High Capacity Associative Memories and Connection Constraints

NEIL DAVEY and ROD ADAMS

Department of Computer Science, University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK

{N.Davey, R.G.Adams}@herts.ac.uk

Abstract: High capacity associative neural networks can be built from networks of perceptrons, trained using simple perceptron training. Such networks perform much better than those trained using the standard Hopfield one shot Hebbian learning. An experimental investigation into how such networks perform when the connection weights are not free to take any value is reported. The three restrictions investigated are: a symmetry constraint, a sign constraint and a dilution constraint. The selection of these constraints is motivated by both engineering and biological considerations.

1 Introduction

In this paper we examine the performance of certain high capacity associative memory models in response to a variety of constraints that can be imposed on the values that the connections in the network can take. The networks analysed are variations of the basic Hopfield model, employing normal, deterministic dynamics. However the weight matrices are not calculated using one-shot Hebbian learning, but by other rules that produce much higher capacity. The constraints that we examine are motivated by either neuro-biological or engineering considerations. In general the weights in such networks may take on any real value and no interdependencies on the weights are imposed. The standard one-shot Hopfield learning rule produces weights that are symmetric, a property that is useful in producing simple dynamic behaviour. However other learning rules for these networks do not produce symmetric weights, but may be forced to so do. Section 3 examines the consequences of this constraint. The second issue to be examined is the extent to which these networks can tolerate constraints on the sign that the weights may take. The primary motivation for this is that in real neural systems synapses do not normally change from being excitatory to inhibitory, or vice-versa, so that in a simple interpretation, the sign of a weight is not free to change during training. Since biological networks of neurons are not fully connected it is interesting to examine the degree to which the modelled networks can tolerate removal of connections – dilution – prior to training, and this constitutes the final set of experiments reported here.

The next section describes the architecture of the neural networks that underlie this investigation. Section 3 gives the learning rules used to train the networks and Section 4

discusses the types of constraint to the weights that are investigated. Results and conclusions follow in the last two Sections.

2 Models Examined

We consider networks of N units which we train with a set of N–ary, bipolar (+1/-1) training vectors, $\{\boldsymbol{\xi}^p\}$. The N by N weight matrix is denoted by $\mathbf{W}$, and the state (output) of the i'th unit is denoted by S_i

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 *next* state, S_i', of a unit is derived from its local field and its *current* state:

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

where the threshold, θ_i, is normally taken as zero. Unit states may be updated synchronously or asynchronously. Here we use asynchronous, random order updates. These network dynamics and a *symmetric* weight matrix guarantee simple point attractors in the network's state space.

A training vector, $\boldsymbol{\xi}$, will be a stable state of the network if the *aligned local fields*, $h_i \xi_i$ are non-negative for all i (assuming all θ_i are zero). Each training vector that is a stable state is known as a *fundamental memory* of the trained network. The *capacity* of a network is the maximum number of fundamental memories it can store. The *loading*, α, on a network is the ratio of the number of vectors in the training set to the number of units in the network, N.

3 Learning Rules

In the late 1980s it was demonstrated that perceptron like learning could be used in associative memory networks, giving much higher capacity than the basic model. In fact, as Gardner (Gardner, 1988) showed, a Hopfield type network of N units may store up to $2N$ uncorrelated patterns (a loading, α, of 2), with this figure increasing for correlated patterns. Learning rules of this type are designed to drive the *aligned local fields* of patterns in the training set over a threshold value, T.

The training patterns will be stable if T is non-negative (see Section 2) and, for ease of training, a value of 1 (or even 0) may be taken. However, by raising T the attractor performance of the network may be improved (Krauth, and Mezard, 1987). Some care must be taken though. Consider a network in which all training patterns are stable ($h_i \xi_i \geq T$ for all patterns and units): any uniform, upward scaling of the weight matrix will increase the aligned local fields, but will obviously not improve the attractor performance. Optimal attractor performance is achieved when the threshold is maximised

with respect to the size of the weights, so the relevant characterization is the *normalised stability measure*, defined as:

$$\gamma_i = \frac{h_i \xi_i}{|W_i|}$$

where W_i is the incoming weight vector to unit i. The minimum of all the γ_i therefore gives a measure of the likely attractor performance (Kepler, and Abbot, 1988) and we take:

$\kappa = \min_{p,i}(\gamma_i^p)$. The largest possible value that κ can take, $\kappa_{\max}$ is determined by the loading on the network – the higher the loading the lower the value of $\kappa_{\max}$ (see Figure 4). This corresponds with the intuition that good attractor performance is likely to decrease with increasing loading.

3.1 Local Learning (LL)

Diederich and Opper's (Diederich, and Opper, 1987) local learning rule is an iterative learning rule in which the local fields for each training pattern are driven to the correct side of +T or –T as appropriate. This is equivalent to the condition that:

$$\forall i, p \bullet h_i^p \xi_i^p \geq T$$

So the learning rule is given by:

Begin with a zero weight matrix
Repeat until all local fields are correct
Set the 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
on connections into unit i according to:

$$\Delta w_{ij} = \frac{\xi_i^p \xi_j^p}{N}$$

This is the perceptron learning rule with a fixed margin of T and a learning rate of $\frac{1}{N}$. The process will converge on a suitable weight matrix if one exists (Diederich *et al.*, 1987) , at which point the trained patterns are guaranteed to be stable. We refer to this as the LL (local learning) rule. As shown by Abbott (Abbott, 1990), this rule leads to a network in which

$$\frac{T}{2T+1}\kappa_{\max} \leq \kappa \leq \kappa_{\max}$$

where $\kappa_{\max}$ is the optimal value of κ as described earlier. From this it is apparent that increasing T will in turn increase the lower bound of κ, and this may give better attractor

performance. However the limiting value of this lower bound, as $T \to \infty$, is $\frac{\kappa_{\max}}{2}$ so that increasing T does not necessarily force the network to optimal performance.

3.2 Krauth and Mezard Local Learning (KM)

A modification to the local learning rule, proposed by Krauth and Mezard (Krauth *et al.*, 1987) can be shown to produce a κ value that does tend towards $\kappa_{\max}$ as T increases. In this version the patterns are not presented to the network in an arbitrary order. Instead the pattern that has the smallest aligned local field is chosen as the one for next presentation:

Begin with a zero weight matrix

Repeat until all local fields are correct

For each unit, i, in turn

Select the pattern, ξ^p with lowest aligned local field at this unit and update the incoming weights according to:

$$\Delta w_{ij} = \frac{\xi_i^p \xi_j^p}{N}$$

Krauth and Mezard (Krauth *et al.*, 1987) prove that, with this rule, $\kappa \to \kappa_{\max}$ as $T \to \infty$.

4 Constraints

The weights in a neural network are constrained, primarily, by the task that the network is required to undertake: in the case of an associative memory, that is to store patterns. However in order to examine the consequences of neurological or engineering factors other constraints may be imposed. The specific constraints on the weights in the network that we examine are described here.

4.1 Symmetry

The original Hopfield network has a symmetric weight matrix and such weights have the desirable property of guaranteeing point atttractors, with asynchronous updating and cycles of at most length 2 with synchronous updates. As the symmetry is broken, more complex dynamics become progressively more likely. On the other hand Krauth, Nadal and Mezard (Krauth, Nadal, and Mezard, 1988) showed that, under certain circumstances, decreasing the symmetry of the weight matrix should improve attractor performance. Moreover a network with symmetric weights has only half the number of degrees of freedom, so it is surprising that according to Nardulli and Pasquariello (Nardulli, and Pasquariello, 1991) the storage capacity of a fully symmetric network is theoretically the same as an asymmetric one. In (Gardner, Gutfreund, and Yekutieli, 1989) numerical simulations suggest that at low loading there is a range of weight matrices with varying symmetry that will embed the training patterns, but that as the loading increases towards

saturation the degree of symmetry tends towards a specific, high, value. The practical implications of this are one of the issues investigated here.

Learning rules based on the perceptron training rule are not guaranteed to produce symmetric weights, and in fact will produce weight matrices that are progressively less symmetric as the loading increases. Nevertheless Gardner (Gardner, 1988) pointed out that an iterative perceptron like training rule could be made to produce symmetric weights, by simply updating both w_{ij} and w_{ji}, when either changes. She also showed that such algorithms would find a symmetric weight matrix, if one existed, for a particular training set. To investigate the implications of having a symmetry constraint we compare asymmetric and symmetric versions of both the Diederich and Opper local learning method (LL) and the Krauth and Mezard optimal version (KM).

The *Symmetric Local Learning rule* (SLL) is therefore:

Begin with a zero weight matrix

Repeat until all local fields are correct

Set the state of network to one of the $\boldsymbol{\xi}^p$

For each unit, i, in turn

Calculate $h_i^p \xi_i^p$.

If this is less than T then change the weights between unit i and all other units, j, according to:

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

The KM rule can be treated in the same way and we denote the symmetric version as SKM.

4.2 Sign Constraints

A possible difficulty with the normal perceptron learning rule is that weights can (and do) change sign during the learning process. The biological equivalent of this would be for a synapse to change from excitatory to inhibitory or vice versa. This is not thought to happen, and indeed Dale's rule (Dale, 1935) states that all the efferent synapses from a given neuron are all either excitatory or inhibitory. For a neural network this is equivalent to requiring that all outgoing weights from a given unit have the same sign, and this cannot change over time. There are now known to be exceptions to this picture, so that, for example, the sign of the synapse may be determined by properties of the post-synaptic cell (Amit, Wong, and Campbell, 1989b; Wong, and Campbell, 1992).

A general sign constraint mechanism therefore consists of a matrix of signs, $g_{ij} = \pm 1$, corresponding to each weight in the network, together with requirement that: $g_{ij} w_{ij} > 0$.

The *sign-bias* of the weights is the ratio of positive to negative weights.

4.2.1 Capacity

The effect of imposing a sign constraint to every connection in a standard Hopfield network was first investigated in 1986 (Sompolinsky, 1986) where it was shown that the capacity only falls from $\alpha = 0.14$ to $\alpha = 0.09$, for uncorrelated patterns. Later Amit et al. (Amit *et al.*, 1989b) showed that the perceptron learning rule could also be effective under such a constraint. They also showed (Amit, Campbell, and Wong, 1989a) that the theoretical maximum capacity of a sign constrained network was exactly half that of the unconstrained version (a simpler argument showing this is given in Campbell and Robinson (Campbell, and Robinson, 1991)), namely $\alpha = 1.0$ for signed nets and $\alpha = 2.0$ for unconstrained nets. This is a surprising result as the volume of weight space that the network may use is reduced by a much higher proportion. They also showed that this capacity (for unbiased patterns) is independent of the specific sign constraint used. In particular, a network of units using only excitatory (or inhibitory) connections could store up to N uncorrelated patterns. The argument to demonstrate this is straightforward: Suppose a set of random set of patterns is learnable with a particular sign constraint. Then if weight w_{ij} is flipped, the stability of the stored patterns can be restored by flipping bit j in each of these patterns. So that an equal number of different, but still random patterns can be learnt by the network with the new sign constraint. However the presence of correlated training data will make the capacity of network sensitive to the specific *sign-bias*. Viswanathan (Viswanathan, 1993) studied networks which strictly adhered to Dales rule, so that all the outgoing weights at a given neuron had the same sign, $\forall i, i' \quad g_{ij} = g_{i'j}$. The results showed that the theoretical capacity of such networks was always greatest when the number of excitatory and inhibitory neurons was equal, $\langle g_{ij} \rangle = 0$. Moreover when the training data becomes increasingly correlated the theoretical capacity increases, so that with the optimal sign constraint ($\langle g_{ij} \rangle = 0$) the initial capacity for unbiased data of N would increase as the correlation increased.

4.2.2 Dynamics

The dynamics of the network are affected by the sign bias. Wong and Campbell (Wong *et al.*, 1992) showed that in a diluted network, with any sign constraint that had a non-zero bias of positive or negative weights, developed a new form of attractor: the uniform state (all +1/-1). As the sign-bias increases then the uniform state becomes progressively more likely to attract other states. It is likely that this behaviour would extend to fully connected networks, since for example, in a network with positive weights only, the energy function $E\{\boldsymbol{S}\} = -\frac{1}{2}\sum_{i,j} w_{ij} S_i S_j$, will have a *global* minimum at the uniform, +1, state. A consequence of the increasing influence of the uniform attractor could be to decrease the attractor basin size of the stored patterns.

4.2.3 Learning Rules

Amit et al (Amit *et al.*, 1989b) suggest how a learning rule based on standard perceptron learning can be modified to comply with a particular sign constraint. The idea is straightforward: whenever a weight change is proposed that will result in a violation of the sign constraint, the change is not made. A variant of this is to zero such a violating

weight. Specifically, given a particular sign-bias, $g_{ij} = \pm 1$, and an initialisation of zero weights the Signed version of LL, *Signed-LL*, can be formally stated as:

Repeat until all local fields are correct

Set the state of network to one of the $\boldsymbol{\xi}^p$

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:

$$w'_{ij} = w_{ij} + \frac{\xi_i^p \xi_j^p}{N}$$

whenever the resulting weight meets the sign constraint, $g_{ij} w'_{ij} > 0$, *otherwise leave the weight unchanged*

The variant of this, mentioned above, is to use

$$w'_{ij} = \max\left(g_{ij} \left(w_{ij} + \frac{\xi_i^p \xi_j^p}{N} \right), 0 \right)$$

and we will denote this variant as Signed-LL-Zero

Note that this form of learning can be used in any variant of perceptron learning, so that signed KM is straightforwardly derived from the KM algorithm.

Of course symmetry can also be maintained for signed networks, by requiring that the sign constraints are symmetric, $g_{ij} = g_{ji}$ and using SLL modified to adhere to the sign constraint, as above. This learning rule is denoted as Signed-SLL.

As is well known, normal perceptron learning will converge on a solution, if one exists, since the weight changes always move the weight vectors towards ones that embed the training vectors (Hertz, Krogh, and Palmer, 1991). With the sign constrained version it is also possible to show (Amit *et al*., 1989b) a similar result. Providing a solution satisfying the sign constraint exists, then any weight change given by the Signed-LL rule will move the weights nearer to the desired solution.

4.3 Dilution

The weights in a network can be diluted (removed) either before or after training takes place. For any one-shot rule, where a single weight is immediately determined by the training patterns, without referral to the connectivity of the network, the two approaches are obviously equivalent, and it is known (Sompolinsky, 1986) that capacity drops linearily with the proportion of weights removed.

In the scheme adopted here a fraction of the weights of the network are set to a constant value of zero (effectively removed from playing any part in the network dynamics). This

may be done in such a fashion that the symmetry of the connection matrix is maintained, that is if w_{ij} is removed then so is w_{ji}, or alternatively in a completely random way. We use both approaches. If symmetry is maintained in dilution, subsequent training uses symmetric local learning, otherwise normal perceptron style learning is used. The dilution rate, *d*, is the proportion of weights that are removed prior to training.

5 Analysis Tools

5.1 Introduction

The experiments described in the next section are designed to give empirical information about the performance of the networks under the constraints described in Section 4. To this end we use several measures of performance: the training time and κ values at specific loadings and learning thresholds are reported. Where appropriate the degree of symmetry in the networks weight matrix is also reported as described in 6.1.1. The training sets are all randomly generated and by default have no bias towards +1 or –1. However on occasion we are interested in the response of the network to training sets that are biased: the bias of a training set is the probability that any bit will be +1.

The most interesting performance measure is the ability of the network to act as a pattern completion/corrector. This is difficult to ascertain and our approach to measuring this is described next.

5.2 Attractor Basin Size

An effective associative memory model is expected, not only to have the training patterns as fixed points of the network dynamics, but also that these fixed points should act as attractors in the state space. The ideal behaviour of such an associative memory would be such that a given initial state should relax to the nearest trained pattern. It is therefore important to know the mean size of the basins of attraction of the trained patterns.

Since the attractor basins cannot be expected to be Hamming hyperspherical (Storkey, and Valabregue, 1999), it is usual to take the minimum Hamming radius:

$$R\left(\boldsymbol{\xi}^p\right) = \inf\left\{\left|\mathbf{q} - \boldsymbol{\xi}^p\right| : \mathbf{q} \in Basin\left(\boldsymbol{\xi}^p\right)\right\}$$

The mean radius of attraction over the patterns, *R*, can act as a measure of the quality of a particular associative memory. It is also common for *R* to be normalised with respect to the size of the network, so that it lies between zero and one.

For very small networks it is possible to exhaustively explore the state space (see, for example Personnaz, Guyon, and Dreyfus, 1986), in order to calculate *R* exactly, but for more realistic sizes the nature of the attractors is very hard to compute (Floréan, and Orponen, 1993; Kepler, and Abbott, 1988) and only empirical methods are available.

A sample of states at a fixed distance, *r*, from a trained pattern, $\boldsymbol{\xi}^p$, is made, and if all of them relax to $\boldsymbol{\xi}^p$, it is concluded that $R(\boldsymbol{\xi}^p)$ is at least as big as *r*. Clearly, the larger the sample size the higher the quality of the estimate, in all of our experiments the sample

size is 50. An analysis of the affect of sample size on the estimate of R can be found in (Davey, and Hunt, 2000b).

In our implementation we have slightly adapted the method of Kanter and Sompolinsky (Kanter, and Sompolinsky, 1987) in the calculation of R. For each of the sample states chosen a fixed fraction, m_0, 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. 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 ξ^p . Averaging m_0 over different stored patterns yields:

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

As is pointed out in (Kanter *et al*., 1987), for finite size associative memories, another factor needs to be considered. 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:

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

where m_1 is the largest overlap with the rest of the stored patterns. This is a double average over both different sets of stored patterns and different sample states.

So in our implementation, a fixed number of random starting points are chosen, each of which has a low overlap with the members of the training set (low average m_0). If, as is likely, the start state does not relax to the closest training pattern in one or more of the random cases, the value of m_0 is increased (by $1/N$), and the search is repeated. This continues until all random start states relax to the closest stored pattern. This procedure is performed for six different sets of stored patterns for each network type.

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.

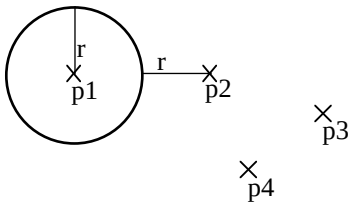

Figure 1 Calculating R. In this figure p1, p2, p3 and p4 are patterns. The closest pattern in the training set to p1 is p2, at a distance of 2r. Optimal performance occurs when all vectors within the hypersphere centred on p1 and radius r, are attracted to p1. If all patterns stored in a network exhibit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

6 Results

6.1 Symmetry of the Weights

For both symmetric and non-symmetric versions of the networks studied here the theoretical capacity is known to be 2N for unbiased, random patterns and higher for biased ones. Moreover both the basic learning rules described earlier will find an appropriate weight matrix if one exists. So it is sensible to compare the symmetric and non-symmetric networks in terms of their attractor performance and convergence time of the learning rule, but not for maximum capacity. In general, as would be expected, the size of the attractor basins decreases as loading increases, as can be seen in Figure 2.

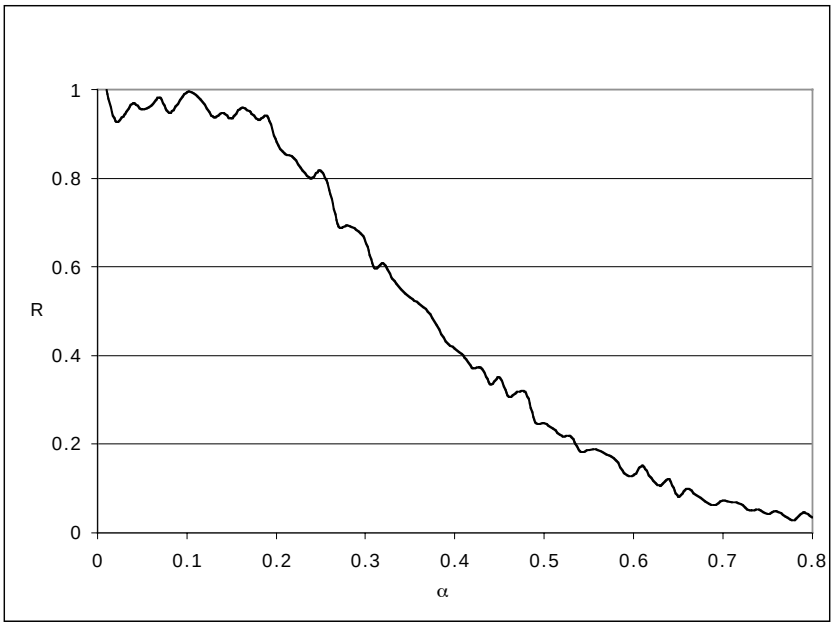

Figure 2: The basin of attraction size for the Symmetric Local Learning Network, with 100 units and unbiased patterns. Results are averages over 10 networks at intervals of 0.01 in loading, α. Graphs for the other networks show a similar pattern.

The next set of results, shown in Table 1, compares Local Learning (*LL*) with the symmetric version, *SLL*. In all cases the loading of the (100 node) network is $\alpha = 0.3$, the patterns are unbiased and the results are averages over fifty runs. At this loading the theoretical value of κ_{max} is 1.27.

	T	κ	R	Training Epochs
LL	1	0.84	0.57	7.7
LL	10	1.14	0.64	54.8
LL	100	1.18	0.63	500.6
SLL	1	0.80	0.54	11.6
SLL	10	1.14	0.65	35.6
SLL	100	1.18	0.65	307.8

Table 1: The comparative performance of local learning and its symmetric counterpart, under a loading of 0.3 (30 patterns in a 100 node network). The patterns are unbiased and the results averages over 50 runs.

It can be seen that the imposition of symmetry does not affect the attractor performance (R) of the network. Moreover the increase in T raises the value of κ but, interestingly, this does not improve attractor performance, in the change from T = 10 to T = 100, in either case. The actual value of κ obtained is much higher than the theoretical lower bound, which for this learning rule, at this loading is: 0.42 for T = 1, 0.60 for T = 10 and 0.63 for T = 100.

The training time (epoch count) is increasing linearily with T, which is in accordance with the theoretical upper bound on training time (Krauth *et al*., 1987). However it is apparent that the convergence of SLL, at the higher values of T, is significantly faster than the non-symmetric version.

The results for the Krauth and Mezard rule, shown in Table 2, again with $\alpha = 0.3$, unbiased patterns and the results averaged over fifty runs show a similar pattern to LL. The imposition of symmetry does not make much difference to R, with KM being marginally better than SKM. A comparison of Tables 1 and 2 shows the R values for LL to be similar to those for KM, although as in accordance with the theoretical result, the κ values are higher for KM, getting close to the theoretical maximum (1.27) with the highest threshold. The results do not contain the training epoch count as the algorithm does not take place in a simple epoch by epoch fashion.

	T	κ	R
KM	1	0.87	0.57
KM	10	1.19	0.66
KM	100	1.23	0.64
SKM	1	0.87	0.56
SKM	10	1.19	0.61
SKM	100	1.23	0.62

Table 2: The comparative performance of Krauth / Mezard local learning and its symmetric counterpart, under a loading of 0.3 (30 patterns in a 100 node network). The patterns are unbiased and the results averages over 50 runs.

6.1.1 Symmetry

It is interesting to look at the degree of symmetry in the weight matrices produced by the asymmetric versions of the learning rules. To this end the symmetry measure of Krauth, Nadal and Mezard (Krauth *et al.*, 1988) was applied to the resulting weight matrices. It is defined as:

$$\sigma = \frac{\sum_{i,j} w_{ij} w_{ji}}{\sum_{i,j} w_{ij}^2}.$$

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. The results, in Table 3, show that the weight matrices produced for all thresholds are highly symmetric with the symmetry increasing with the threshold.

T	σ - LL	σ - KM
1	0.961	0.968
10	0.983	0.991
100	0.983	0.991

Table 3: Symmetry of LL and KM with alpha = 0.3, and unbiased patterns. Averages over 50 runs

6.2 Sign Constraints

6.2.1 Capacity

The first set of results measures the capacity of signed networks trained using Signed-LL, varying both the bias of the training sets, and the weight sign-bias. The actual capacity can only be estimated; an incremental search was undertaken for the first point at which

the network failed to learn five different sets of random patterns. The highest loading for which this was possible was taken as the capacity of the network.

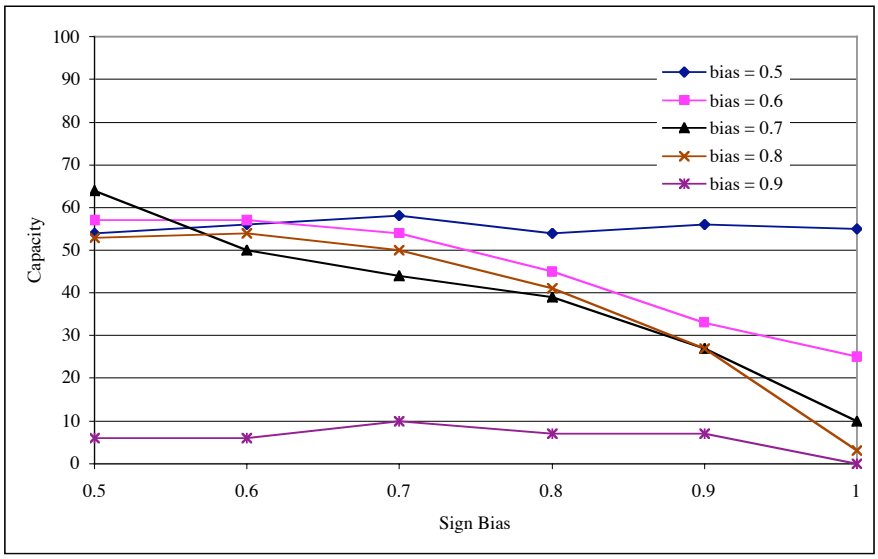

Figure 3: Capacity of 100 unit networks, trained using Signed-LL, with varying degrees of Sign Bias and with different correlations within the training sets (data-bias 0.5 to 0.9).

In Figure 3 it can be seen that when the patterns are not correlated (data-bias = 0.5) the capacity is independent of the specific sign bias, as expected. However this capacity is significantly less than the theoretically predicted one of 100 patterns in a 100 unit network. As the training sets become more correlated, an increasing sign bias causes the capacity to fall considerably. This is in accord with the theoretical prediction of Viswanathan (Viswanathan, 1993) for the special case of networks that adhere to Dale's law. The exception is with highly correlated patterns (data-bias = 0.9) where capacity is very low whatever the sign bias. It is also noteworthy that the networks can withstand some bias in the signs: with these networks capacity was maintained reasonably up to a sign bias of 0.8.

The second of Viswanathan's theoretical predictions, that increasing correlation should increase capacity is however, not confirmed in the general set of sign biases studied here.

6.2.2 Basins of Attraction and Symmetry of Weights

In these experiments the mean normalised radii of the basins of attraction, R, associated with fundamental memories is estimated. The minimum of the normalised stability factors, κ, and the symmetry of the weights σ, is also reported. All three sets of results are with 15 random patterns in 100 unit networks, with T = 10, and results averaged over 50 runs. This loading is chosen as, in most cases, it is well within the capacity of the networks.

Uncorrelated Data

Considering first the uncorrelated data, Table 4, where it can be seen that the signed networks show progressively poorer performance (R values) as the sign of the weights becomes more correlated. This confirms the increasing importance of the uniform attractor, as the sign of the weights become similar (see Section 3.2). However for each sign-bias the κ values of each type of network are very similar so that the normal relation between R and κ is broken; for this type of network a very unusual result.

It is also interesting to note that the non-symmetric version of the signed nets, Signed-LL, performs better than the symmetric version, Signed-SLL. Normally the symmetric weight models are preferred, as they have simpler dynamics (Davey, Adams, and Hunt, 2000a), and it is particularly unusual that networks with a relatively low degree of symmetry (σ = 0.41), as in the case of the 0.50 sign-bias version of Signed-LL should perform so well.

As the sign bias increases the weights become progressively more symmetric, so that at a Sign-Bias of 1.00 the weights are very nearly symmetric. σ = 0.95. This is not unexpected: as the sign bias of the weights increases the more likely it is that two weight pairs, w_{ij} and w_{ji}, will have the same sign and can therefore take similar values.

For comparison the unrestricted learning rule SLL is also included and it can be seen that it attains a κ value roughly twice that of the signed networks. This is in accord with the theoretical prediction – as is shown in Figure 4 for any given kappa-max the maximum theoretical capacity of a signed net is half that of its unsigned counterpart (and vice versa)

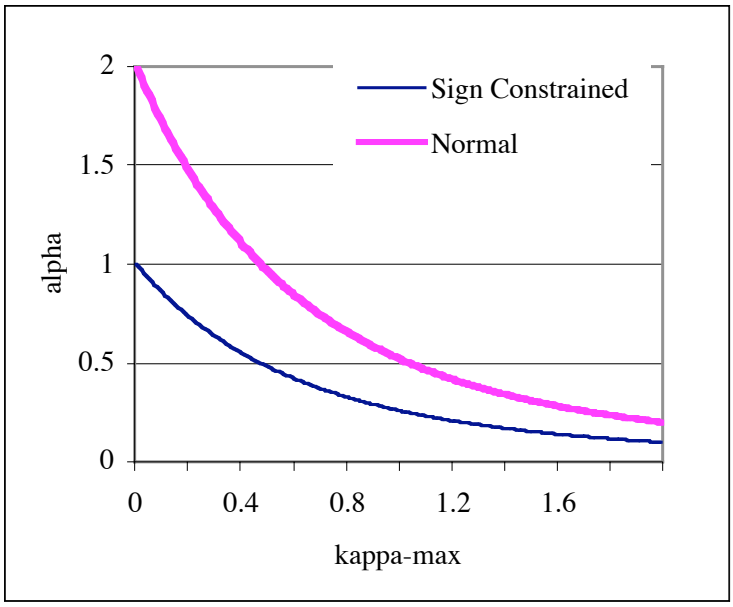

Figure 4: Theoretical relationship between maximum capacity, α, and the maximum possible value of κ, κ-max for unbiased random data.

Network	Sign-Bias	R	κ	σ
Signed-LL	0.50	0.78	0.99	0.41
	0.75	0.52	0.98	0.54
	1.00	0.23	1.00	0.95
Signed-SLL	0.50	0.65	0.95	1.00
	0.75	0.39	0.95	1.00
	1.00	0.20	0.94	1.00
SLL	-	0.96	1.84	1.00

Table 4: Uncorrelated Data (bias 0.5). Attractor Performance, R, κ and σ for three different types of network. Each result is for 100 unit networks trained with 15 patterns averaged over 50 runs. For this loading the masimum theoretical value of κ is more than 2.0 for the unsigned network.

Correlated Data

Tables 5 and 6 give similar results when the data is correlated. The overall pattern of results, for R and σ is as for the uncorrelated data. However the κ results show decreasing κ as the sign-bias increases, contributing to the resulting poor attractor performance. Normally in these networks increasing correlation in the training set should improve performance, and this is confirmed here in the slightly higher R values, when comparing Tables 4, 5 and 6.

Network	Sign-Bias	R	κ	σ
Signed-LL	0.50	0.82	0.98	0.41
	0.75	0.60	0.91	0.51
	1.00	0.12	0.64	0.88
Signed-SLL	0.50	0.70	0.95	1.00
	0.75	0.46	0.88	1.00
	1.00	0.07	0.56	1.00
SLL	-	0.99	1.85	1.00

Table 5: Correlated Data (bias 0.6). Attractor Performance, R, κ and σ for three different types of network. Each result is for 100 unit networks trained with 15 patterns averaged over 50 runs.

Network	Sign-Bias	R	κ	σ
Signed-LL	0.50	0.97	0.87	0.40
	0.75	0.92	0.83	0.49
	1.00	-	-	-
Signed-SLL	0.50	0.85	0.83	1.00
	0.75	0.83	0.80	1.00
	1.00	-	-	-
SLL	-	1.00	1.60	1.00

Table 6: Correlated Data (bias 0.8). Attractor Performance, R, κ and σ for three different types of network. Each result is for 100 unit networks trained with 15 patterns averaged over 50 runs. Results for a sign-bias of 1.00 are not reported as these networks fail to learn at this loading and pattern bias.

6.3 Dilution

In these experiments networks are either diluted asymmetrically and trained using LL, or are diluted symmetrically and trained using SLL.

6.3.1 Capacity

As described in the previous section, to find the capacity of the diluted networks we search for the point at which the learning rule fails to converge, when presented with ten sets of patterns at the given loading. We investigate 100 unit networks with dilution rates varying from 0 to 0.9 in increments of 0.1. The symmetric learning rule, SLL, is used here, with T=1. All training sets are unbiased (b = 0.5).

The results (Figure 5) show a similar pattern to that reported for one-shot Hebbian learning (Sompolinsky, 1986): a roughly linear decrease in capacity with increasing dilution. Interestingly the capacity of this form of network with 80% of the connections removed is roughly equivalent to a fully connected standard Hopfield network. Note also that for all dilutions up to 0.6, at least 30 patterns are learnable by the network.

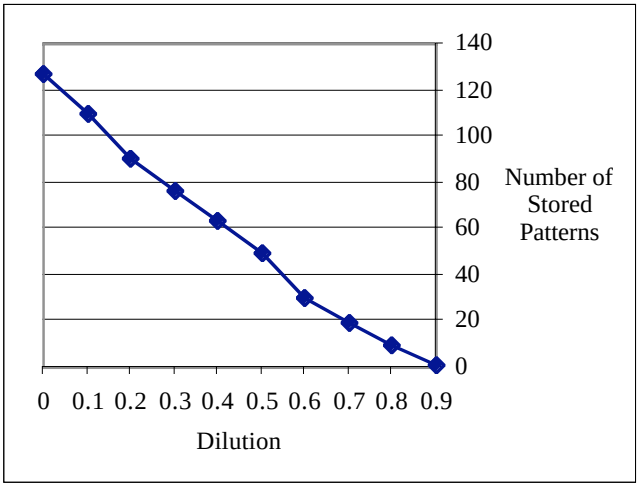

Figure 5: Capacity of diluted networks (N = 100) trained with the SLL rule.

6.3.2 Effect of Varying Training Threshold

In this section we present the results of varying the learning threshold, T, for networks trained using both the LL and SLL rules. In all cases dilution of 0.4 is considered, since networks with this many connections have a capacity well in excess of the 30 patterns we wish to store, as shown in Figure 4.

Attractor Performance

Table 7 shows how the attractor performance changes for the network with d = 0.4, as the learning threshold is increased. As a base case the undiluted version of the network is also given. Consider first the results for the non-symmetric networks (LL). It is immediately apparent that the effect of dilution is to lower the κ value and correspondingly lower the R values. Increasing the learning threshold from 1 to 10 does improve the R value slightly, but the R value does not approach that of the undiluted network. Increasing the learning threshold further (from 10 to 100) does not appear to bring benefit.

The symmetric networks (SLL) show a similar pattern. However the most significant result here is that the attractor performance of the symmetrically diluted and trained networks is markedly inferior to the non-symmetric versions.

Network	T	κ	R
LL (d = 0)	1	0.83	0.56
LL (d = 0.4)	1	0.55	0.23
LL (d = 0.4)	10	0.68	0.26
LL (d = 0.4)	100	0.67	0.23
SLL (d = 0)	1	0.80	0.55
SLL (d = 0.4)	1	0.53	0.10
SLL (d = 0.4)	10	0.62	0.11
SLL (d = 0.4)	100	0.63	0.11

Table 7: Attractor performance of diluted networks, under a loading of 0.3 (N = 100). Training sets are unbiased (b = 0.5) and results are averages over 50 runs.

Training Times

Table 8 shows how the training time varies as T is increased. Again the undiluted networks are shown for comparison. The symmetric and non-symmetric versions take a similar number of epochs to train. With the threshold, T, at 1, the effect of dilution is to significantly increase the training time, when compared with the undiluted networks. Moreover, as T is increased the training time increases in a roughly linear way. This pattern is also seen in undiluted networks (Krauth *et al*., 1987).

Network	T	Epochs
LL	1	10.32
LL (d = 0.4)	1	27.63
LL (d = 0.4)	10	184.47
LL (d = 0.4)	100	1941.84
SLL	1	8.26
SLL (d = 0.4)	1	27.11
SLL (d = 0.4)	10	195.53
SLL (d = 0.4)	100	1881.84

Table 8: Training times for diluted networks under a loading of 0.3 (N = 100). Training sets are unbiased and results are averages over 50 runs.

6.3.3 Symmetry

Next the symmetry of the asymmetric networks is examined. If a network is randomly diluted at a rate of d = 0.4, but the remaining weights are symmetric wherever possible,

we would expect σ (our measure of symmetry) to be roughly 0.6, which is therefore the maximum value that σ could be expected to achieve. As can be seen from Table 9, the weight matrix for the undiluted network is very nearly symmetric, but σ is significantly less than 0.6 for each of the diluted networks. The inference we draw from this is that the learning rule is introducing greater asymmetry to the weight matrix in order to cope with the asymmetric dilution - compare with the 0.96 in the last row which is very close to the maximum value it could be of 1.0 for the undiluted network. This degree of asymmetry is likely to be a problem for these types of networks, as symmetry is necessary for prohibiting non-point attractors. So when the heavily diluted LL networks are run they often reach multi-point orbits, which are difficult to identify.

Dilution	**T**	**σ**
0.4	1	0.49
0.4	10	0.49
0.4	100	0.48
0.0	1	0.96

Table 9: Symmetry of weight matrices in networks trained with LL. Averages over 50 runs.

6.3.4 Effect of Varying Dilution and Bias

In this section we examine how the attractor performance and training times change as the dilution rate is varied. The SLL rule is used here due to the difficulty of measuring R for highly diluted networks trained with the LL rule, in which the dynamics are increasingly complicated (see above). Training sets which are unbiased (b = 0.5) and correlated (b = 0.7) are used.

Attractor Performance

In Figure 6 it can be seen that the R values decrease with increasing dilution and that the networks perform better with correlated patterns, regardless of the amount of dilution. Once again, this also holds for undiluted networks (Davey *et al*., 2000b).

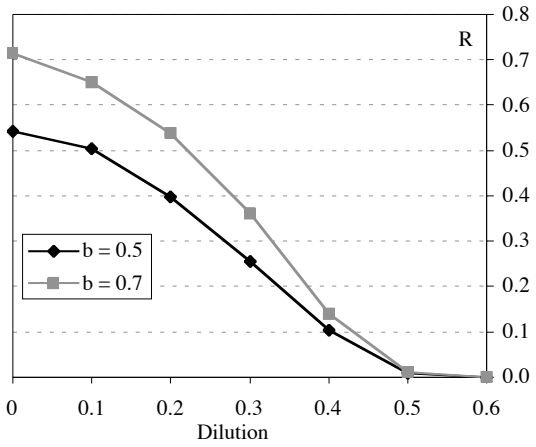

Figure 6: Attractor performance of networks trained using SLL, under a loading of 0.3 (N = 100) and varying dilution. Patterns are either unbiased (b = 0.5) or correlated (b = 0.7). Results are averages over 50 runs.

Training Times

Finally, in Figure 7, the effect of increasing dilution on training times is given. Increasing dilution increases the training time, the bias of the training patterns does not have a significant effect.

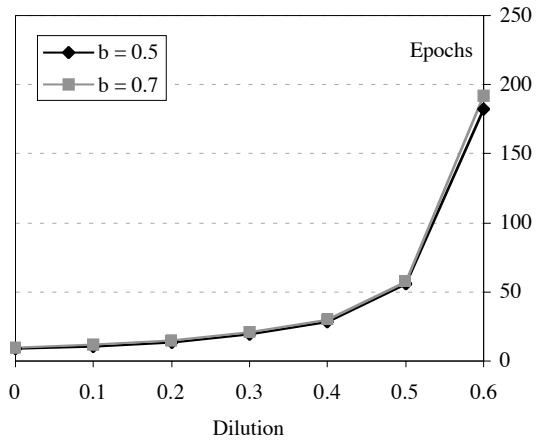

Figure 7: Training times for networks trained using SLL, under a loading of 0.3 (N = 100) and varying dilution. Patterns are either unbiased (b = 0.5) or correlated (b = 0.7). Results are averages over 50 runs.

7 Conclusion

This paper has discussed how three types of constraint on the weights in a network of perceptrons affects the performance of the network as an associative memory. Considering each constraint in turn:

Symmetry

Symmetry of the weights in an associative neural network is a mixed blessing. Desirable from the perspective of dynamics, but with potentially damaging implications for the attractor performance. However as shown above, for both forms of learning rule, the addition of a symmetry constraint did not have an adverse affect on the attractor basins. The reason for this is probably that the matrices that result from unconstrained learning are already highly symmetric and become more so with a larger learning threshold, which is itself an interesting result.

It is also apparent that, for the LL rule, imposing a symmetry constraint during learning had a helpful affect on convergence – almost halving the training epochs required. In symmetric local learning, for each epoch, each weight is changed twice, so if both these changes are constructive in moving the weight towards its final value, the learning may, at best, be twice as efficient. If the LL weight matrix was actually strongly asymmetric, then it is improbable that the SLL double weight change would be constructive, and the observed halving of training epochs would not have occurred.

The best versions of these fully connected. high capacity Hopfield networks are those with strictly symmetric weights, since they have simple dynamics (only point attractors in the phase space), and learn faster.

Sign Constraints

Complete freedom in assigning weights to connections may not be an adequate model of biological systems, where amongst other constraints, connections may be only excitatory or only inhibitory. An investigation into how the proportion of signed to unsigned weights in a network, its sign-bias, affects the behaviour of the network was undertaken.

One of the important results here is that the actual capacity of a sign constrained network is a lot less than the theoretical maximum. The presence of correlation in the training data decreased the capacity, contrary to both the behaviour of unsigned nets and the theoretical prediction of Viswanathan (Viswanathan, 1993). The degree of correlation in the signs of the weights was shown to affect the dynamics of the trained networks, so that the best attractor performance (R values) was attained with neutral sign correlation, where the uniform attractor was not significant. It was also observed that with sign constrained networks, the normal static measure of likely performance, the smallest normalised stability measure, was not a good predictor of performance. The specific sign bias of these networks is important in attaining good performance and it suggests that in biological systems the ratio of excitatory to inhibitory synapses may not be accidental.

Dilution

Dilution of the weights in a high capacity associative neural network is interesting from both the neurophysiological perspective and from an engineering point of view, in which the number of connections can be viewed as a resource to be minimised. There are at least

two ways in which pre training dilution can be undertaken in such networks, either maintaining symmetry or not. In the latter case the asymmetry of the remaining weights causes problems with the network dynamics, as discussed in section 4.2.3.

The capacity of the SLL networks is shown to decrease linearily with the rate of dilution, a similar pattern to that of networks trained with one-shot Hebbian learning. However the SLL network maintains a relatively high capacity for dilution rates up to 80%. The attractor performance of the diluted networks is poorer than the undiluted counterparts, and although increasing the learning threshold does improve performance it is not possible to recover to the level attained by fully connected networks, and training times are significantly increased. The presence of correlated training patterns is not a problem for these networks, indeed the attractor performance is actually better for biased patterns, as shown in section 6.3.4.

Importantly it is shown that symmetrically diluted networks do not perform as well, in terms of attractor performance, as their asymmetric cousins. An interesting question that it has not been possible to explore here is whether a symmetric dilution policy together with the asymmetric learning rule would bring benefit. The low symmetry of the asymmetrically diluted LL networks (Table 9) suggests that this is a possibility worthy of exploration.

In overall conclusion it can be seen that the sign-constrained networks perform reasonably as associative memories, but are weaker than the networks without this constraint. The fully connected, symmetrically trained networks, SLL, give the best performance. However the interaction between symmetry and dilution shows that this conclusion is not necessarily appropriate for diluted networks. Some preliminary work on networks with small-world connectivity patterns (Bohland, and Minai 2001; Watts, and Strogatz, 1998) suggests that asymmetric dilution gives better attractor performance than symmetric dilution.

References

Abbott, L. F. (1990). Learning in neural network memories. *Network: Computational Neural Systems* 1, 105-122.

Amit, D. J., Campbell, C., and Wong, K. Y. M. (1989a). The interaction space of neural networks with sign-constrained synapses. *Journal of Physics A: Mathematical and General* 22, 4687.

Amit, D. J., Wong, K. Y. M., and Campbell, C. (1989b). Perceptron learning with sign-constrained weights. *Journal of Physics A: Mathematical and General* 22, 2039.

Bohland, J., and Minai , A. (2001). Efficient Associative Memory Using Small-World Architecture. *Neurocomputing* 38-40, 489-496.

Campbell, C., and Robinson, A. (1991). On the storage capacity of neural networks with sign-constrained weights. *Journal of Physics A: Mathematical and General* 24, L93.

Dale, H. H. (1935). Pharmacology and nerve endings. *Proceedings of the Royal Society of Medecine* 28, 319-332.

Davey, N., Adams, R. G., and Hunt, S. P. (Year). "High Performance Associative Memory Models and Symmetric Connections." Paper presented at the

International ICSC Congress on Intelligent Systems and Applications (ISA 2000), 2000a.

Davey, N., and Hunt, S. P. (Year). "A Comparative Analysis of High Performance Associative Memory Models." Paper presented at the 2nd International ICSC Symposium on Neural Computation (NC 2000), Berlin, 2000b.

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

Floréan, P., and Orponen, P. (1993). Attraction radii in binary Hopfield nets are hard to compute. *Neural Computation* 5, 812-821.

Gardner, E. (1988). The space of interactions in neural network models. *Journal of Physics A* 21, 257-270.

Gardner, E., Gutfreund, H., and Yekutieli, I. (1989). The Phase Space of Interactions in Neural Networks with definite Symmetry. *Journal of Physics A* 22, 1995-2008.

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

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

Kepler, T. B., and Abbot, L. F. (1988). Domains of attraction in neural networks. *Journal of Physics: France* 49, 1657-1662.

Kepler, T. B., and Abbott, L. F. (1988). Domains of attraction in neural networks. *Journal Physique de France* 49, 1657-1662.

Krauth, W., and Mezard, M. (1987). Learning algorithms with optimal stability in neural networks. *Journal of Physics A: Mathematical and General* 20, L745-L752.

Krauth, W., Nadal, J.-P., and Mezard, M. (1988). The roles of stability and symmetry in the dynamics of neural networks. *Journal of Physics A: Mathematical and General* 21, 2995-3011.

Nardulli, G., and Pasquariello, G. (1991). Domains of attraction of neural networks at finite temperature. *Journal of Physics A: Mathematical and General* 24, 1103.

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

Sompolinsky, H. (1986). Neural networks with nonlinear synapses and a static noise. *Physical Review A: Atomic, Molecular, and Optical Physics* 34, 2571-2574.

Storkey, A., and Valabregue, R. (1999). The basins of attraction of a new Hopfield learning rule. *Neural Networks* 12, 869 - 876.

Viswanathan, R. R. (1993). Sign-constrained synapses and biased patterns in neural networks. *Journal of Physics A: Mathematical and General* 26, 6195.

Watts, D., J., and Strogatz, S., H. (1998). Collective dynamics of 'small-world' networks. *Nature* 393, 440-442.

Wong, K. Y. M., and Campbell, C. (1992). Competitive attraction in neural networks with sign-constrained weights. *Journal of Physics A: Mathematical and General* 25, 2227.