Evolving High Capacity Associative Memories with Efficient Wiring

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 connectivity at all distances, but with the probability of a connection decreasing linearly 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 [1]. For example in the nematode worm the neurons are actually placed in almost exactly the right position for most efficient wiring [2]. Normally in artificial neural networks the neurons are not considered to have a geometric position and so considerations of efficient wiring are not relevant. 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 [3].

When an artificial neural network is intended 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 [4, 5]. In the work we report here we attempt to find efficient connectivity in a high capacity associative memory model by evolving networks using a *genetic algorithm* (GA).

II. THE ASSOCIATIVE MEMORY MODEL

The particular neural network used here is a high performing variant of the canonical Hopfield model [6]. 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 [7] algorithm is:

```
Begin with zero weights
Repeat until all local fields are correct
 Set state of network to one of the ξ^p
 For each unit, i, in turn:
   Calculate h_i^p ξ_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 [8].

The units are given a simple spatial relation: they are arranged in a ring (see Figure 1). The distance between any two units in the ring is taken as the (minimum) number of steps on the ring to get from one unit to the other.

As already stated the neural networks used here do not have full connectivity: in fact each unit is connected to 20 other units – so $K = 20$. With random training sets, no connection strategy can better a random one, in terms of pattern completion capability [4]. 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 1, right) [9] can give good performance [4]. 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.

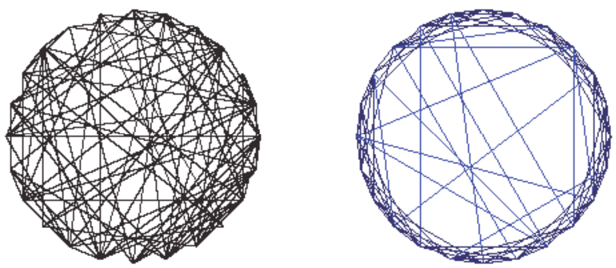

Figure 1: Units arranged in a simple 1-dimensional ring. On the left the units have random connectivity and on the right they have local connectivity and some distal connections – a *small world* model.

III. 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 [10]. 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 $2N$ (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 (5 used in each case) and initial states. Details of the algorithm used can be found in [11]. 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 total wire length in the network L. Three different fitness functions are used for the GA. First we attempt to minimize L alone, so $f_L = \frac{1}{L}$. We also attempt to maximize R alone, so $f_R = R$. Finally we try to find networks with low L and high R. Experimentation showed the fitness function $f_{RL} = \frac{R}{L^4}$ gave the appropriate balance between L and R for the networks used here.

IV. The Experiments

Two sizes of network were used in the experiments: firstly a 50 unit network and secondly one with 100 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. In both sizes of networks each unit had 20 incoming connections. The training sets consisted of six random patterns for the 50 unit networks and seven random patterns for the 100 unit networks.

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, and the R and L values are calculated. This is repeated five times and the average value of R is 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 (see Figure 2). Any mutation that takes place is also constrained to maintain the same overall pattern of connections.

The details of the GA used are as follows. Rank-based selection is used, with a structure length of 2500 bits (50 × 50) or 10,000 in the case of larger networks. The crossover rate is 0.6 and the mutation rate is 0.001 for the smaller network, dropping by 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. The process is summarized as:

1. Create a population of 50 random 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.

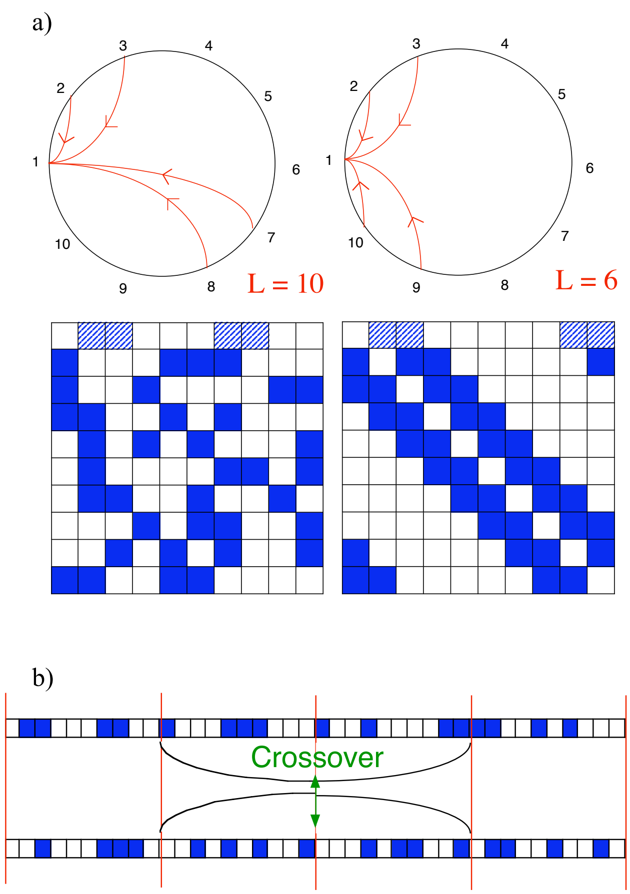

Figure 2:
a) Two possible networks: one with random connectivity and one with local connectivity. Connections to unit 1 are drawn. The corresponding connection matrices are given under the network. Note that the top row indicates the connections for unit 1.
b) A possible crossover is shown between two connection matrices. Crossover is constrained to take place only at the boundaries of complete unit connections as shown.

V. RESULTS

The GA assesses networks relatively crudely, averaging R over just 5 training sets. In order to get a more reliable indication of performance, all the results presented are averages of assessing the final networks with 100 new random training sets.

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. Table 1 gives the result of a run of 10,000 generations.

TABLE 1:
THE RESULT OF TRYING TO FIND A NETWORK WITH LOW L

	Initial Random	Best at Generation 10,000
L	12.78	6.05
R	0.50	0.26

It is apparent that it is possible to evolve networks with short wiring length, but because the fitness function ignores any network performance factors, we see, as expected, that the optimized network has a significantly poorer performance (as measured by R) than that of the original random network. The resulting connectivity matrix can be seen in graphic form in Figure 3. The initially random connectivity matrix is changed to one in which almost all the connections are local. Note that the leading diagonal is empty, as no unit is allowed a self-connection in these models.

The next experiment, attempting to maximize R, was run to confirm that no connection strategy can better a random one. The result confirms this, as can be seen in Table 2. It has not been possible for the GA to find a network that has an R value much better than the original random network.

TABLE 2:
THE RESULT OF TRYING TO FIND A NETWORK WITH HIGH R

	Initial Random	Best at Generation 10,000
L	12.78	12.82
R	0.50	0.50

Thirdly 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 3 gives the result.

TABLE 3:
THE RESULT OF TRYING TO FIND A NETWORK WITH LOW L AND HIGH R

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

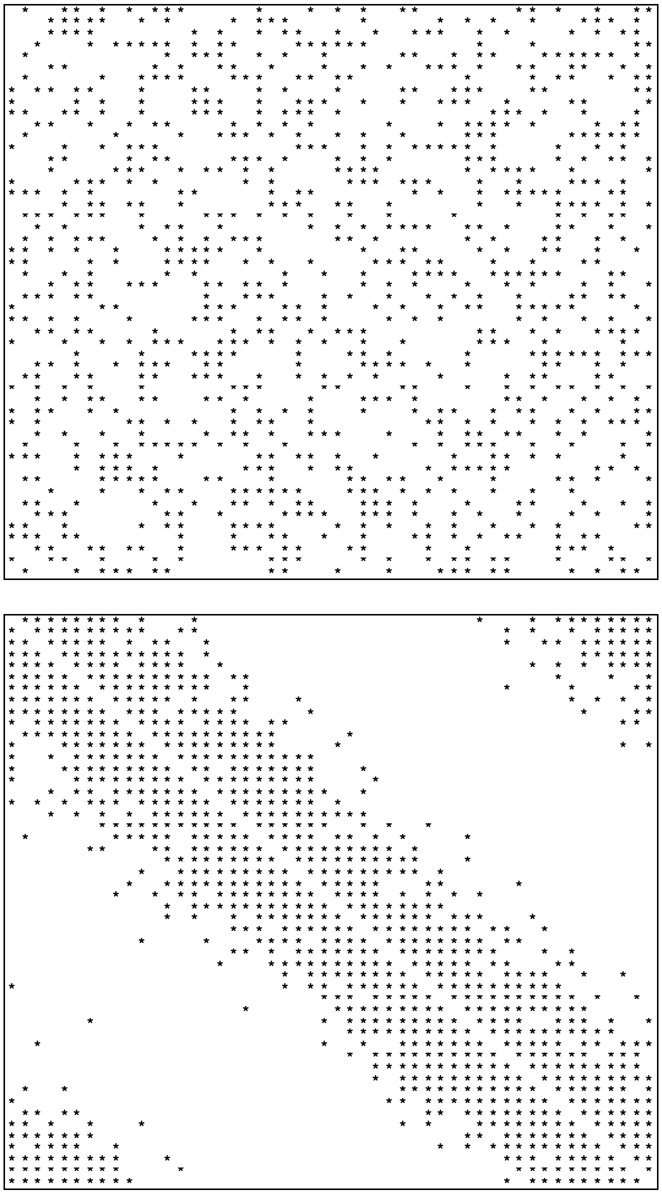

Figure 3: The disordered random connectivity matrix (above) is transformed into one with almost completely local connections (below), when L is minimized.

It can be seen that although R is maintained at about its original value, L is significantly reduced. The final connectivity matrix is shown in Figure 4. Whilst there is a predominance of local connections, a fraction of distal connections are maintained.

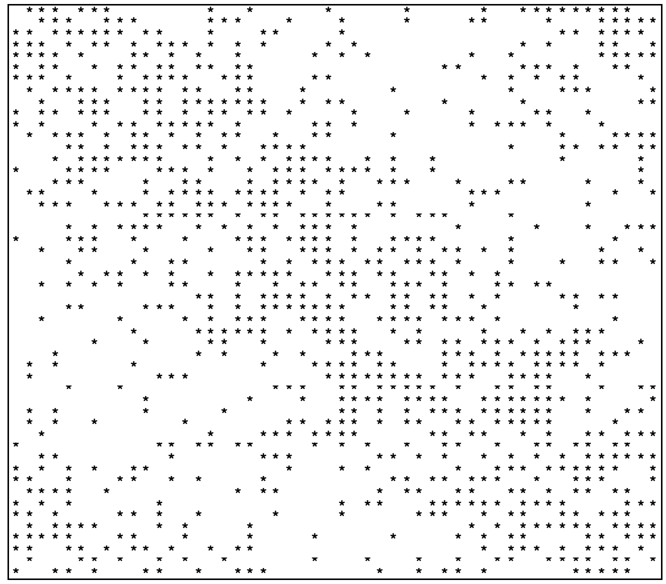

Figure 4: The connectivity matrix that results with fitness function f_{RL}

In order to further investigate the evolved connectivity pattern, a histogram of the connection lengths, for both the networks optimized with f_L and f_{RL}, was produced, and is shown in Figure 5.

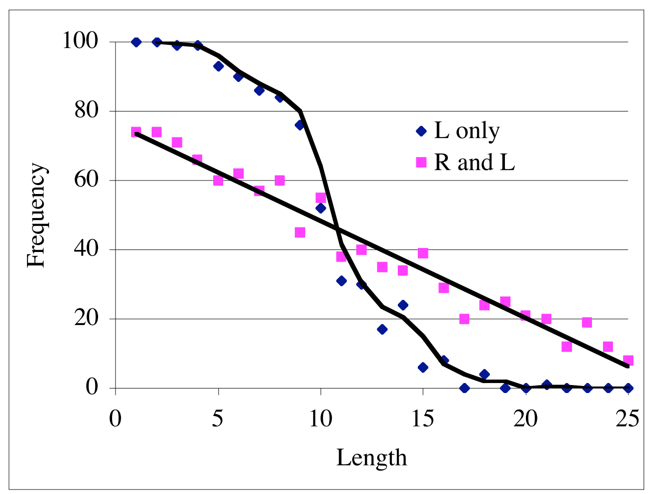

Figure 5: A histogram of the final connection lengths in a network optimized for low L (diamonds) and a network optimized for both low L and high R (squares). A trend line has been added for the f_{RL} plot, and a moving average for the f_L plot, to guide the eye.

A fully localized network would have a histogram that showed all 100 connections between units up to 10 units away being made, and no others. As expected the f_L network approximates to this pattern. The f_{RL} network, however, has some connections of all lengths. Interestingly, the histogram shows a fairly linear decrease in frequency with length, as shown by the trend line. In fact the Pearson

Correlation Coefficient for this data is 0.98.

The final experiment used the larger 100 unit network, with K kept at 20 and seven random training patterns.

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

	Initial Random	Best at Generation 40,000
L	25.24	16.39
R	0.65	0.58

As before it can be seen, in Table 4, that although R is maintained at about its original value (although here there is a small falloff), L is again significantly reduced. The resulting connectivity matrix is given in Figure 6, and shows a similar pattern to that in Figure 4. Figure 7 gives the histogram of connection length in the evolved 100 unit network. Once again we see a fairly linear fall in frequency with length (the Pearson Correlation Coefficient is 0.97). The pattern is quite similar to the smaller, 50 unit, network suggesting that this is a non-accidental feature of the evolved connectivity.

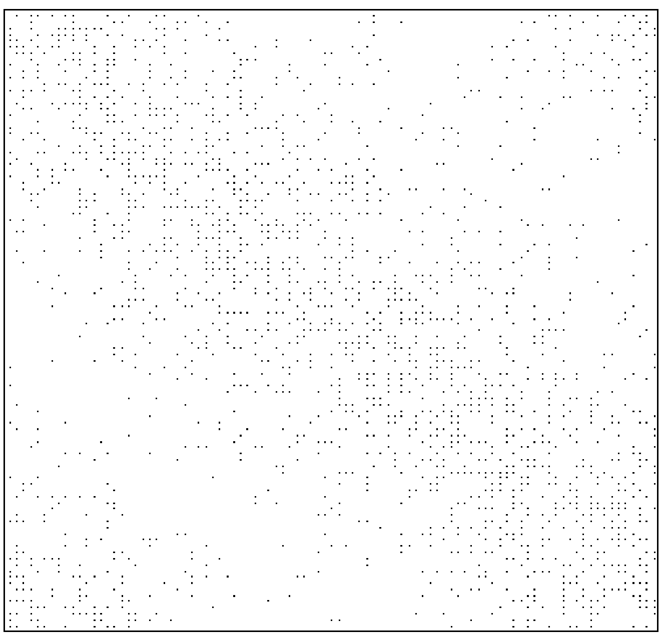

Figure 6: The connectivity matrix that results with fitness function f_{RL} and a 100 unit network.

In earlier work [4] 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.

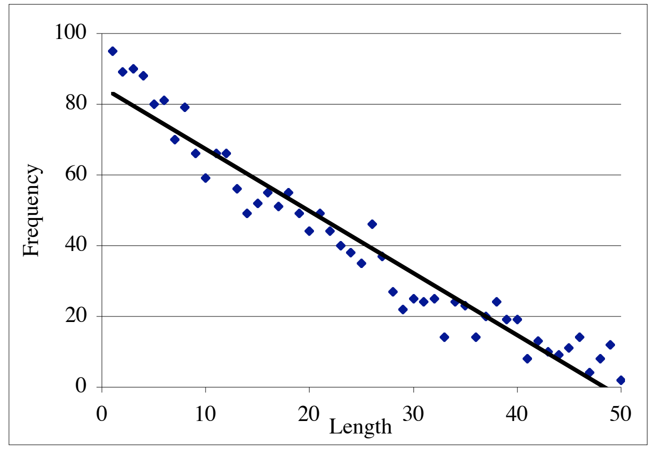

Figure 7: A histogram of the final connection lengths of a 100 unit network optimized for both low L and high R.

V1. DISCUSSION

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 [4, 12-14]. 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 matrix with respect to the task in hand. Of course, evolutionary computation has been applied extensively to finding architectures (and weights) for neural networks [15], but as far as we know no one has used these methods in the context proposed here.

Even with the small associative memories used here, the search space in which the GA operates is vast: 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. The results show, however, that it is possible to find connectivity that not only minimizes L, but also one that does this whilst maintaining 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 apparently decreasing linearly with distance. It will be interesting to explore how well this result corresponds with real neuronal systems. Whilst the number of incoming connections of a node in the network is fixed, the number of efferent, outgoing, connections is not. It will also be instructive to investigate whether there is a pattern to this evolved efferent connectivity. Perhaps some units are acting like hubs as in certain *scale-free* networks [16].

It will also be interesting to allow the network complete freedom in connectivity, with no constraint on the number of incoming connections that a node may have.

Although this is early work, it is already apparent that it is possible to evolve non-trivial connection strategies, to produce efficient associative memories. Much further analysis and experimentation is needed to properly understand the intriguing relationship between functionality and connectivity in these models.

References

1. Sporns, O., et al., *Organization, development and function of complex brain networks.* Trends in Cognitive Sciences, 2004. **8**(9): p. 418- 425.
2. Cherniak, C., *Component placement optimization in the brain.* J. Neurosci., 1994. **14**(4): p. 2418-2427.
3. Brucoli, M., L. Carnimeo, and G. Grassi. *Discrete-time cellular neural networks for associative memories: a new design method via iterative learning and forgetting algorithms*. in *38th Midwest Symposium on Circuits and Systems*. 1995.
4. Davey, N., B. Christianson, and R. Adams. *High Capacity Associative Memories and Small World Networks*. in *IJCNN*. 2004. Budapest.
5. Noest, A.J., *Domains in Neural Networks with Restricted-Range Interactions.* Physical Review Letters, 1989. **63**(16).
6. Hopfield, J.J., *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, 1982. **79**: p. 2554-2558.
7. Diederich, S. and M. Opper, *Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules.* Physical Review Letters, 1987. **58**(9): p. 949-952.
8. Davey, N. and R. Adams, *High Capacity Associative Memories and Connection Constraints.* Connection Science, 2004. **16**(1): p. 47-66.
9. Watts, D. and S. Strogatz, *Collective Dynamics of 'small-world' networks.* Nature, 1998. **393**: p. 440-442.
10. Hertz, J., A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*. 1991, Redwood City, CA: Addison-Wesley Publishing Company. 327.
11. Davey, N., S.P. Hunt, and R.G. Adams, *High capacity recurrent associative memories.* Neurocomputing, 2004. **62**: p. 459-491.
12. Torres, J.J., et al., *Influence of topology on the performance of a neural network.* Neurocomputing, 2004. **58-60**: p. 229-234.
13. Stauffer, D., et al., *Efficient Hopfield pattern recognition on a scale-free neural network.* European Physical Journal B, 2003. **32**(3): p. 395-399.
14. McGraw, P. and M. Menzinger, *Topology and computationlal performance of attractor neural networks.* Physical Review E, 2003. **68**: p. 047102.
15. Yao, X., *Evolving Artificial Neural Networks.* Proceedings of the IEEE, 1999.
16. Newman, M.E.J., *Models of the Small World.* Journal of Statistical Physics, 2000. **101**(3/4): p. 819-841.