

The Effects of Changing the Update Threshold in High Capacity Associative Memory Model

Weiliang Chen, Rod Adams, Lee Calcraft and Neil Davey

School of Computer Science

University of Hertfordshire

Hatfield, Herts, AL10 9AB, United Kingdom

e-mail: {W.3.Chen, r.g.adams, l.calcraft, n.davey}@herts.ac.uk

Abstract: *It has been found that the performance of an associative memory model trained with the perceptron learning rule can be improved by increasing the learning threshold. When the learning threshold increases, the range of possible values of the update threshold becomes wider and the network may perform differently with different choices of this parameter. This paper investigates the effect of varying the update threshold. The result indicates that a non-zero choice of update threshold may improve the performance of the network.*

Keywords: *associative memory, Hopfield network, perceptron, dynamics, update threshold*

1. Introduction

One important type of Artificial Neural Network (ANN) is the Associative Memory (AM) model. This model is used to investigate how neural networks can be used to perform as “content-addressable memory”, where the memorized patterns can be recalled from part of their uncompleted contents.

It has been found that in the AM model trained with perceptron learning, the performance of the network can be improved by increasing the learning threshold used in the learning algorithm [1, 4]. On the other hand, the effects of varying the update threshold have not been investigated yet. This paper reports an investigation into this issue.

Our main result indicates that a non-zero update threshold may improve the performance of the network.

2. Background

An AM model usually includes two processes, the training process and the network dynamics. Given a network of N units and a set of N -ary, bipolar (+1/-1) training patterns, $\{\xi^p\}$, $\xi^p = [\xi_0^p, \xi_1^p, \xi_2^p, \dots, \xi_N^p]$, the model learns patterns by modifying the N by N weight matrix denoted by W . After training, a specific pattern of unit states is first presented to the network. The network state is then modified according to an update rule that defines the network dynamics.

The most well-known and fundamental AM model is the Hopfield network with a one-shot Hebbian learning rule [6], whose training process can be described as follows:

Denoting the weight of the connection from unit j to unit i in W by w_{ij} , for each pattern from $\{\xi^p\}$ and each unit, update the weights according to:

$$w_{ij} = \frac{1}{N} \sum_p \xi_i^p \xi_j^p \quad \text{if } i \neq j \quad (1)$$

$$w_{ij} = 0 \quad \text{if } i = j$$

In the dynamics of this model, the changes of unit states are determined by the unit's net input, or *local field*, given by $h_i = \sum_{j \neq i} w_{ij} S_j$, where S_j is the current state of unit j . The new state of a unit after update is given by:

$$S'_i = \begin{cases} 1 & \text{if } h_i > \varphi \\ -1 & \text{if } h_i < -\varphi \\ S_i & \text{otherwise} \end{cases} \quad \text{where } S'_i \text{ is the new state of } S_i. \quad (2)$$

φ is defined as the *update threshold* and is normally set to 0.

The update of unit states can be either synchronous or asynchronous. During the dynamics the network may evolve to a fixed point. If a pattern in $\{\xi^p\}$ is one of the fixed points of the network then this pattern is successfully stored and is considered a *fundamental memory*.

Although widely studied [1, 4, 9], the standard Hopfield network has a critical drawback which restricts its application. It has been proved that this kind of model has quite a low storage capacity, which is approximately $0.14N$, given an N unit, fully connected network. On the other hand, another kind of AM model, classified as the ‘‘Gardner Class’’ by Abbott [1] due to the original contribution of Gardner [5], has a significantly higher storage capacity of $2N$. The model examined in this paper, using the perceptron learning rule, belongs to this class.

Unlike the one-shot Hebbian rule used in the standard Hopfield network, the model examined in this paper uses an iterative learning algorithm based on the *aligned local field* of a unit, given by $h_i^p \xi_i^p$, and a non-negative parameter, the *learning threshold*, denoted by T . The whole process of training can be described as:

Begin with a zero weight matrix

Repeat until all *aligned local fields* are not less than T

Set the state of the network to one of the ξ^p

For each unit, i , in turn

Calculate aligned local field $h_i^p \xi_i^p$

If this is less than T then change the weight on connections into unit i according to:

$$\forall j \neq i \quad w'_{ij} = w_{ij} + \frac{\xi_i^p \xi_j^p}{N}, \quad (3)$$

The original network dynamics as described earlier in (2), is still used in the examined model. After convergence, since all local fields of units of training patterns are driven to the correct side of $\pm T$ as appropriate, it is guaranteed that all training patterns are stable and become fundamental memories of the network.

The result of a previous study [4] indicates that the performance of these networks can be improved by increasing the learning threshold from 0 to 10. On the other hand, there is little or no improvement in performance when increasing it from 10 to 100. Thus in the experiments here the learning threshold, T , is always set to 10.

With perceptron training, all the aligned local fields of every unit of every pattern in the training set will be at least as big as the learning threshold, T . That is $\forall i, \mu \quad \xi_i^\mu h_i^\mu \geq T$. This means that the update threshold can be varied up to a value of T , without destabilizing the training patterns. Thus the effects of varying the update threshold become interesting to us, which is the motivation for the experiments undertaken here.

3. Performance Measures

Two measures are applied to evaluate the performance of the network: the *normalised mean radius of the basins of attraction* and the *Effective Capacity*.

The pattern correction ability of the network is measured by R , the *normalised mean radius of the basins of attraction*, as a measure of attractor performance in these networks [7]. Details of the algorithm used can be found in [4]. A value of $R = 1$ implies perfect performance and a value of $R = 0$ implies no pattern correction.

The second metric that we use to measure the performance is the *Effective Capacity* of the network, EC [2]. The Effective Capacity 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. We take a fairly arbitrary definition of reasonable as correcting the addition of 60% noise to within an overlap of 95% with the original fundamental memory. In the experiments in this paper, we also varied the percentage of noise added to the fundamental memories. For large fully connected networks the EC

value is about 0.1 of the conventional capacity of the network.

More details about these two measures can be found in [3].

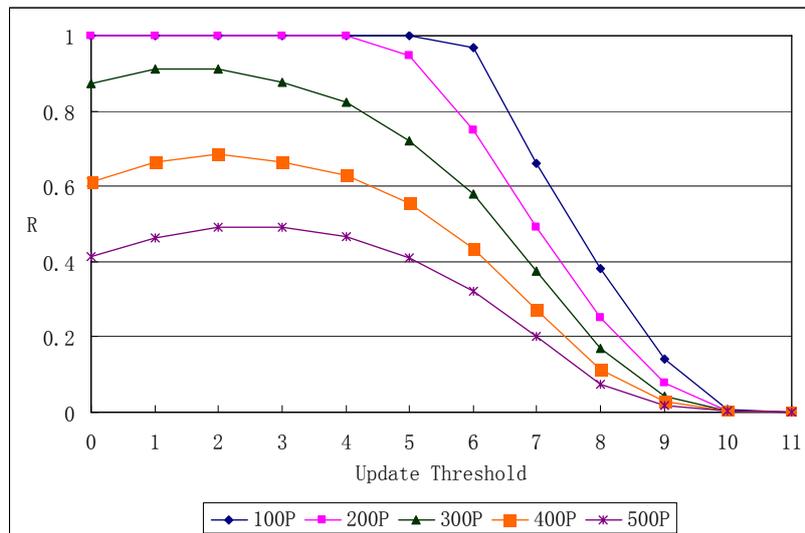
4. Experiments and Results

All experiments were conducted on a 1000 unit, fully connected network without self-connections. Training sets were generated randomly without bias. As described in the previous section, the learning threshold, T , was set to 10, therefore the update threshold of the dynamics (see equation (2)), φ , could be varied within a reasonable range from 0 to 11. Two sets of experiments were performed, each set using a different performance measure. The first set of experiments measured the normalised mean radius of the basins of attraction. The Effective Capacity of the network was measured in the second set of experiments. Each set of experiments was repeated 20 times and average values are reported here.

4.1 Performance Measured by the *normalised mean radius of the basins of attraction*

The performance of R was measured in 5 experiments, with different numbers of training patterns ranging from 100 to 500. In each experiment the update threshold φ was varied from 0 to 11. R values were measured and Figure 1 contains the results.

Figure 1: Normalised mean radius of the basins of attraction for different update threshold. Experiments run on a 1000 unit, fully connected network without self-connections. The number of training patterns in each experiment varies from 100 to 500 (denoted by 100P to 500P in legend).



As expected, R increases when the number of training patterns (network loading) decreases. Perfect performance is achieved with a low loading (100 to 200 patterns) when φ is set to 0. Since the learning threshold is 10, it is likely that all aligned local fields of the training patterns are above 10 but

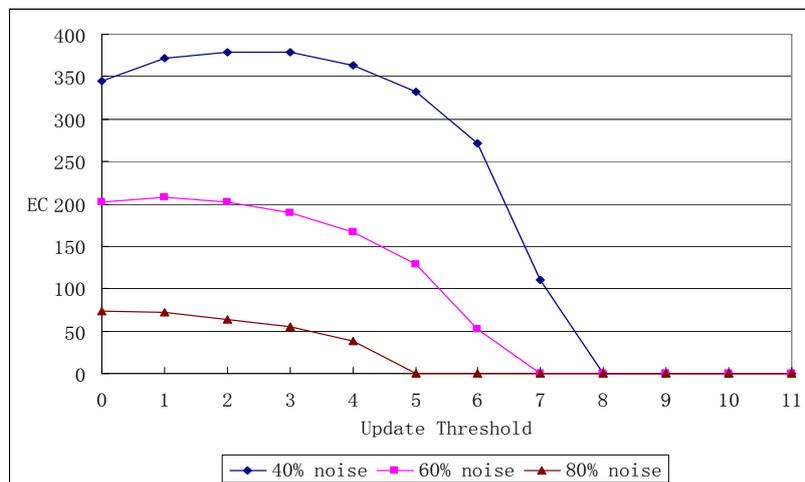
not too far from 10. Hence we expect performance to be poor when φ is set to 11. The results show that this is in fact the case.

The most interesting finding in these experiments is how R changes in each experiment, when φ increases from 0 to 11. The result shows that the relationship between R and φ is far from a simple linear one. In all the experiments, as φ increases, R tends to first increase (or to stay the same if it has already achieved perfect performance) then reduce to zero fairly quickly. In those experiments which do not start with perfect performance, the best R value is achieved with a non-zero value of φ between 1 and 3. A possible explanation of the results is given in the discussion below.

4.2 Performance Measured by the *Effective Capacity*

The performance according to Effective Capacity, EC , was measured in 3 experiments, by increasing the *noise percentage* from 40% to 80%, whilst keeping the *overlap criterion* at 95% throughout. The update threshold, φ , was again varied from 0 to 11. Results (Figure 2) indicate that the performance of EC drops down to 0 with a high setting of φ (8 with 40% noise, 7 with 60% noise and 5 with 80% noise). Again an improvement with a non-zero update threshold is also found in some of these experiments, with better performance for update threshold values between 1 and 4 for the 40% noise version. The improvement in the low noise percentage experiment is greater than the ones with a higher noise percentage. No improvement is seen in the series of experiments with 80% noise.

Figure 2: Effective Capacity for different update threshold. Experiments run on a 1000 unit, fully connected network without self-connections. The overlap criterion is set to 95%. All training patterns are generated randomly without pattern bias.



5. Discussion and Conclusion

The effect of varying the update threshold in the high capacity Associative Memory model examined in this paper can be summarized as follows. We found that in some circumstances using a non-zero update threshold does improve network performance. For example, in a network with a learning threshold of 10, using an update threshold of a small number such as 2, instead of zero will normally

improve performance, especially with a high loading.

It is possible to give an intuitive explanation of the results above. In the convergence process during recall, a pattern will relax to one of the fixed points of the network, as described in Section 2. Some of these fixed points, described as fundamental memories, are the training patterns. However, not all the fixed points belong to this type. Fixed points which are not fundamental memories are called *parasites*, and they may disrupt the performance of the network. Increasing the update threshold gradually increases the number of fixed points in the network until every pattern is a fixed point. This is probably the case when the update threshold bigger is than 10 as in the earlier experiments. However, a small increase of the update threshold, from 0 to 2 for instance, increases the probability that the network relaxes to a fundamental memory. Therefore, the network performance is initially improved but then drops as the update threshold is increased.

Furthermore, it is known that neurons are organized into modules in the mammalian cortex [8]. It will therefore be interesting to investigate the interaction between groups of units with each group having a different update threshold.

References

- [1]. Abbott, L.F., *Learning in neural network memories*. **Network: Computational Neural Systems**, 1990. 1: p. 105-122.
- [2]. Calcraft, L., *Measuring the Performance of Associative Memories*, in *Computer Science Technical Report*. 2005, University of Hertfordshire.
- [3]. Davey, N., Calcraft, L., and Adams, R., *High Capacity, Small World Associative Memory Models*. To appear in *Connection Science*.
- [4]. Davey, N., Hunt, S.P., and Adams, R.G., *High capacity recurrent associative memories*. *Neurocomputing*, 2004. 62: p. 459-491.
- [5]. Gardner, E., *The space of interactions in neural network models*. *Journal of Physics A*, 1988. 21: p. 257-270.
- [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]. Kanter, I. and Sompolinsky, H., *Associative Recall of Memory Without Errors*. *Physical Review A*, 1987. 35(1): p. 380-392.
- [8]. Mountcastle, V.B., *The columnar organization of the neocortex*. *Brain*, 1997. 120(4): p. 701-722.
- [9]. Schultz, A., *Five Variations of Hopfield Associative Memory Network*. *Journal of Artificial Neural Networks*, 1995. 2(3): p. 285-294.