HIGH CAPACITYASSOCIATIVE MEMORY MODELS – BINARY AND BIPOLAR REPRESENTATION

Neil Davey, Ray Frank, Steve Hunt, Rod Adams and Lee Calcraft
Department of Computer Science, University of Hertfordshire,
Hatfield, AL10 9AB.
UK
{N.Davey, R.J.Frank, S.P.Hunt, R.G.Adams, L.Calcraft}@herts.ac.uk

ABSTRACT
Hopfield type associative memory networks usually use a bipolar representation. It is also possible to use a binary 0/1 representation, although in the standard model this lowers performance. This paper reports an empirical investigation into the performance of both binary and bipolar associative memories, trained using the simple perceptron learning rule. Such networks normally have much better performance than the standard model. It is found that the binary networks perform just as well as the bipolar networks, although they take significantly longer to train.

KEY WORDS
Associative memory, Hopfield Networks, Binary representation, Bipolar representation

1. Introduction

High capacity associative memory models can be constructed from networks of perceptrons, trained using the normal perceptron training procedure [1, 2]. 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 a maximum of 2N random unbiased patterns, and this capacity ($\alpha = 2$) is increased if the training set is correlated [3].
In the standard model the units in the network are bipolar, taking either the value +1 or –1. It is also possible to use a binary, 0/1, network, and these two models can be shown to be functionally equivalent [4]. However the choice of representation can affect the speed and efficacy of the learning rule. For example the standard covariance matrix (one shot Hebbian learning), together with 0/1 states, gives only half the capacity of the same matrix with the bipolar representation [5].
The simple perceptron learning rule is quite different when the patterns to be learnt are binary as opposed to bipolar; with binary patterns, learning only takes place on active connections, that is on afferent connections from units in the +1 state. In the bipolar case learning takes place on all incoming connections. The binary perceptron network is therefore interesting as it does not have the biologically implausible nature of the bipolar learning.
Here we conduct an comparative empirical investigation into the behaviour of bipolar and binary learning rules.
The paper first introduces the basic model. In Section 3 the formal equivalence of bipolar and binary networks is demonstrated. Section 4 discusses the four different forms of perceptron learning rules used, and Section 5 describes the various performance measures. In Section 6 the results are presented and the paper concludes with a discussion

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 S is the current state and w_{ij} is the weight on the connection from unit j to unit i. We use S to denote a bipolar state, S = ±1 and σ for a binary state, σ = 0/1.
The dynamics of the network is given by the standard update:

$$S_i' = \begin{cases} 1 & \text{if } h_i > \theta_i^S \\ -1 & \text{if } h_i < \theta_i^S \\ S_i & \text{if } h_i = \theta_i^S \end{cases} \quad \text{where } \theta_i^S \text{ is the unit threshold}$$

or in the binary case:

$$\sigma_i' = \begin{cases} 1 & \text{if } h_i > \theta_i^\sigma \\ 0 & \text{if } h_i < \theta_i^\sigma \\ \sigma_i & \text{if } h_i = \theta_i^\sigma \end{cases} \quad \text{where } \theta_i^\sigma \text{ is the unit threshold}$$

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 is one of these fixed points then it is successfully stored, and said to be a *fundamental memory*.

3. Equivalence of bipolar and binary networks

Given a set of weights for either the bipolar or binary network, an identically functioning network with the other type of unit can be constructed [4]. For example if the units are bipolar (taking $\theta_i^S = 0$ for simplicity) then the transformation to an equivalent network with binary states is accomplished with the mapping:

$$\sigma_i = \frac{S_i + 1}{2},\ w_{ij}^{\sigma} = 2w_{ij}^{S},\ \theta_i^{\sigma} = \sum_j w_{ij}^{S}$$

So that:

$$\begin{aligned}
&S_i' = 1 \\
&\Leftrightarrow h_i = \sum_{j \neq i} w_{ij}^S S_j > 0 \\
&\Leftrightarrow \sum_{j \neq i} w_{ij}^S (2\sigma_j - 1) > 0 \\
&\Leftrightarrow \sum_{j \neq i} 2w_{ij}^S \sigma_j > \sum_{j \neq i} w_{ij}^S \\
&\Leftrightarrow \sum_{j \neq i} w_{ij}^{\sigma} \sigma_j > \theta_i^{\sigma} \\
&\Leftrightarrow \sigma_i' = 1
\end{aligned}$$

The networks therefore will have identical dynamics. It is worth noting that with random uncorrelated training patterns the unit thresholds, θ_i^{σ}, will approach zero as the size of the network increases (being a sum of random numbers, and assuming the weights are scale free with respect to the size of the network) so that the network will work with binary units just as with bipolar units.

4. Learning

To train a network of perceptrons to act as an associative memory, the input and output layers consist of the same set of neurons. The weights can then be trained using any perceptron training procedure, so that the network auto-associates. See Figure 1.

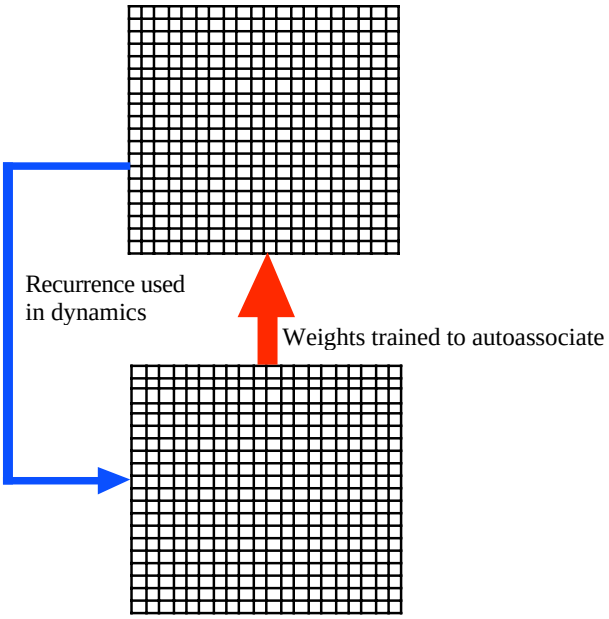

Figure 1: A set of perceptrons, used as an associative memory.

The actual training rule we use (in both cases, binary and bipolar) is the simple perceptron rule, with learning threshold *T*. So if the training set consists of the patterns $\{\boldsymbol{\xi}^p\}$ and if the network has *N* units the learning rule is:

```
Begin with zero weights
Repeat until all units are correct
  Set state of network to one of the ξ^p
  For each unit, i, in turn:
   Calculate its net input h_i^p.
   If (ξ_i^p = on and h_i^p < T) or (ξ_i^p = off and h_i^p > -T)
   then change all the weights to unit i
    according to:
```

$$w_{ij} = w_{ij} + \frac{\xi_j^p}{N-1} \quad \text{when } \left(\xi_i^p = on \text{ and } h_i^p < T\right)$$

$$w_{ij} = w_{ij} - \frac{\xi_j^p}{N-1} \quad \text{when } \left(\xi_i^p = off \text{ and } h_i^p > -T\right)$$

```
  The value ξ_i^p = on denotes the ith bit of
  pattern p being +1
  and the value ξ_i^p = off denotes the value
  -1 or 0 according to the type of network
```

Here then the learning rate is $\frac{1}{N-1}$, and is thus inversely proportional to the number of connections each unit makes, thereby ensuring the weights are scale free with respect to the size of the network.

Now the key point to note is that with the binary representation no weight changes can occur on inactive inputs, since in this case $\xi_j^p = 0$. However with the bipolar representation weights will change on every input weight, whenever a unit is incorrect. This will therefore

cause the two types of network to arrive at different weight matrices.

Of course the perceptron convergence theorem guarantees that if a set of weights exist that solve the problem then the perceptron learning rule will converge upon it, regardless of the representation used, and an upper bound can be put on the number of steps, *M*, required [6]. In the bipolar case: $M < \frac{(N-1)(1+2T)}{D^2}$, where D is the, training set dependent, *difficulty* of the learning task (the smaller the possible set of solutions the smaller is D, and the harder is the learning task).

4.1 Weight Symmetry

We also examine one further modification of the learning rule. In the standard Hopfield network the weights are symmetric ($w_{ij} = w_{ji}$), a sufficient condition for guaranteeing point attractors only. The perceptron learning rule described above does not produce symmetric weights, but it is easy to modify it to do so [3]. The idea is simply to always change both w_{ij} and w_{ji} together, effectively halving the number of independent weights in the network. Remarkably this does not reduce the capacity or performance of the network [7]. The symmetric training procedure is therefore:

```
Begin with zero weights
Repeat until all units are correct
  Set state of network to one of the ξ^p
  For each unit, i, in turn:
   Calculate its net input h_i^p.
   If (ξ_i^p = on and h_i^p < T) or (ξ_i^p = off and h_i^p > -T)
    then change the weights to unit i and j
    according to:
```

$$\left.\begin{aligned} w_{ij} &= w_{ij} + \frac{\xi_j^p}{N-1} \\ w_{ji} &= w_{ji} + \frac{\xi_j^p}{N-1} \end{aligned}\right\} \text{ when } \left(\xi_i^p = on \text{ and } h_i^p < T\right)$$

$$\left.\begin{aligned} w_{ij} &= w_{ij} - \frac{\xi_j^p}{N-1} \\ w_{ji} &= w_{ji} - \frac{\xi_j^p}{N-1} \end{aligned}\right\} \text{ when } \left(\xi_i^p = off \text{ and } h_i^p > -T\right)$$

5. Performance Measures

We compare the performance of the two types of representation empirically. We use random, unbiased (equal probability of ±1 or 0/1) training sets of various sizes. We report four measures of performance as described below.

5.1 Stability Measure

The learning rules drive the net inputs of the units in the network to the correct side of the learning threshold *T*. Increasing *T* may improve the attractor performance of the network [8]. Some care must be taken though, since if we consider a network in which all the training patterns are stable, that is $h_i > T$ or $h_i < -T$ as appropriate, for all patterns, and units, *i*, then any uniform, upward scaling of the weight matrix will increase the magnitude of the h_i but will obviously not increase the attractor performance. In fact the optimal attractor performance is achieved when the threshold is maximised with respect to the size of the weights. For this reason the relevant characterization is the *normalised stability measure*, defined as:

$$\gamma_i^p = \begin{cases} \dfrac{h_i^p}{|\mathbf{w}_i|} & \text{if } \xi_i^p = 1 \\ \dfrac{-h_i^p}{|\mathbf{w}_i|} & \text{otherwise} \end{cases}$$

where $\mathbf{w}_i$ is the incoming weight vector to unit i. The minimum of all the γ_i^p therefore gives a measure of the likely attractor performance [9] and we take $\kappa = \min_{i,p}\left(\gamma_i^p\right)$

5.2 Attractor Basin Size

The key performance indicator in this type of network is the size of the basins of attraction of the trained patterns. Throughout this work the measure used is the *mean normalised radius* of these basins denoted as *R* [10].

The way this has been estimated throughout the work presented here is as follows.

For each of a set of sample states (50 here) a fixed fraction, m_0, of the state is made identical to the corresponding part of one of the stored patterns, $\boldsymbol{\xi}^p$, and the rest of the state is random. Each of these sample states is then required to relax, under the dynamics of the system, to the correct $\boldsymbol{\xi}^p$. An incremental search is undertaken for the smallest value of m_0 for which this happens. Initially a low value is taken for m_0 and consequently it needs to be incrementally increased until all of the sample states relax to a $\boldsymbol{\xi}^p$. Averaging the final values of m_0 over different sets of stored patterns yields:

$R = 1 - \langle m_0 \rangle$

As is pointed out in [11] for finite size associative memories, another factor needs to be considered. Each of the initial states used in this calculation may overlap one of the other stored patterns more closely than the original $\boldsymbol{\xi}^p$, and to compensate for this the definition of *R* is modified to:

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

where m_1 is the overlap with the closest of the other stored patterns. The double average for m_1 is taken over

the 50 different starting points and over the different sets of patterns.

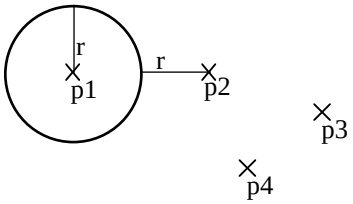

Figure 2: Calculating *R*. In this figure p1, p2, p3 and p4 are patterns in the training set. The closest pattern in the training set to p1 is p2, at a distance of 2r. Optimal performance occurs when all vectors within the hypersphere centred on p1 and radius r, are attracted to p1. If all patterns stored in a network exhibit this performance, its normalised average basin of attraction, *R*, is 1

A value of $R = 1$ denotes perfect attractor performance and a value of 0 signifies that any single bit flip of a trained pattern will not be corrected.

5.3 Training Times

The next performance measure we report is the training time required to learn the training set. This is quantified as the number of epochs (complete presentations of the training set) needed for convergence of the training process.

5.4 The Weight Symmetry

The final measure we report is the symmetry of the weight matrix for the non-symmetric rule. This is calculated as: $\rho = \frac{\sum_{i,j} w_{ij} w_{ji}}{\sum_{i,j} w_{ij}^2}$. For a symmetric matrix this takes the value +1. For an anti-symmetric matrix it takes the value –1 and for a random set of weights it will be roughly zero.

6. Results

In this section we evaluate the binary and bipolar networks trained both symmetrically and non-symmetrically. All the networks used consist of 100 units. The loading on the network is varied from 10 patterns to 75 in steps of one. At each loading the networks are evaluated with 50 random training sets of uncorrelated data. We denote the four types of network as shown in Table 1:

Network Type	Non-Symmetric Weights	Symmetric Weights
Bipolar	*Bipolar-NS*	*Bipolar-S*
Binary	*Binary-NS*	*Binary-S*

Table 1: The names given to the four different networks.

6.1 Training Times

Figure 3 shows the number of epochs required to train the networks. It is firstly apparent that the binary networks take longer to train than the bipolar ones. This is unsurprising as at any given presentation of a training pattern fewer weight changes are made in the binary learning. It is also clear that in both types of network, symmetric networks can be trained more quickly than the non-symmetric ones. Again this is not surprising: symmetric networks have only half the number of independent weights of the equivalent non-symmetric model.

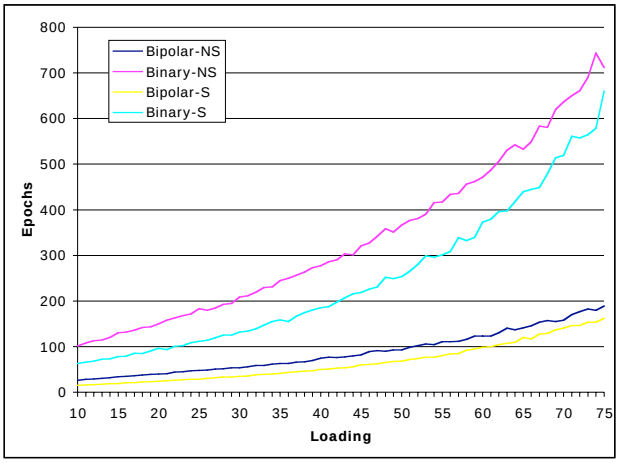

Figure 3: The training time, in epochs, for each of the four different types of network.

6.2 Stability Values

As described earlier the minimum value of the normalised stability measures, κ, indicates how well the network is likely to perform. Figure 4 shows the κ values for all four types of network.

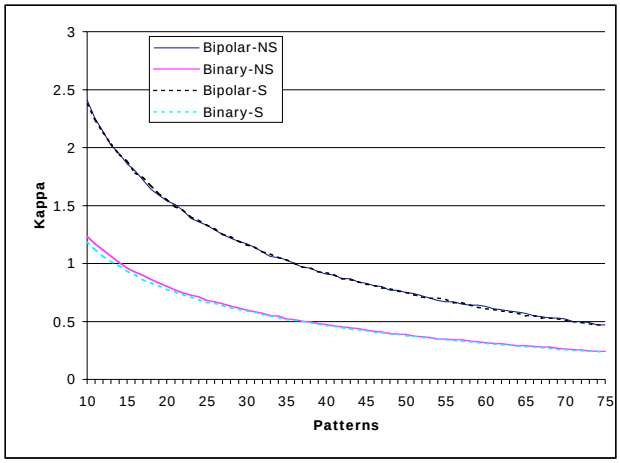

Figure 4: The κ values for the four types of network. The Bipolar networks have almost identical κ values, upper pair of lines, with the Binary networks having values half that of the Bipolar networks, lower pair of lines.

It can be seen that the symmetric and non-symmetric learning rules produce very nearly identical kappa values (the plots very nearly coincide). The binary networks produce kappa values that are almost exactly half that of the bipolar nets; this is in accord with the network equivalence mapping described in Section 2 in which the weights of the binary network are twice the size of the bipolar equivalent.

6.3 Size of the Basins of Attraction

As described earlier the key performance measure for associative memory networks is the size of the basins of attraction of the trained memories. Figure 5 shows the measured *R* values. It is immediately apparent that all four variations of the model produce almost identical values for *R*, at all loadings.

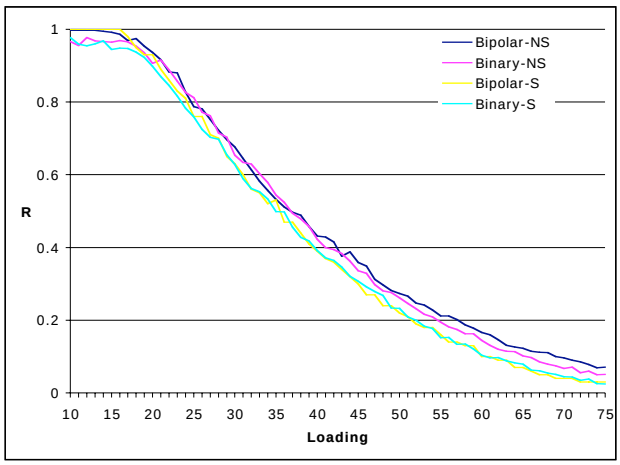

Figure 5: The *R* values for the four types of network. Four very similar plots.

Figure 6 shows, for interest, the relationship between *R* and κ for one of the network types

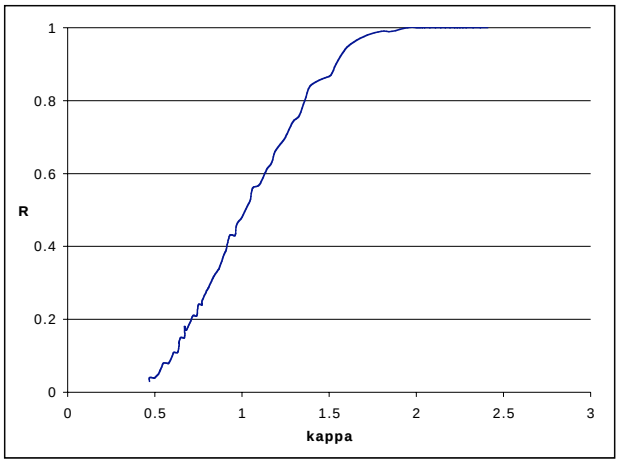

Figure 6: The value of *R* as κ varies, for the Bipolar-NC network.

For much of the range of κ the relationship is linear, showing how well κ predicts actual attractor performance.

6.4 Symmetry

For the non-symmetric networks it is interesting to examine the weight symmetry as described in Section 4.4. Figure 7 gives the results for the Binary-NS and Bipolar-NS rules.

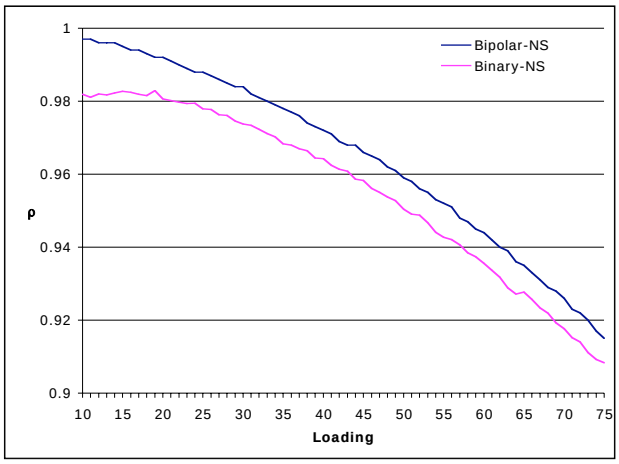

Figure 7: Symmetry values for the weight matrix of the four types of network.

As we have reported elsewhere [12] the non-symmetric learning rules produce very symmetric weights, with the symmetry decreasing with loading. It is interesting to note that the Binary network has slightly lower symmetry than the Bipolar network.

7. Discussion

As observed in Section 3, a trained bipolar network can be transformed into an exact binary equivalent, so one further question can be asked: is the binary learning rule

simply finding this scaled set of weights? To answer this question we calculated the correlation coefficient between the weights in the four types of network discussed here.

Loading	*BinNS/BipNS*	*BinS/BipS*	*BinNS/BinS*	*BipS/BipNS*
10	0.93	0.92	0.98	1.00
20	0.95	0.94	0.97	0.99
30	0.96	0.94	0.96	0.98
40	0.96	0.94	0.94	0.95

Table 2: The correlation of the weights in 50 unit networks trained at various loadings. The results are averages over 10 runs.

It can be seen in Table 2 that there is a high degree of weight correlation between the bipolar and binary net (columns 2 and 3) but they are not identical. Indeed there is higher correlation between the weights in the symmetric/non-symmetric models (columns 4 and 5).

Associative memories with binary representations are interesting as the learning rule is less biologically implausible than the corresponding bipolar rule. Here we have shown that in binary networks, trained using standard perceptron learning, perform just as well as their bipolar equivalent. Their only drawback is that the training times are significantly increased.

One further, and important point should be noted. In the bipolar networks there is a symmetry between the +1 and –1 states, so that each fundamental memory has a conjugate state, with reversed polarity, that acts as an equally significant attractor. For example if the state $\{S_i\}$ is stable, then $\forall i \bullet S_i = sign\left(\sum_j w_{ij} S_j\right)$ so that $\forall i \bullet -S_i = sign\left(\sum_j w_{ij}\left(-S_j\right)\right)$ implying that $\{-S_i\}$ is also stable. Therefore when bipolar networks are started in random initial states they will relax to fundamental memories, and their inverses, with equal frequency.

The binary networks, however, do not have this problem: +1 and 0 are not symmetric. We find that when a binary network is started in a random state, a fundamental memory is found with the same frequency as the combined frequency of finding fundamental memories and finding their inverses, in the bipolar network. The complete elimination of unwanted inverse states is a notable benefit of the binary representation.

References:

[1] B. M. Forrest, "Content-addressability and learning in neural networks," *Journal of Physics A*, vol. 21, pp. 245-255, 1988.

[2] S. Diederich and M. Opper, "Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules," *Physical Review Letters*, vol. 58, pp. 949-952, 1987.

[3] E. Gardner, "The space of interactions in neural network models," *Journal of Physics A*, vol. 21, pp. 257-270, 1988.

[4] D. J. Amit, *Modeling Brain Function: The world of attractor neural networks*. Cambridge: Cambridge University Press, 1989.

[5] J. J. Hopfield, "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*, vol. 79, pp. 2554-2558, 1982.

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

[7] G. Nardulli and G. Pasquariello, "Domains of attraction of neural networks at finite temperature," *Journal of Physics A: Mathematical and General*, vol. 24, pp. 1103, 1991.

[8] L. F. Abbott, "Learning in neural network memories," *Network: Computational Neural Systems*, vol. 1, pp. 105-122, 1990.

[9] T. B. Kepler and L. F. Abbot, "Domains of attraction in neural networks," *Journal of Physics: France*, vol. 49, pp. 1657-1662, 1988.

[10] L. Personnaz, I. Guyon, and G. Dreyfus, "Collective Computational Properties of Neural Networks: New Learning Mechanisms," *Physical Review A*, vol. 34, pp. 4217-4228, 1986.

[11] I. Kanter and H. Sompolinsky, "Associative Recall of Memory Without Errors," *Physical Review A*, vol. 35, pp. 380-392, 1987.

[12] N. Davey and R. Adams, "High Capacity Associative Memories and Connection Constraints," *Connection Science*, vol. 16, pp. 47-66, 2004.