Structured Connectivity in an Associative Memory Model

Simon Turvey, Stephen Hunt, Ray Frank, Rod Adams, *Member IEEE,* Neil Davey

Abstract—**High capacity associative memory models with dilute structured connectivity are trained using naturalistic bitmap patterns. The connectivity of the model is chosen to reflect the local spatial continuity of the data. The results show that the structured connectivity gives the networks a higher effective capacity than equivalent randomly diluted networks. Moreover the locally connected networks have a much lower mean connection length than the randomly connected network. It is shown that a small amount of additional connectivity can correct any neurons that fail to train.**

Index Terms—**Associative Memory, Neural Network, Dilution, Capacity.**

I. Introduction

High capacity associative memory models can be constructed from networks of perceptrons, trained using the normal perceptron training procedure. Such networks have a capacity much higher than that of the standard Hopfield network, and in fact their capacity is related to the capacity of a single perceptron. A perceptron with N inputs can learn 2N random unbiased (not correlated) patterns, giving a capacity 2, but this capacity is increased beyond 2 if the training set is correlated [1]. This implies that a Hopfield network of N units, when trained using Perceptron learning, will have also have a capacity of 2. These improvements in capacity are matched by improvements in the performance of the networks as associative memories: the attractor basin size of trained patterns is increased.

In this paper we are interested in networks with diluted connectivity, where an individual perceptron is connected to only a fraction of the other perceptrons in the network. Diluting these networks on a random basis causes the capacity to fall in a roughly linear way with the fraction of connections removed [2].

In diluted networks a perceptron only sees a particular fragment of the training set, namely that part that comes from its connected units. We are interested in whether characteristics of certain types of training data can be exploited by diluting the network connectivity in a definite, structured way. In particular we investigate whether networks with a specific pattern of reduced connectivity can give enhanced performance with naturalistic, bitmap training patterns with inherent spatial continuity.

Sections II, III and IV describe the Network Dynamics, Network Topology and Training Procedure respectively. Sections V and VI discuss the effect that correlation in the data may have on capacity and the correlations that are present in our data. Sections VII and VIII give the results and Section IX concludes.

Simon Turvey is a Visiting Research Fellow at the University of Hertfordshire, Hatfield, UK., (email S.P.Turvey@herts.ac.uk)

Stephen Hunt, Ray Frank, Rod Adams and Neil Davey are Lecturers at the University of Hertfordshire, Hatfield, UK., (email {S.P.Hunt, R.J.Frank, R.G.Adams, N.Davey}@herts.ac.uk

II. Network Dynamics

All the high capacity models studied here are modifications to the standard Hopfield network. The net input, or *local field*, of a unit, is given by: $h_i = \sum_{j \neq i} w_{ij} S_j$ where S 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 Θ is the heaviside function. Unit states may be updated synchronously or asynchronously. Here we use asynchronous, random order updates. A symmetric weight matrix and asynchronous updates ensures that the network will evolve to a fixed point. If a training pattern ξ^{μ} is one of these fixed points then it is successfully stored, and 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.

III. Network Topology

Associative memory models based on the Hopfield architecture are usually fully connected, so that any spatial relationship between the units in the network is irrelevant. Here, however, we arrange the units in the network into a two dimensional grid, as in a two dimensional SOM [3]. The reason for this choice is that this structure matches the correlation in the two dimensional data sets used (see Section VI). This introduces a topology on the units in the network that can be used to define a distance between any two units in the network. We use square neighbourhoods (as is normally the case in a SOM), so that the 8-units in the immediate square around a unit are defined to be at unit distance from that unit, as shown in Figure 1. We say that the network has structured connectivity with d = 1 if every unit is connected to every other unit at distance 1 and no others, and has structured connectivity with d = 2 if every unit is connected to every other unit at distance of not more than 2, and so on. Note that this is a symmetric connection strategy. Wraparound on the grid is not used, so that the edge units have fewer connections than the inner units. For comparison purposes we also use networks with random diluted connectivity, so that a random proportion of connections are removed prior to training.

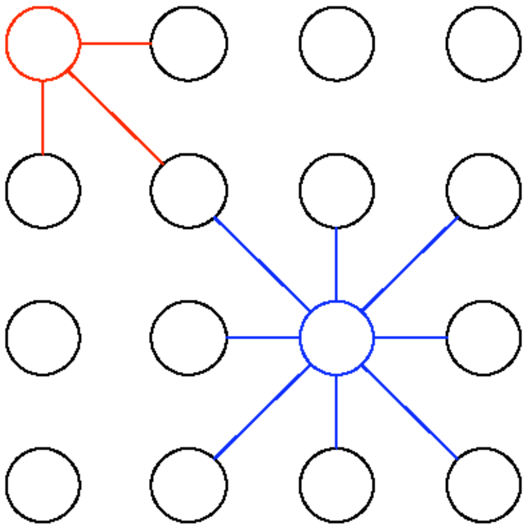

Figure 1. A small network in which neighbourhood connectivity has been established at a distance (d) of 1. Connections are shown for two neurons as an example.

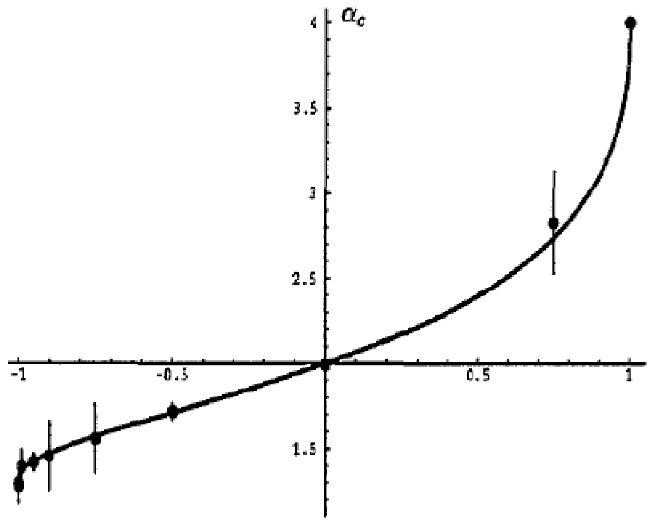

Figure 2. The capacity of a perceptron as the pair wise normalized overlap of training patterns with the same output, is varied (x-axis). With zero pair wise overlap the normal capacity of 2 is shown, but as overlap increases so does the capacity, approaching a limiting value of 4. Taken from Lopez et al [5].

IV. Training

The networks are trained using a modification of the normal perceptron training rule that ensures symmetric weights [1]. The 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 $h_i^p \xi_i^p$. If this is less than T
    then change the weights to unit *i*
    according to:

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

Where ξ^p denotes the training patterns, and T is the learning threshold which here has the value of 0. All weights on removed connections are fixed at zero throughout.

V. Capacity Results for Perceptron Networks

A perceptron with N inputs can learn up to 2N random patterns, and as the correlation in the training set increases so does the capacity of the perceptron. Imposing symmetry on the weights, in a network of perceptrons, does not affect this maximum capacity [4]: even with uncorrelated training sets capacity may be greater than 2N. This occurs when correlated subsets of the training set have correlated outputs [5]. So, for example, if pairs of the training set are correlated and have the same output then the training set is more likely to be learnable. Put simply, if similar patterns have the same label then a perceptron is more likely to be able to learn the classification. The increasing capacity is shown in Figure 2. As the normalized pair wise overlap of training patterns with the same output increases to its maximum of one then the capacity approaches four.

VI. Training Sets Used

Two sets of training patterns, representing reasonably naturalistic images were created. All the generated patterns were 400 bits, 20 by 20 bitmap images, with black as –1 and white as +1. The geometric data uses solid geometric shapes placed at random within the 2-dimensional grid. Each image has four random shapes taken from: triangles, squares or circles. Shapes may overlap but are clipped if they overrun an edge. The character data consists of alpha-numeric bitmaps. Examples from these data sets are shown in Figure 3.

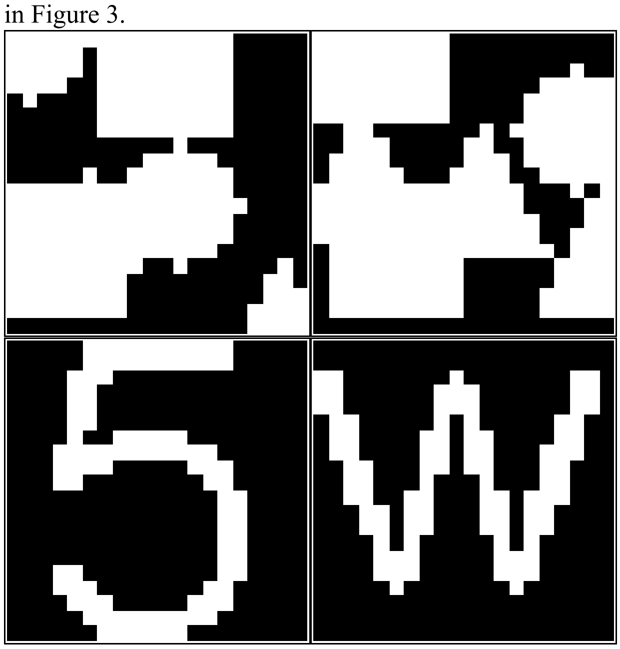

Figure 3. Example bitmaps from the geometric (above) and character training sets.

The geometric data set is roughly unbiased (bias, the proportion of +1's, is 0.52), whereas the character data has a bias of 0.2, since the image is mainly the black background (-1). Both sets have the desired characteristic of within pattern spatial continuity. This can be seen in the mean local correlation of the images, for different neighbourhood sizes, see Tables 1 and 2. For both data sets the correlation of individual bits with their neighbours decreases as that neighbourhood is increased.

TABLE I.
MEAN LOCAL CORRELATION FOR THE GEOMETRIC IMAGES

Neighbourhood Size, d	Mean Local Correlation
1	0.89
2	0.83
3	0.77
4	0.72
5	0.68
Full Grid	0.5161

TABLE II.
MEAN LOCAL CORRELATION FOR THE CHARACTER IMAGES

Neighbourhood Size, d	Mean Local Correlation
1	0.87
2	0.78
3	0.74
4	0.71
5	0.70
Full Grid	0.68

VII. FAILED NEURON RESULTS

The networks used here are highly diluted, for example in networks with structured connectivity at d = 1 (units connected to those in an immediate square neighbourhood only) each unit is connected to no more than 8 other units, and corner units are connected to only 3 other units. So with any training set it is very likely that some units will fail to train. We therefore report the number of units that fail to train at a given loading and expect this figure to be lower for networks with structured connectivity than for those with the same level of random connectivity. The network is trained for 1000 epochs, well beyond the number of epochs normally required for convergence at the kind of loadings we use here. The number of units that have failed to converge at this point is counted.

A. Geometric Data

Figure 4 shows how the number of neurons that fail to train increases with the loading on the network. For comparison the results for networks with equivalent levels of random connectivity are also shown in Figure 5. The randomly connected networks show the expected pattern. The capacity of such networks should be about $2n$ where n is the number of inputs for *each* perceptron. So that for the random network with a mean connection per neuron of just under 8 (equivalent to the d = 1 structured network) most units should fail with about 16 patterns – a loading of 16/400 or 0.04. However the structured network shows a very different pattern at this level of connectivity with a roughly linear increase in failed neurons as the loading increases, but no sudden jump in the failure rate. Remarkably the d = 3 network (each unit having roughly 40 inputs) has a very low failure rate throughout the loading range – up to 100 patterns. The equivalent randomly connected network has more than half the units failing to train with 75 patterns in the training set (loading = 0.1875).

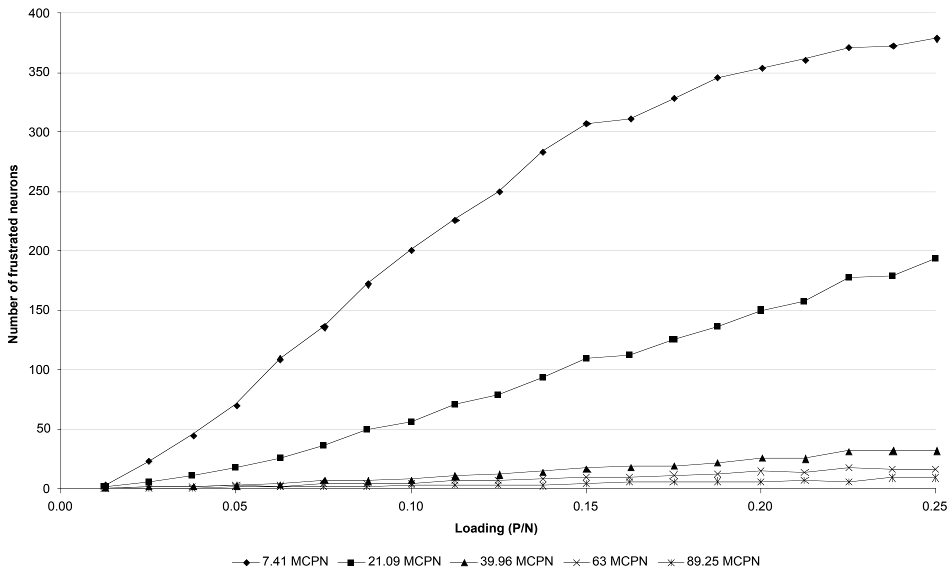

Figure 4. Failed neuron count against increasing pattern load for networks constructed with structured connectivity at levels of 7.41, 21.09, 39.96, 63, and 89.25 mean connections per neuron (corresponding to d = 1, d = 2 etc) and trained using geometric data. Mean values over 5 runs at each loading are given.

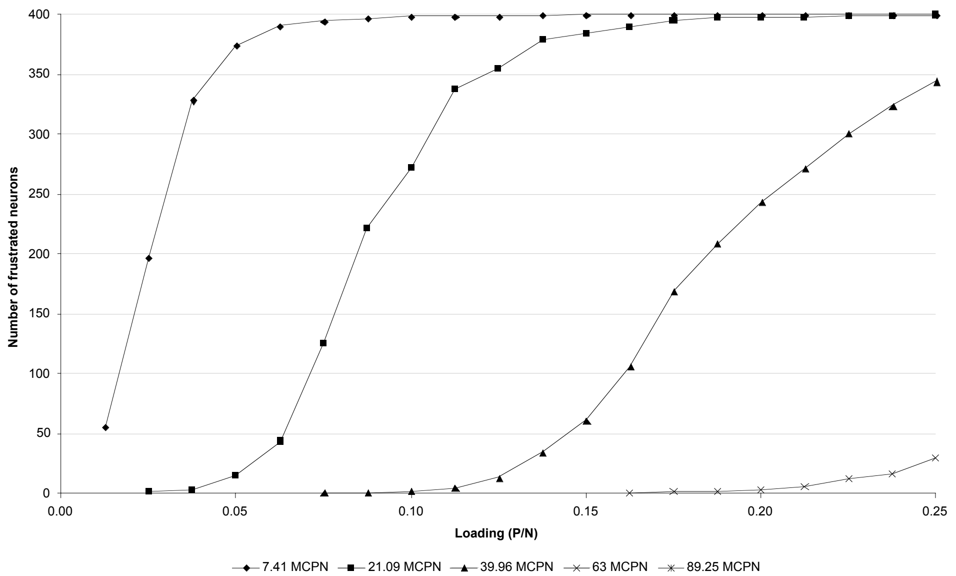

Figure 5. Failed neuron count against increasing pattern load for networks constructed with random connectivity trained with using geometric data. The level of mean connections per neuron was the same as that for the structured connectivity. Mean values over 5 runs.

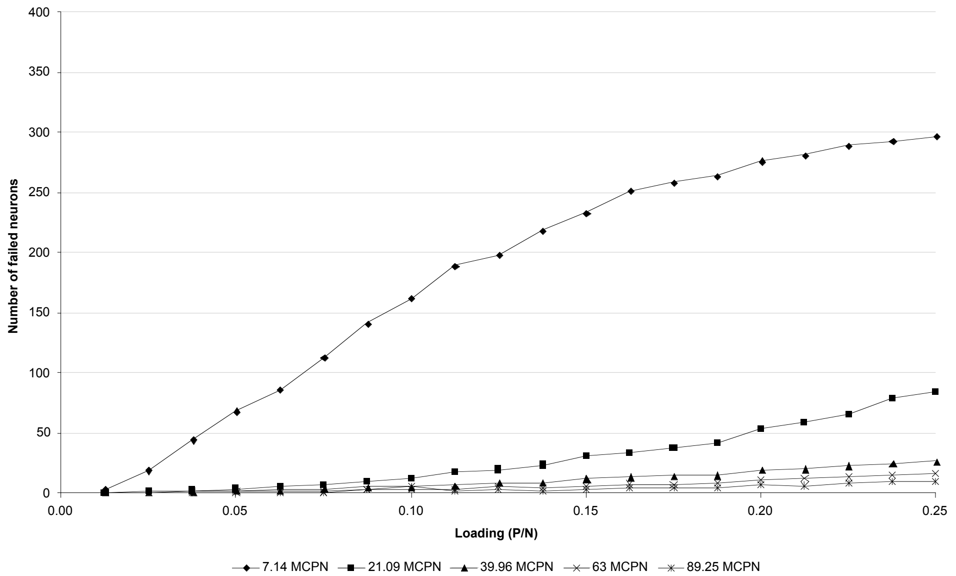

Figure 6. Failed neuron count against increasing pattern load for networks constructed with structured connectivity trained using character data. Mean values over 5 runs.

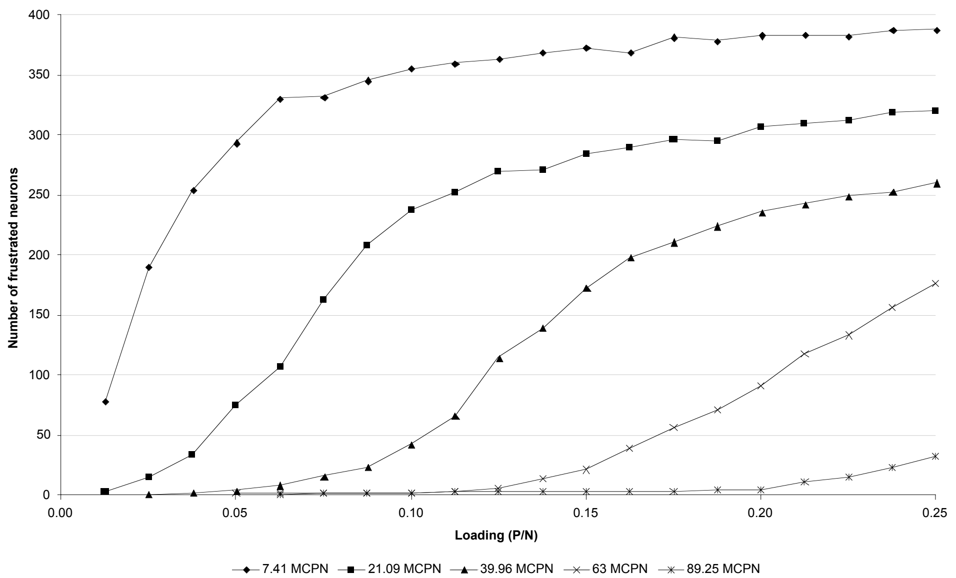

Figure 7. Failed neuron count against increasing pattern load for networks constructed with random connectivity trained using character data. Mean values over 5 runs.

B. Character Data

The character data is biased and so the capacity of an individual perceptron should here be higher than for the unbiased geometric data. However the dramatic benefit of structured local connectivity is even more apparent here, see Figures 6 and 7. Once again the d = 3 network shows very low failure rate across all loadings and even the d = 2 network has less than 25% failures at the top loading of 0.25 (100 patterns).

VI11 Dealing with the Failed Neurons

Of course a network in which a number, albeit small, of neurons fail to learn the training data is not satisfactory. To deal with these units we simply add additional, random, connectivity. Since the probabilty that a perceptron will fail to learn decreases as the number of inputs increases (assuming fixed loading), we can deal with these units by giving them additional connections. The specific method we employ is to train the network with normal local connectivity. For each unit that has failed to learn its training set, an additional, symmetric connection is added between it and a randomly chosen target. Any unit with changed connectivity is now retrained. The process is repeated until all units have successfully learnt their training set.

Figure 8 shows the resulting level of connectivity at different loadings and for different neighbourhoods for the geometric data. It is apparent that only the d = 1 networks required significantly more connections to learn the training sets as the loading increased. In fact the connectivity required to learn 100 patterns (loading of 0.25) in the d = 1 network was at least twice the original level, so that more than half the connections had been added after initial training. The actual number of additional connections needed to learn at a loading of 0.25 is shown in Table 3.

Table III.
The mean added connectivity per neuron needed to learn 100 patterns (loading 0.25) for 5 different neighbourhoods. Averages over 5 runs

d	Added Connectivity
1	13.535
2	4.078
3	0.663
4	0.422
5	0.22

Figure 9 gives the storage efficiency, the ratio of patterns stored to the mean connections per neuron, for the various networks, at different loadings, once again using the geometric data. It can be seen that the d = 1 network is consistently the most efficient of the networks at all loadings. Efficiency decreases as the neighbourhood increases. Results for the Character data show a very similar pattern.

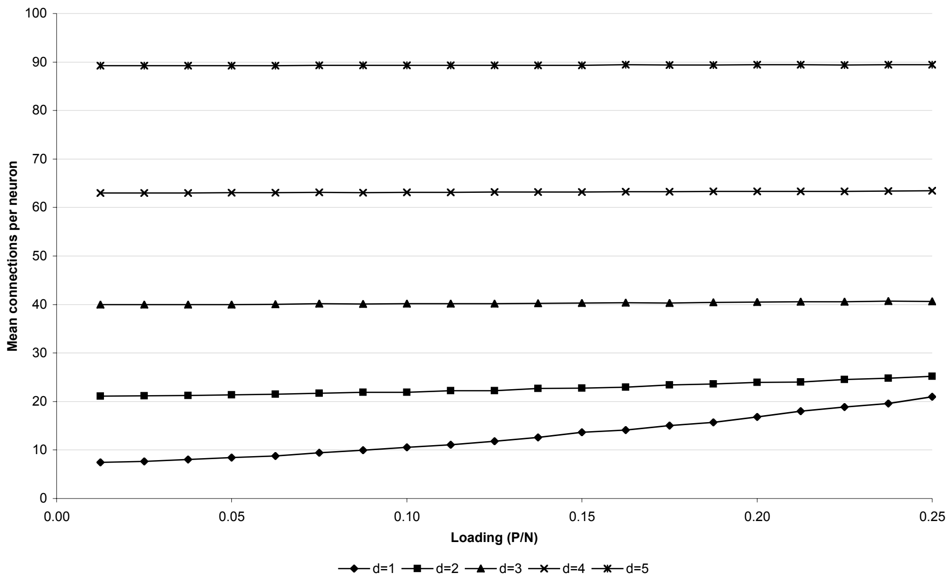

Figure 8. Mean connections per neuron is plotted on the vertical axis as the loading increases on the horizontal axis. Five different neighbourhoods are shown, from d = 1, the lowest line, up to d = 5.

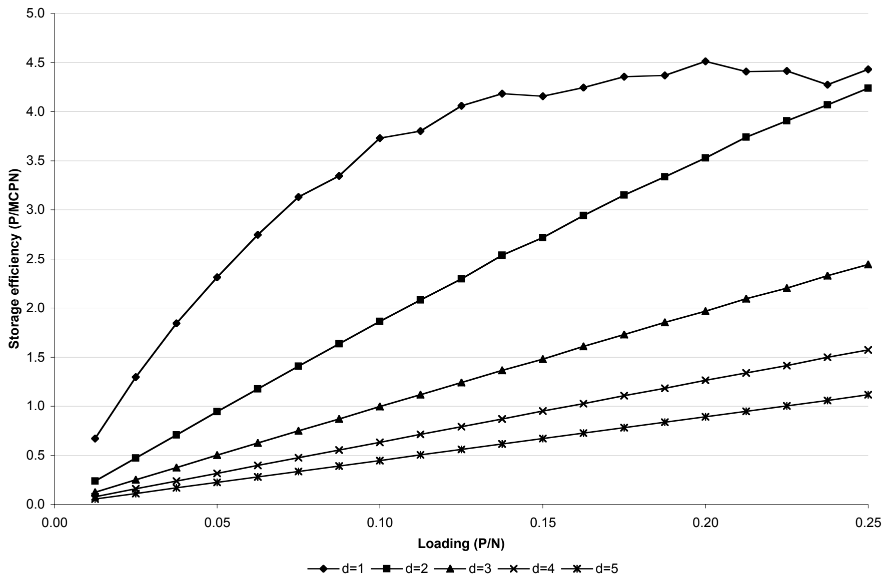

Figure 9. The storage efficiency of the 5 different neighbourhoods at varying loadings. The vertical axis is the ratio of patterns stored to the mean connections per neuron. The horizontal axis shows increasing loading.

IX. Discussion

Much natural data shows spatial and/or temporal continuity and this aspect of the data could be exploited by an engineered or evolved system – artefactual or natural. Here we have shown that a simple associative memory model, a network of perceptrons, can exploit the local correlation present in simple bitmap images. The effective capacity (tolerating a small number of failed units) of the networks with structured connectivity is much better than those with an equivalent number of random connections.

A significant further benefit of the locally connected networks should also be noted. The mean connection length is obviously much lower in these networks. For example the d = 1 network has mean connection length of 1, whereas the randomly connected network in a 20 by 20 grid has a mean connection length of about 9.3. This has significance for any physical instantiation of these networks.

For the small number of units that fail to train a small amount of additional connectivity is shown to correct the problem. In fact only the d = 1 networks required a significant amount of additional connectivity.

A more important problem with the idea of locally structured connectivity is that the pattern correction/completion behaviour of the network can be adversely affected. The recall process may get stuck in patterns with large subdomains of errors [7]. The subdomain may not have enough distal input to overcome its locally stable configuration. This issue may be addressed by introducing further random connectivity and the results of doing this are promising [6, 8].

In summary this paper has described how structured local connectivity can increase the effective capacity of an associative memory when dealing with spatially continuous data and can produce large savings in the length of connections required.

References

[1] Gardner, E., *The space of interactions in neural network models.* Journal of Physics A: Mathematical and General, vol. 21, no. 1, pp. 257-270, 1988

[2] Davey, N., R. Adams, and S.P. Hunt. *High Performance Associative Memory and Weight Dilution*. in *Proceedings of the 8th International Conference on Neural Information Processing (ICONIP 2001)*. Shanghai, China, 2001

[3] Hertz, J., A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Redwood City, CA, USA: Addison-Wesley Publishing Company. 1991,

[4] Nardulli, G. and G. Pasquariello, *Domains of attraction of neural networks at finite temperature*. Journal of Physics A: Mathematical and General, vol. 24, no. 5, pp. 1103-1117, 1991

[5] Lopez, B., M. Schroder, and M. Opper, *Storage of correlated patterns in a perceptron.* Journal of Physics A: Mathematical and General, vol. 28, pp. L447-L452, 1995

[6] Turvey, S.P., *Analysing and Enhancing the Performance of Associative Memory Architectures*, in *Computer Science*, PhD Thesis, University of Hertfordshire: Hatfield, UK, 2003.

[7] Noest, A.J., *Domains in Neural Networks with Restricted-Range Interactions.* Physical Review Letters, vol. 63, no. 16, pp. 1739-1742, 1989.

[8] Bohland, J. and A. Minai *Efficient Associative Memory Using Small-World Architecture.* Neurocomputing, vols. 38-40, pp. 489-496, 2001