The Capacity and Attractor Basins of Associative Memory Models

N.Davey & S.P.Hunt

N.Davey@herts.ac.uk, S.P.Hunt@herts.ac.uk
Department of Computer Science
University of Hertfordshire
Hatfield, Herts., UK. AL10 9AB
tel (01707) 284321

Abstract. The performance characteristics of five variants of the Hopfield network are examined. Two performance metrics are used: memory capacity, and a measure of the size of basins of attraction. We find that the post-training adjustment of processor thresholds has, at best, little or no effect on performance, and at worst a significant negative effect. The adoption of a local learning rule can, however, give rise to significant performance gains.

1 Introduction

In this paper we investigate the performance of several variations of the basic Hopfield model of associative memory. The standard model is well studied and is known to have a capacity of roughly 0.14 of the number of units in the network, for maximally decorellated random patterns. The modified models studied here claim higher performance.

2 The Models Examined

In this section we take a set of P, N-ary, bipolar (+1/-1) training vectors, $\{\xi^p\}$. The N by N weight matrix is denoted by $\mathbf{W}$, and the state of the i'th unit is denoted by S_i

2.1 The Standard Model

In this model the net input, or local field, of a unit, is given by: $h_i = \sum_{j \neq i} w_{ij} S_j$

The next state of the unit is then given by: $S_i(t+1) = \begin{cases} 1 & \text{if } h_i > \theta_\mathrm{i} \\ -1 & \text{if } h_i < \theta_\mathrm{i} \\ S_i(t) & \text{if } h_i = \theta_\mathrm{i} \end{cases}$

where the threshold, θ_I, is normally taken as zero. The update can be synchronous or asynchronous. Here we use asynchronous, random order updates. The weight matrix is calculated using one-shot Hebbian learning:

$$w_{ij} = \frac{1}{N} \sum_{k=1}^{P} \xi_i^p \xi_j^p$$

2.2 The Local Learning Model

In this model we employ a local learning rule proposed by Diederich and Opper [1987]. The Hebbian learning rule is modified to create a simple iterative scheme that will converge to a weight matrix in which all the training patterns are guaranteed to be stable. This is the only model considered here in which the learning is iterative and may take many epochs.

For training patterns to be stable a sufficient condition is that:

$$\mathbf{W}\xi^{p} = \xi^{p} \text{ for all } p$$

(assuming that all $\theta_i = 0$), or equivalently, for $\mathbf{W}\Xi = \Xi$, where Ξ is the matrix whose columns are the ξ^p.

One solution to this is given by the projection learning rule [Personnaz et al, 1986] which gives $\mathbf{W} = \Xi\,\Xi^{I}$, where $\Xi\,\Xi^{I}$ is the pseudo-inverse of Ξ. Unfortunately this learning rule is non-local [Storkey, 1997], and therefore lacks biological plausibility.

A learning rule is local if the update of a particular connection depends only on information available to the processing elements on either side of the connection. Fortunately there is an iterative, and local, learning rule which converges to the appropriate weight matrix. This is given by:

```
1.  Begin with a zero weight matrix
2.  Repeat until all patterns are stable
    2.1 Set the initial state of network to one of the $\xi^p$
    2.2 For each unit, $i$, in turn
        2.2.1  Calculate the net input to unit $i$, and hence its next state.
        2.2.2  If $i$ wishes to change state, update its incoming weights according to:
```

$$\Delta w_{ij} = \frac{S_i^p S_j^p}{N-1}$$

This process will converge on a suitable weight matrix if such a matrix exists [Diederich and Opper, 1987], at which point the trained patterns are guaranteed to be stable. The maximum useful capacity of a network trained in this way is $N - 1$ linearly independent patterns.

2.3 Storkey Learning

Storkey [1997] has proposed an incremental local learning rule that requires only one pass through the training set, and gives a weight matrix that is a first order approximation to the pseudo-inverse. Given an initially zero weight matrix, the addition of a new training pattern, ξ^p, changes weights according to the prescription:

$$\Delta w_{ij} = \frac{1}{N}\left(\xi_i^p \xi_j^p - \xi_i^p h_j - \xi_j^p h_i\right)$$

2.4 Adjustable Threshold Network

In the normal Hopfield network the thresholds are all set to zero. In the Adjustable Threshold Network [Schultz, 1995] an attempt is made to use the thresholds to maximise the network performance once a weight matrix is in place.

The motivation for this method can be seen from a consideration of a unit, i, in a trained network, whose local field for each of four training vectors is: -3, -1, 5 and 7. If i has a zero threshold, its state will be –1 for the first two patterns and +1 for the latter two. Schultz proposes maximising the "slack" over the training set, by moving the threshold for each unit so that it gives maximum separation between the positive and negative local fields; in this case that would mean putting the threshold half way between –1 and +5, giving $\theta_i = +2$. In general then, the threshold for a unit, i, is set half way between the positive and negative fields with the smallest magnitudes, as given by:

$$\theta_i = \frac{h_i^+ + h_i^-}{2}, \quad \text{where} \quad \begin{array}{l} h_i^+ = \max\left\{h_i^p \mid h_i^p \geq 0\right\} \\ h_i^- = \min\left\{h_i^p \mid h_i^p < 0\right\} \end{array}$$

2.5 Adjustable Threshold Local Learning Network

If threshold adjustments are performed in the manner described in section 2.4, it can be seen that those training patterns that were stable before adjustment should remain stable after adjustment. Consequently, variable thresholds should be usable with any learning rule.

The weight matrix for an adjustable threshold local learning network is determined in the manner set out in section 2.2, and the thresholds are then adjusted as in 2.4. Adjusting thresholds in this way should not change the stability of the patterns learnt under local learning, so the maximum capacity of this model should also be N-1 linearly independent patterns.

3 Measurement of Performance

Five associative memory models were investigated: the standard Hopfield network, the Storkey network, the local learning network, the adjustable threshold network, and finally the adjustable threshold local learning network. Each model was investigated first with sets of unbiased random training patterns, in which ($prob(\xi_i^p = +1) = 0.5$), and then again with biased data in which the bits in each pattern had a 0.3 probability of being +1.

3.1 Network Capacity

For an associative memory to be useful, the stored patterns should be stable in the configuration space: if the network state is a stored pattern, the state should not change. As the size of the training set increases, the probability that all the stored patterns are stable decreases. Moreover, if the stored patterns are correlated their stability becomes less likely. As is well known [Hertz, Krogh & Palmer, 1991], the maximum capacity of a standard Hopfield network is roughly 0.14*N* random patterns.

In these experiments, network capacity was determined by measuring the proportion of trained patterns that were stable under different memory loads. Each network under investigation was trained with 50 different sets of *P* unbiased random patterns, for a range of different values of *P*. The proportion of training patterns which were stable was measured after each training run, and the results were averaged over the 50 runs for each value of *P*. This procedure was then repeated with sets of biased random patterns, as described above.

3.2 Basins of Attraction

For useful pattern association to occur, the patterns stored in an associative memory of this kind must act as attractors. Ideally they will be the only attractors and will act parsimoniously, so that a given initial state will relax to the nearest trained memory. For very small networks it is possible to exhaustively explore the state space (see, for example [Personaz et al, 1986]), but for more realistic sizes the nature of the attractors can only be explored statistically.

The radius of the basin of attraction of a pattern is defined as the largest Hamming distance within which *almost all* states flow to the pattern[1]. The average of the radii of attraction of each of the stored patterns gives a measure of the pattern completion capability for a given trained network. In these experiments we have calculated a value, *R*, which provides a normalised measure of the average radius of basins of attraction, following an approach similar to that of Kanter and Sompolinsky [1987].

[1] It is frequently impractical to test *all* starting states within a given radius of a stored pattern. An alternative approach is to test a sizeable sample of those states (50 in this paper). If every starting state in the sample flows to the stored pattern, then we infer that *almost all* states will do so.

A series of initial states is chosen. In each case, a fixed fraction, *m*, of the state is identical to the corresponding part of one of the stored patterns, ξ^{p}, and the rest of the state is random. If the value of *m* is high then the network will relax to $\xi.^{p}$ for every one of those initial states. The value of *m* is reduced until a value, m_0, is reached at which one or more initial states do not relax to the desired state. Averaging m_0 over different stored patterns yields

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

As is pointed out in [Kanter and Sompolinsky, 1987], the initial states used in this calculation may overlap one of the other stored patterns more closely than ξ^{p} and to compensate for this, the definition of *R* is modified to:

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

where m_1 is the largest overlap with the rest of the stored patterns.

In our implementation, the search for the value m_0 is undertaken from low *m* to high *m*. 50 random starting points are chosen, each of which has low overlap with every member of the training set (low average *m*). If, as is likely, the start state does not relax to the closest training pattern in any of the 50 cases, the value of *m* is increased (by 0.01), and the search is repeated. This continues until all of the 50 random start states relax to the closest stored pattern. This procedure is performed for six different sets of stored patterns for each network type: three sets of unbiased random patterns, and three sets of biased random patterns.

The perfect attractor network has *R* = 1, which means that it is possible to move away from any stored pattern, and stay within its basin of attraction up to the point at which another stored pattern becomes nearer (see Figure 1). Note that the calculation of average attractor basin size for the trained patterns can only be undertaken when these patterns are themselves stable.

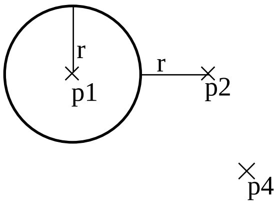

Figure 1. The closest pattern in the training set to p1 is p2, at a distance of 2r. Optimal performance occurs when all the vectors within the hypersphere centred on p1 and radius r, are attracted to p1. If all patterns stored in a network exhibit this performance, its normalised average basin of attraction,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

4 Results

4.1 Capacity

The standard Hopfield net was compared to the Adjustable Threshold model and the Storkey model, with 100 unit nets and results averaged over 50 runs. The results can be seen in Figures 2 and 3. Note that the local learning network with and without adjustable thresholds is able to store up to N-1 patterns, and is therefore not added to the Figures.

Perhaps surprisingly, the capacity of the adjustable threshold network is lower than the standard model. Moreover, the dynamics of the adjustable threshold modification do not guarantee single point attractors and, as can be seen, the network breaks down particularly badly with the biased random patterns, where it only manages to store 3 patterns before breakdown. However, the Storkey network shows a significantly improved performance over the standard model, both with unbiased and biased data.

4.2 Attractor Basins

The average size of the attractor basins of a network can only be measured for associative memories in which almost all the trained patterns are stable, so it is only possible to report R for the standard Hopfield model at low capacity, where it is near to unity. The same is true for the simple adjustable threshold model. However, for the pseudo-inverse based models it is possible to examine R at higher loading. We have therefore concentrated on these models.

The results for the local learning model, the adjustable threshold local learning model, and the Storkey model, all with 100 units, are shown in Figures 4 and 5. The figures reported are means over three runs for each pattern increment. The attractor test was run fifty times for each network, at each increment, with differing random starts.

Considering the two local learning networks first, we can see that they perform well, even at high loading – for example with 30 stored patterns (more than twice the maximum capacity of a standard Hopfield network of this size), R 0.6, for both types of network and both types of training pattern. In comparison the Storkey model performs less well, particularly on the biased data.

4.3 Training Times

As already mentioned the full local learning rule is incremental, unlike all the other rules considered here which use one-shot learning. Figure 6 shows the number of epochs required to reach a weight matrix which gives stability for all patterns in the training set. Results are averaged over two runs at each pattern increment.

Whilst it is not possible to train even a very lightly loaded network in a single pass through the training set, it is clear from these results that the local learning model has short training times when compared with iterative learning rules based upon error minimisation.

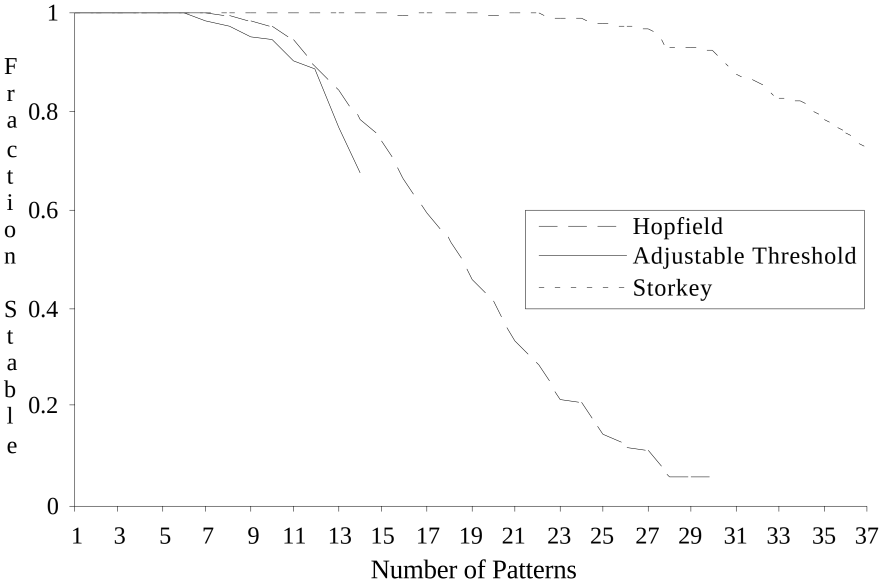

Figure 2 Capacity of 100 unit networks trained on unbiased ($prob(\xi_i^p = +1) = 0.5$) random patterns. The results are averaged over 50 runs.

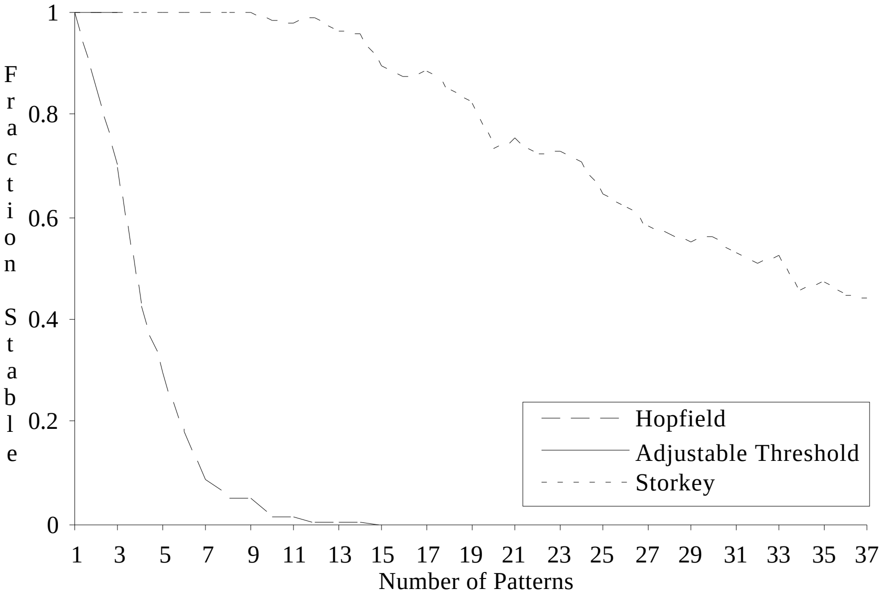

Figure 3 Capacity of 100 unit networks trained on biased ($prob(\xi_i^p = +1) = 0.3$) random patterns. The results are averaged over 50 runs.

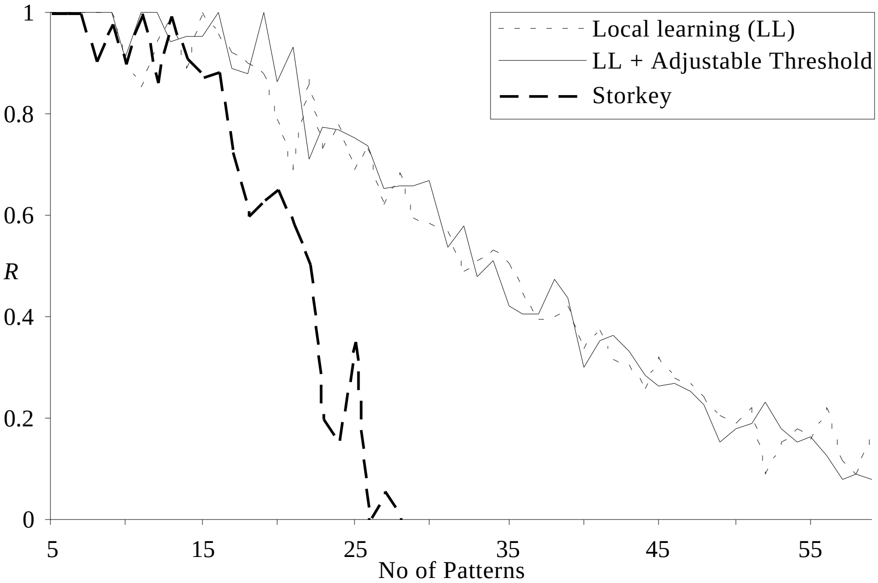

Figure 4. 100 unit networks and random $prob(\xi_i^p = +1) = 0.5$ patterns. Results are the mean of 50 runs. The vertical axis shows the normalised average basin of attraction, *R*

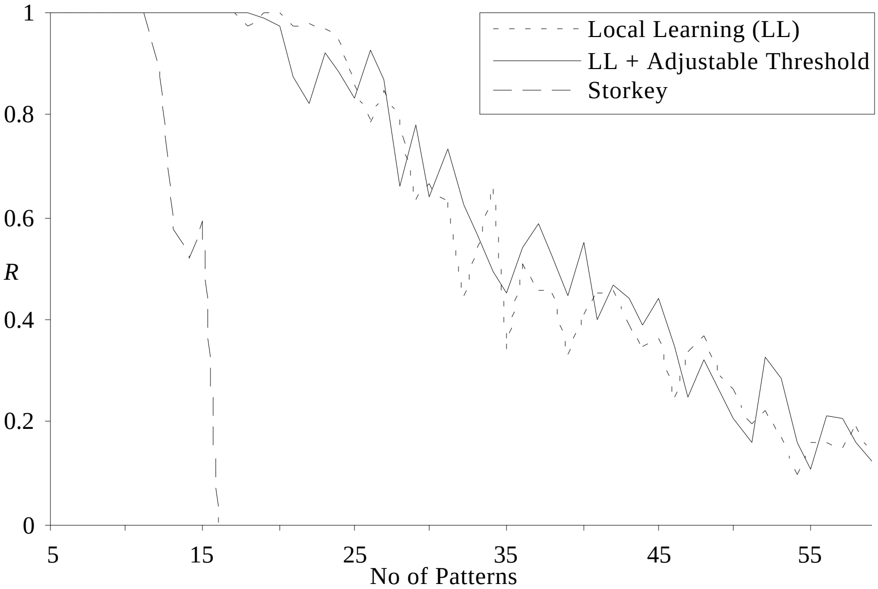

Figure 5. 100 unit networks and random $prob(\xi_i^p = +1) = 0.3$ patterns. Results are the mean of 50 runs. The vertical axis shows the normalised average basin of attraction, *R*

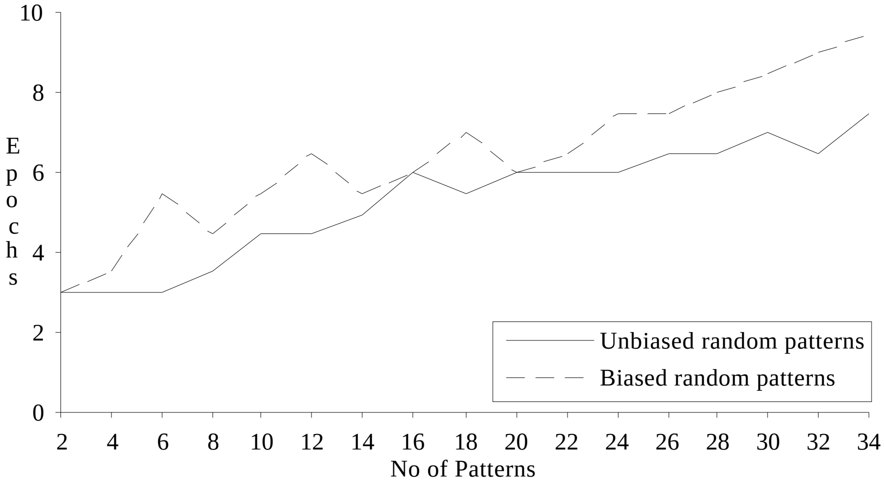

Figure 6 Training times for 100 unit local learning networks. The results are averaged over 2 runs. The vertical axis shows the average number of epochs required to train the network.

5 Discussion

5.1 Capacity

As expected, the Hopfield network performs poorly in these tests. The Storkey model gives a very significant improvement in storage capacity over the standard Hopfield network, for a very modest increase in training time. This takes the form of an improvement both in the maximum capacity at which the proportion of stable patterns is 1, and in the rate at which this fraction decreases with increasing memory load. What is particularly noteworthy is that the Storkey model performs better even on partially correlated pattern sets (i.e. the biased data) than does the Hopfield network on unbiased random patterns.

The post-training adjustment of processor thresholds has a deleterious effect on the capacity of the standard Hopfield model. In fact it appears to lead to the emergence of cyclic attractors at even modest memory loads. It is interesting to note, however, that the application of threshold adjustments to networks trained using the local learning rule has no such deleterious effect.

Both variants of the local learning model perform perfectly at loadings up to 80 or more patterns, as would be expected. Whilst this is at the cost of some increase in training time, that increase is small given the massive improvement in the utility of the model.

5.2 Basins of Attraction

Again the standard Hopfield network performs poorly, as does the simple adjustable threshold network. In fact, neither of them performs well enough to be included in these tests. The Storkey model puts in a creditable performance, with mean R greater than 0.85 for training sets of up to 16 unbiased patterns, and up to 12 biased patterns. However, its performance under heavier loads falls away very rapidly, reaching $R = 0$ for training sets containing as few as 26 unbiased and 16 biased patterns.

The local learning model continues to achieve perfect ($R = 1$) performance under increasing memory loads for longer than the Storkey model. In addition, R falls off relatively slowly, and approximately linearly, as memory loading is increased. Surprisingly, the performance on the biased (0.3) data is a little better than on the unbiased (0.5) data. This behaviour does not appear to be affected to any great extent by the adjustment of processor thresholds.

In conclusion, the best performance was achieved by the local learning model, which has very high capacity and robust attractor performance. The Storkey network is, by comparison, a poor performer, but it is still the best of the one-shot learning models we have tested by some considerable margin.

6 References

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

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

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

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

Schultz,A. (1995). Five Variations of Hopfield Associative Memory Network. *Journal of Artificial Neural Networks* **2**(3), 285-294

Storkey,A., and R. Valabregue (1997) Hopfield Learning Rule with High Capacity Storage of Time-Correlated Patterns. *Electronics Letters* **33**(21), 1803-1804