Capacity measure remained constant.

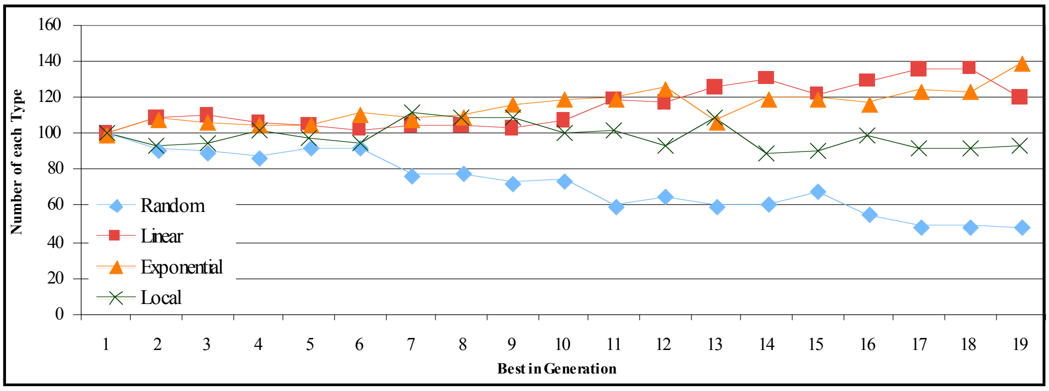

Figure 10. The number of units connected by each of four different connection strategies starting with equal numbers of all four (ie 100). The network has 400 units and the GA was optimising both high *R* and low *L*. The number of input connections is maintained at 20 for each unit. *R* is used in the fitness function.

TABLE 8: NETWORKS EVOLVED WITH LOW *L* AND HIGH *R*. NETWORK SIZE 400 UNITS. THE LAST TWO COLUMNS SHOW THE EFFECTIVE CAPACITY RESULT OF TRYING TO FIND A NETWORK WITH LOW *L* AND HIGH *R*

Patterns	*L*-Start	*L*-End	*R*-Start	*R*-End	*EC*-Start	*EC*-End
5	38.9	25.0	0.66	0.85	6.13	5.80
6	38.9	25.0	0.50	0.64		
7	38.9	25.0	0.35	0.41		
8	38.9	25.0	0.21	0.23		
9	38.9	25.0	0.06	0.05		

The best network had evolved exponential units with a fairly steep slope and the linear restriction favoured was a distance of 68 out of the maximum distance of 200. Figure 11 shows the distributions. This shows that the units connected by local, exponential and restricted linear had very few connections from units beyond a distance of 70. The only units with connections beyond this distance and up to the maximum of 200 units away were the remaining random ones. Since these units have been reduced to less than 50 of the total (ie about 12% of all connections) then there is clearly little distal connectivity needed to maintain a reasonable ability to pattern correct in the network.

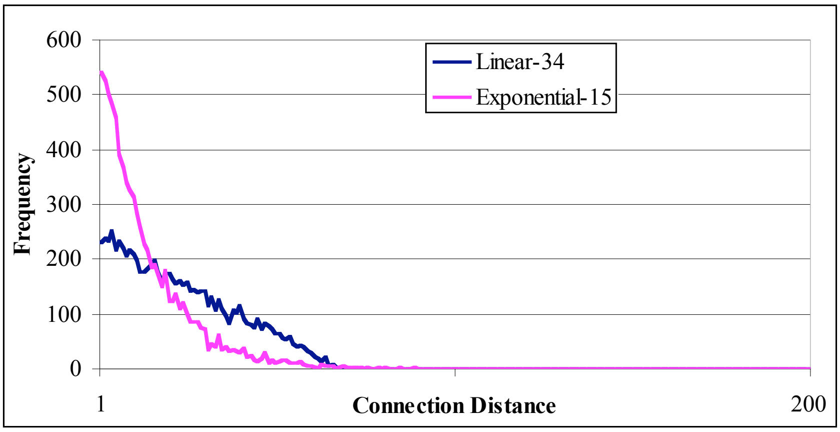

Figure 11: Histograms for the exponential and linear units in the final best network. Network size 400 units, network evolved for high *R* and low *L* using *R* in the fitness function.

Finally we repeated the last experiment using *EC* as the factor in the fitness function. Since *EC* does not have such a large variation we were able to use a fitness function of $f_{EL} = \frac{EC}{L}$. Figure 12 gives the results of the number of connectivity distributions during the run and Table 9 gives the values for *L* and *EC*. As can be seen this gives similar results to the previous GA run using *R* as the performance factor in the fitness function. The number of locally connected units is maintained at about 100, the Linear and Exponentially connected units have increased, and the locally connected units have decreased, but this time down to about 25 units.

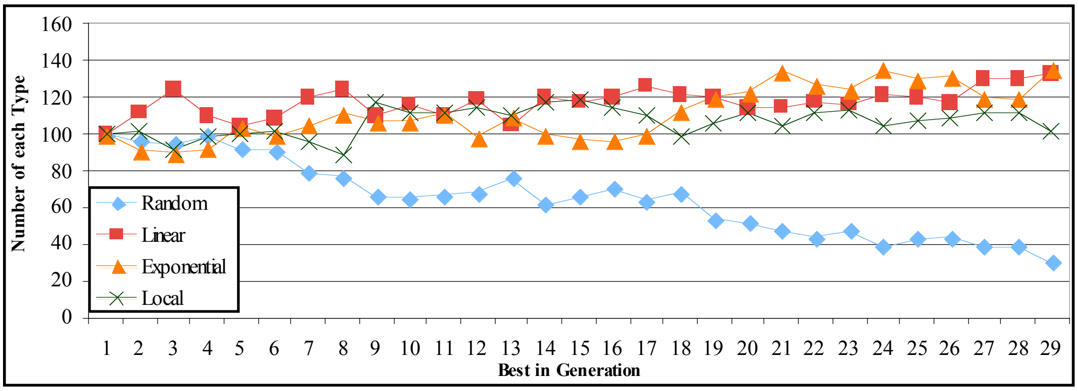

Figure 12: The number of units connected by each of four different connection strategies starting with equal numbers of all four (ie 100). The network has 400 units and the GA was optimising both high *EC* and low *L*. The number of input connections is maintained at 20 for each unit.

TABLE 9:
THE EFFECTIVE CAPACITY RESULT OF TRYING TO FIND A NETWORK WITH LOW *L* AND HIGH *EC*. NETWORK SIZE 400 UNITS.

L-Start	*L*-End	*EC*-Start	*EC*-End
38.9	23.3	6.30	5.90

Finally figure 13 shows the connectivity frequencies of the evolved linear and exponential units. In the experiment that used *EC* in its fitness function the restricted linear value was 92, which is larger than that found in the experiment that used *R* in the fitness function. This was compensated for by finding a steeper slope on the exponential units - 18 as against 15 previously.

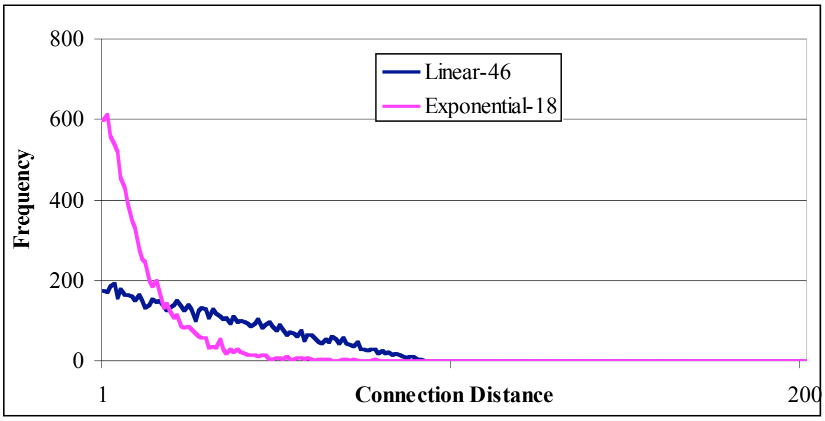

Figure 13: Histograms for the exponential and linear units in the final best network**.** Network size 400 units, network evolved for high *EC* and low *L* using *EC* in the fitness function.

Perhaps the most interesting finding of this experiment is the nature of the exponential and restricted linear distributions that were favored in the evolved network. Recall that the specific slope of the exponential and the cut-off of the linear distribution are parameters that the GA was attempting to optimize. The actual distributions evolved are shown in Figures 12 and 14. Both distributions are relatively tight: in one case with very few connections of length greater than 70 (Figure 12 - using *R* in the fitness function) and the other has very few connections of length greater than 90 (Figure 14 - using *EC* in the fitness function). The final configuration in this most successful network is therefore an interesting mixture of many local connections, a number of fairly local connections (from the exponential and linear units) and a few distal connections from the random units. This is illustrated in Figure 14.

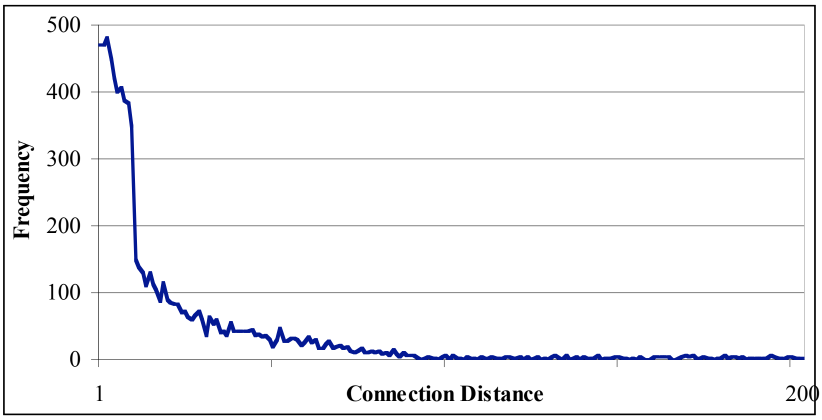

Figure 14: A histogram of the final configuration of the most successful network. It shows that the mixture of types of units in this network produce many local connections and very few distal connections.

Finally, the actual position around the ring of the four types of unit is shown in Figure 15. Here it is apparent that the various types of unit are distributed reasonably evenly around the ring.

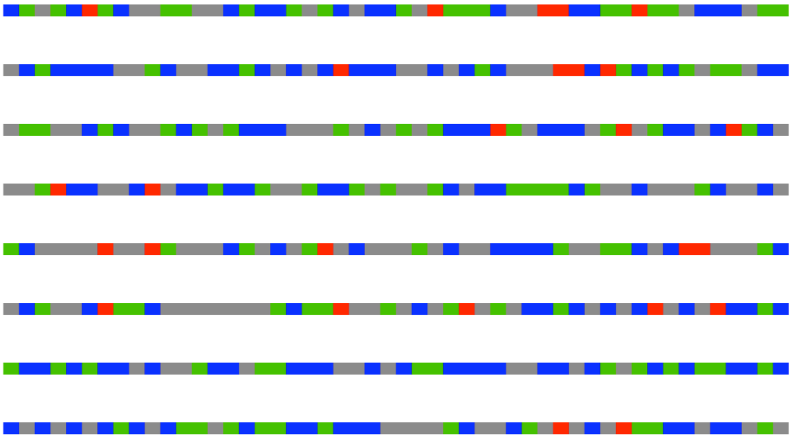

Figure 15: The spatial arrangement of the four different types of unit in the most successful network. The ring of 400 units is laid out in 8 rows of 50. The random nodes are coloured red, the local nodes blue, exponential grey and restricted linear green. It is apparent that the various types of unit are distributed reasonably evenly around the ring.

VIII. Conclusion

In earlier work (Davey, Christianson et al. 2004) we investigated high performance associative memories with a small-world architecture. The results presented here show that the evolved networks are at least as effective as the best of the small-world networks. Here the actual connectivity of the network is not specified, but is allowed to evolve using a genetic algorithm to balance the fitness of the network performance, as measured by both the normalized mean radius of attraction and the Effective Capacity, against the wiring length of the connected network.

Little attention has been paid, in the literature so far, to the issue of efficient wiring of neural networks. This is partly due to the fact that in a simulated neural network the actual wiring cost and complexity is of no concern, rather the nature of the resulting computation is of interest. It has only recently become possible to gather detailed information about the wiring of real neuronal networks, and the wiring appears to be far from arbitrary. Recent work has also shown that the connectivity pattern in sparse associative memories is important in determining functionality, and various connection strategies have been proposed (McGraw and Menzinger 2003; Stauffer, Aharony et al. 2003; Davey, Christianson et al. 2004; Torres, Munoz et al. 2004). In the work described here we allow the networks to be relatively free to choose the most suitable connectivity matrix, and by means of a GA to discover such an optimal connectivity with respect to the task in hand. Of course, evolutionary computation has been extensively applied to finding architectures (and weights) for neural networks (Yao 1999), but as far as we know no one has used these methods in the context used here.

In the first GA experiment the connectivity of the network was unconstrained (excepting that self-connections were not allowed and a fixed number of input connections were maintained for each unit) since a full connectivity matrix was used to describe the connectivity. Even with the small associative memories used here, the search space in which this experiment operates is vast: for instance, in the 100 unit network there are approximately 5.4×10^{22} possible connection matrices. So it was not clear at the outset of this research that a GA would succeed with even the simple task of optimizing f_L, by minimizing the wiring length. The results show, however, that it is possible to find connectivity that not only minimizes the mean wiring length, L, but also one that does this whilst maintaining effective pattern correction performance.

The strategy that the networks appear to use is to maintain some level of connectivity with nodes at all distances, with the probability of a connection being present falling rapidly with distance.

In the second experiment less freedom was allowed in the connectivity of the networks. Each unit had a specific connectivity pattern for all of its input connections, with four types of connection strategy being allowed (local, random, exponential and restricted-linear). Again the GA evolved networks with a mix of different connection lengths, but the majority of the connections were from local or mainly local units. Only a small fraction of connections were distal. Interestingly there was no clustering pattern to the distribution of these connection types, since the four types were distributed reasonably evenly around the ring.

As discussed earlier, the pattern of connectivity in real neuronal networks is likely to be highly optimized to make best use of expensive resources. We have shown here how a GA can be used to optimize resource use in an artificial neural network performing as an associative memory. Our main result is that distributions of connections that fall rapidly with distance, such as those in an exponential or restricted linear fashion perform best with low resource usage. Both evolutionary experiments found such distributions, surprisingly similar to those found in real neuronal systems (see Figure 1). So the results suggest that large-scale computation can be accomplished in networks with predominantly local connectivity. Further work reported in (Calcraft, Adams et al. 2007) confirms this suggestion with investigations of networks with many thousands of units. Moreover we have found that modular architectures that have similarity to cortical structures, such as mini-columns also produce parsimonious and well performing networks (Chen, Adams et al. 2008) We are currently investigating the generality of our results and we are evaluating the importance of connectivity in networks of spiking neurons.

However many interesting questions remain to be answered, such as: what are the developmental processes that contribute to the sort of architectures that occur in real neuronal systems?

References

Barabasi, A., R. Albert, et al. (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 A. Minai (2001). "Efficient Associative Memory Using Small-World Architecture." Neurocomputing **38-40**: 489-496.

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

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

Brucoli, M., L. Carnimeo, et al. (1995). Discrete-time cellular neural networks for associative memories: a new design method via iterative learning and forgetting algorithms. 38th Midwest Symposium on Circuits and Systems.

Buzsaki, G., C. Geisler, et al. (2004). "Interneuron Diversity series: Circuit complexity and axon wiring economy of cortical interneurons." Trends in Neurosciences **27**(4): 186.

Calcraft, L. (2005). Measuring the Performance of Associative Memories. Computer Science Technical Report, University of Hertfordshire.

Calcraft, L., R. Adams, et al. (2006). Gaussian and Exponential Architectures in Small World Associative Memories. ESANN. Bruge.

Calcraft, L., R. Adams, et al. (2007). "Efficient architectures for sparsely-connected high capacity associative memory models." Connection Science **19**(2): 163 - 175.

Chen, W., R. Adams, et al. (2008). Connectivity Graphs and the Performance of Sparse Associative Memory Models. To be published in the Proceedings of IJCNN, Hong Kong.

Cherniak, C. (1994). "Component placement optimization in the brain." J. Neurosci. **14**(4): 2418-2427.

Cherniak, C., Z. Mokhtarzada, et al. (2004). "Global optimization of cerebral cortex layout." PNAS **101**(4): 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 R. Adams (2004). "High Capacity Associative Memories and Connection Constraints." Connection Science **16**(1): 47-66.

Davey, N., B. Christianson, et al. (2004). High Capacity Associative Memories and Small World Networks. IJCNN, Budapest.

Davey, N., S. P. Hunt, et al. (2004). "High capacity recurrent associative memories." Neurocomputing **62**: 459-491.

Diederich, S. and M. Opper (1987). "Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules." Physical Review Letters **58**(9): 949-952.

Eguiluz, V. M., D. R. Chialvo, et al. (2005). "Scale-Free Brain Functional Networks." Physical Review Letters **94**: 018102.

Hellwig, B. (2000). "A quantitative analysis of the local connectivity between pyramidal neurons in layers 2/3 of the rat visual cortex." Biological Cybernetics **82**(2): 111.

Hilgetag, C. and M. Kaiser (2004). "Clustered Organization of Cortical Connectivity." Neuroinformatics **2**(3): 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.

Karbowski, J. (2002). "Optimal wiring in the cortex and neuronal degree of separation." Neurocomputing **44-46**: 875.

Keller, E. F. (2005). "Revisiting "scale-free" networks." BioEssays **27**(10): 1060-1068.

Laughlin, S. B. and T. J. Sejnowski (2003). "Communication in Neuronal Networks." Science **301**(5641): 1870-1874.

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

Milgram, S. (1967). "The Small World Problem." Psychology Today **2**: 60-67.

Newman, M. E. J. (2000). "Models of the Small World." Journal of Statistical Physics **101**(3/4): 819-841.

Noest, A. J. (1989). "Domains in Neural Networks with Restricted-Range Interactions." Physical Review Letters **63**(16): 1739-1742.

Shefi, O., I. Golding, et al. (2002). "Morphological characterization of in vitro neuronal networks." Physical Review E **66**(021905): 021905.

Sporns, O., D. R. Chialvo, et al. (2004). "Organization, development and function of complex brain networks." Trends in

Cognitive Sciences **8**(9): 418- 425.

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

Stauffer, D., A. Aharony, et al. (2003). "Efficient Hopfield pattern recognition on a scale-free neural network." European Physical Journal B **32**(3): 395-399.

Torres, J. J., M. A. Munoz, et al. (2004). "Influence of topology on the performance of a neural network." Neurocomputing **58-60**: 229-234.

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

Yao, X. (1999). "Evolving Artificial Neural Networks." Proceedings of the IEEE.