STOCHASTIC DYNAMICS AND HIGH CAPACITY ASSOCIATIVE MEMORIES

N.Davey, R.G.Adams

Department of Computer Science,
University of Hertfordshire,
College Lane, Hatfield, AL10 9AB. United Kingdom
N.Davey@herts.ac.uk, R.G.Adams@herts.ac.uk

ABSTRACT

The addition of noise to the deterministic Hopfield network, trained with one shot Hebbian learning, is known to bring benefits in the elimination of spurious attractors. This paper extends the analysis to learning rules that have a much higher capacity. The relative energy of desired and spurious attractors is reported and the affect of adding noise to the dynamics is empirically investigated. It is found that the addition of noise brings even more benefit in the case of the higher capacity rules.

Key-Words Associative memory, Attractor basins, Hopfield neural networks, Learning rules, Perceptron, Performance measures, Pseudo-inverse.

1 INTRODUCTION

In this paper we examine how a variety of high capacity associative memory models respond to noise in the dynamics. The networks are all variations on the standard Hopfield model, differing in the weight matrix that is used to embed the set of training patterns.

All models of this type function as associative memories. The learnt patterns act as attractors in the state space of the network, so that network states that are near to learnt patterns may move towards a learnt pattern, under the network dynamics. However the learnt patterns are not the only attractors, there may be many others: either correlated with mixtures of the training patterns, or otherwise. These spurious attractors are normally unhelpful.

In the standard stochastic model the addition of noise to the dynamics can be beneficial [1]: the free energy landscape is changed so that, spurious local minima of the energy function may no longer be stable; the probability that the network ends in a learnt pattern can be increased.

The work presented here empirically investigates whether similar benefits can arise in the high capacity versions. All the networks examined share the same dynamics, either deterministic or stochastic; they differ in the way the weights are calculated. The variations of learning rule are described in Section 3.

2 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 w_{ij} is the weight on the connection from unit j to unit i.

The deterministic dynamics of the network is given by:

$$S_i' = \begin{cases} 1 & \text{if } h_i > 0 \\ -1 & \text{if } h_i < 0 \\ S_i & \text{if } h_i = 0 \end{cases}$$

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 said to be a *fundamental memory*, and is successfully stored.

In the stochastic case the update rule is generalised to be probabilistic: whenever a unit is chosen for updating its next state is given by:

$$Prob\left(S_i' = \pm 1\right) = \frac{1}{1 + exp\left(\mp \frac{2h_i}{T}\right)}$$

where T is the temperature of the network.

Now a symmetric weight matrix and asynchronous updates guarantee that the network reaches *equilibrium*. That is the *average over time* of the state of each unit in the network $\langle S_i \rangle$, eventually becomes constant.

With deterministic dynamics (T= 0) a network state is stable if, and only if, all the local fields are of the same sign as the corresponding unit, equivalently the *aligned local fields*, $h_i S_i$, should be positive.

In the stochastic case a sufficiently high temperature will result in ergodic dynamics, in which there are no attractors: all states are equally likely ($\langle \mathbf{S} \rangle = 0$). At lower temperatures, however the network can be in equilibrium with a non-zero, mean state vector. If this vector has a large overlap with a particular network state then it is sensible to say that the network has converged on that state. Thus if $\langle \mathbf{S} \rangle . \xi^{\mu} > 1 - \varepsilon$ then the training pattern μ is once again a fundamental memory.

3 LEARNING RULES

In the basic Hopfield model the weights are given by a one-shot Hebbian rule: $w_{ij} = \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu}$. The resulting network (with deterministic dynamics) performs reasonably at low loading but as the loading increases performance becomes progressively worse, until at a loading of above 0.14N, the network will no longer store the training patterns. Moreover correlation of the training patterns significantly worsens performance. Two other types of training can be used with this type of network, both of which give much higher capacity and performance than the one-shot rule. The first is to use learning rules that find approximations to the projection weight matrix, in which any linearily independent set of patterns can be learnt. The second is to use perceptron like training, continuing until all the local fields are correctly aligned. The capacity of this rule is at least 2*N* (for large N).

3.1 Psuedo-Inverse Learning (PI)

The projection weight matrix is given by: $\mathbf{W} = \Xi\, \Xi^{-1}$ where Ξ is the matrix whose columns are the ξ^{μ}, and $\Xi^{-1}$is its *pseudo-inverse*, the matrix with the property that: $\Xi^{-1}\, \Xi = \mathbf{I}$ (Ξ^{-1} exists if the patterns have no linear dependencies). It is a symmetric matrix.

A variety of iterative methods are available for approximating **W**, some of which are in the spirit of neural computational methods [2,3].

Here we use the Blatt and Vergini algorithm [3]:

```
Beginning with a zero weight matrix
For each pattern in turn
  Clamp the pattern onto the network and set s = 0
        Repeat until $\sum_{i,\mu} \left|1 - h_i^{\mu} \xi_i^{\mu}\right| < \varepsilon$
            Increment s
    For each processing element in turn
        Update incoming weights according to:
```

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

The parameter *k* can be any value that satisfies 1 < k ≤ 4. Since the larger the value of *k* the faster the rule converges, we set *k* to 4.

3.2 Repeated Hebbian Learning (SLL)

It is also possible to use the perceptron learning rule to find a set of weights that will produce correctly aligned local fields. Originally suggested in this context by Forest[4], the rule in its simplest form is:

```
Begin with a zero weight matrix
Repeat until all aligned  local fields are correct
Set the state of network to one of the $\xi^{\mu}$.
    For each unit, i, in turn
    Calculate $h_i^{\mu} \xi_i^{\mu}$, If less than M
     then set $\Delta w_{ij} = \Delta w_{ji} = \frac{\xi_i^{\mu} \xi_j^{\mu}}{N}$
```

M (> 0) is the *learning threshold* (or *margin*). The larger *M* the better the attractor performance of the network is likely to be. Earlier work [5] has shown that a value of 10 is suitable for networks of the size used here.

The symmetric form of the weight update ensures that the final weight matrix is symmetric.

4 ANALYSING PERFORMANCE

4.1 Energy of Attractors

The energy of a state, ***S***, in a network with symmetric weights is:

$$E\{\boldsymbol{S}\} = -\tfrac{1}{2} \sum_{i,j} w_{ij} S_i S_j = -\tfrac{1}{2} \sum_{i} h_i S_i .$$

With deterministic dynamics the energy of the network always decreases when a unit changes state. The stable points of the dynamics are therefore (local) minima of the energy, and the actual minima reached is largely determined by the initial state of the network.

With stochastic updates the network minimises its free energy[1], which is a function of the probability distribution of the network at any time:

If $\sigma_t : \{S\} \rightarrow [0,1]$ is a probability distribution of states, then the free energy is:

$$F(\sigma_t) = \langle E\{S\} \rangle - T.H(\sigma_t)$$

where $H(\sigma_t)$ is the entropy of the distribution.

This means that the network is minimising its mean energy whilst maximising its entropy, with the balance mediated by the temperature of the network.

The effect of noise on the dynamics of these networks can be beneficial if the desired attractors (the trained patterns) have lower (more negative) energy than the spurious ones. So even when the network passes close to a local minima in the energy function the presence of noise may prevent this from becoming dominant, in equilibrium.

The first analysis undertaken is of the average energy of random attractors in the network. The network is started in a random state and allowed to relax, under noiseless dynamics, into an attractor, which is then identified as one of the training patterns or otherwise. Its energy is calculated and finally the mean of these energies is reported.

4.2 Affect of Noise

The second analysis seeks to ascertain how the addition of noise affects the probability of a random state arriving at one of the training patterns. When the network is stochastic it will never actually converge to a specific state. To estimate its asymptotic behaviour we allow the network to evolve from its starting state and after a specific, large number of updates, find the maximum overlap of the final state with each of the stored patterns. That is $\max_{\mu} \left\{ \frac{\mathbf{S}.\xi^{\mu}}{N} \right\}$ is calculated. Whenever this is larger than a prespecified threshold the network is designated as having converged on a stored pattern.

The network is started in a number of random initial states, at a variety of temperatures and the nature of its final states are then identified.

4.3 Basins of Attraction

We use, R, the normalised mean radius of the basins of attraction[6], 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 (in the sense described above) 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 [7].

5 RESULTS

5.1 Energy of Attractors

As described above the energy of the attractors in the networks was estimated by taking 10,000 randomly chosen initial states and allowing the network to relax to an attractor, whose energy is then calculated. The absolute values of the energies are dependent on the size of the network weights, so that it is not sensible to compare energies across different types of network. Rather we are interested in the relative values of the attractor states corresponding to trained patterns and all other spurious attractors. We take $\Phi = \frac{\langle \mathrm{E}_{\text{trained attractor}} \rangle}{\langle \mathrm{E}_{\text{spurious attractor}} \rangle}$. As the energies are negative a value of Φ above 1 shows the trained attractors to have lower energy than the spurious ones.

Table 1 shows the results for networks trained with uncorrelated random patterns. The results are averages of 10 different runs.

For both of the high capacity rules the energy of the fundamental memories is lower than that of the spurious attractors. As the loading increases the difference in energies is decreased, with both SLL and PI showing very similar values. It is therefore possible that under noisy conditions the higher energy spurious attractors maybe destabilised, whist the desired attractors remain. This possibility is investigated next.

Patterns	Hopfield	SLL	PI
10	1.01	1.32	1.33
20	0.85	1.24	1.24
30	-	1.19	1.19
40	-	1.14	1.15

Table 1: Φ values for one hundred unit networks trained with uncorrelated random patterns. The trained networks are started in 10,000 random initial states and allowed to relax to an attractor under deterministic dynamics. The results are averages of 10 different training sets, at each loading. Results are not reported for the Hopfield learning rule at loadings higher than 20 as the trained pattern attractors appear very rarely.

5.2 Effect of Noise

In this experiment trained networks are tested

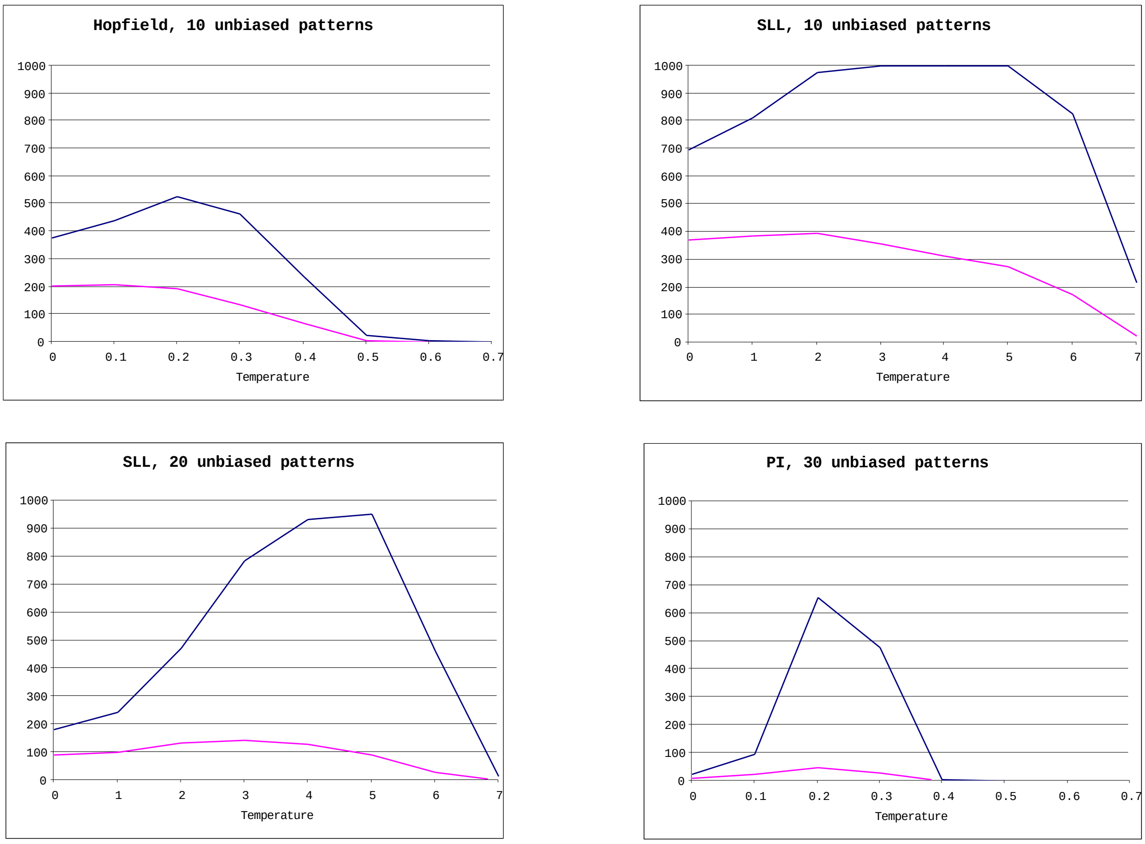

Figure 1: A selection of100 unit networks trained with 10, 20, and 30 patterns, the P.I. and SLL weight matrix are used. The standard Hopfield net with a loading of 10 patterns is also shown for reference. 1000 random initial states are allowed to relax over 10,000 unit updates. The number of fundamental memories reached is shown as well as the number of correct fundamental memories. Results are averages over 10 different runs. The upper line is the number of fundamental memories reached, the lower is the number of correct memories reached.

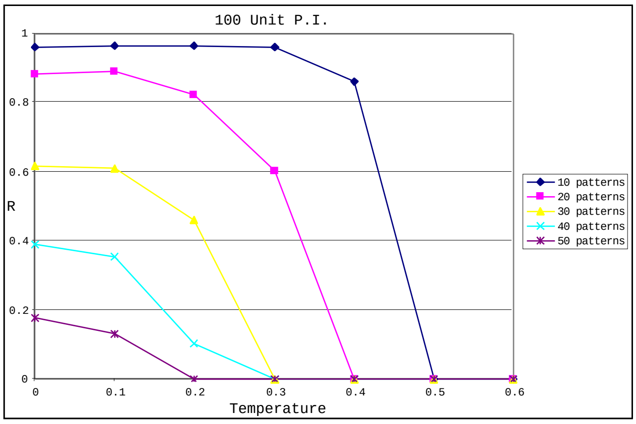

Figure 2: The R values for the pseudo inverse weight matrix at varying loadings and temperatures. Results are averages over 50 runs for each data point.

under a variety of temperature settings for the dynamics. Again the networks are started in a number (1000) of random states and allowed to evolve, in this case for a large, fixed number of unit updates. All results are for 10,000 individual unit updates, or equivalently, for these 100 unit networks, 100 epochs. At the end of this process the final state is compared with all training patterns; if the overlap with any training pattern is more than 0.9 then the network is designated as having evolved to one of the fundamental memories. When the final fundamental memory is the one that was originally closest to the initial state then the network is said to have reached the *correct* attractor.

Figure 1 shows the results at a variety of loadings and temperatures for both SLL and PI learning; the results for the standard rule are also shown at a loading of 10 patterns. The results for both SLL and PI are similar: the addition of noise produces a dramatic increase on the number of fundamental memories found. This can be seen very clearly in the SLL 0.2 loading results. With deterministic dynamics 0.17 of the initial states reaches a fundamental memory. When the temperature is 0.3, 0.99 reach a fundamental memory – spurious attractors have been almost completely eliminated.

The affect on the number of correct fundamental memories found is not as dramatic, although a small increase is present in all cases.

As loading increases the best temperature for destabilising spurious attractors decreases, suggesting that the phase diagram, for this network, in the temperature-loading plane is similar to the one derived for the standard learning rule [8].

5.3 Basins of Attraction

In these experiments the mean normalised radii of the basins of attraction, associated with fundamental memories is estimated.

Figure 2 shows how R varies with loading and temperature, for the pseudo inverse weight matrix. The SLL results are very similar. It can be seen that R is not raised by an increasing temperature. In fact above a fairly low noise level there is a rapid fall in the value of R.

6 CONCLUSIONS

The first set of results showed that, for both types of high capacity weight matrix, the energy of the fundamental memories was lower than that of the spurious attractors. This is potentially helpful and suggests that introducing noise to the dynamics could be beneficial. The second set of results confirmed this conclusion. The addition of noise, at relatively low loadings, completely eliminated the unwanted attractors and at higher loadings caused a significant reduction in their relative frequency. However the results showed that whilst random initial states were more likely to evolve into one of the learned patterns the probability that this learned pattern was the one closest to the initial state was not increased. This is a consequence of the noisy dynamics: in comparison with the deterministic case the dynamic process is less determined by the initial conditions. The final set of results confirms this view. There is no noise level which produces an enlargement in the basins of attraction, in fact as the amount of noise in the system increases the attractor basins decrease in size.

Despite this the addition of noise is of great benefit. Deterministic associative memory models are handicapped by the number of spurious attractors and their proclivity to find these attractors from many starting states, so that reducing their frequency, by the addition of noise, makes the models much more attractive both as computational artefacts and as neurophysiological models.

7 REFERENCES:

[1] Amit, D.J., *Modelling brain function: the world of attractor neural networks*. 1989, Cambridge, UK: Cambridge University Press.

[2] 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.

[3] Blatt, M.G. and E.G. Vergini, *Neural networks: a local learning prescription for arbitrary correlated patterns.* Physical Review Letters, 1991. **66**(13): p. 1793-1797.

[4] Forrest, B.M., *Content-addressability and learning in neural networks.* Journal of Physics A, 1988. **21**: p. 245-255.

[5] Davey, N., R.G. Adams, and S.P. Hunt. *High Performance Associative Memory Models and Symmetric Connections*. in *International ICSC Congress on Intelligent Systems and Applications (ISA 2000)*. 2000.

[6] Kanter, I. and H. Sompolinsky, *Associative Recall of Memory Without Errors.* Physical Review A, 1987. **35**(1): p. 380-392.

[7] Davey, N. and S.P. Hunt. *The Capacity and Attractor Basins of Associative Memory Models*. in *5th International Work-Conference on Artificial and Natural Neural Networks, IWANN'99*. 1999. Alicante, Spain: Springer-Verlag.

[8] Hertz, J., A.Krogh and R.G.Palmer, *Introduction to the Theory of Neural Computation*. 1991: Addison-Wesley.