Associative memory of low activity patterns without the problem of spurious attractors

G. Athithan

Advanced Numerical Research and Analysis Group, Kanchanbagh, Hyderabad 500 058, India

A neural associative memory network for the storage of fixed and low activity patterns is reviewed. The problem of spurious attractors in this network is solved by extending a method proposed earlier for the standard Hopfield network. By scaling the positive thresholds at neural sites with the instantaneous activity in the network it is shown that the basins of attraction for the stored patterns are enlarged considerably. An extension of these techniques to the case of variable activity patterns is presented. For practical applications, an optimal learning rule with a bound on the magnitude of the connection strengths is experimented. The distribution of the connection strengths computed by this learning rule is observed to peak near the bound. It is, therefore, proposed that reducing the connection strengths to +1 or –1 depending on their signs would enable an efficient realization of the network in hardware.

THE modelling of associative memory in terms of a network of elementary neurons is an active area of research in recent times[1–3]. Studies on such modelling would help in understanding the functioning of living brains besides leading to solutions for many practical problems in computer science. After Hopfield proposed a network for associative memory based on an adaptation of the Hebbian principle of learning[4], a lot of work has been done on the network and its variants[2,5].

Approximately half the number of neurons have to be active in the Hopfield network while representing a stored pattern. This corresponds to an activity level of 50% in the network which is neurobiologically very high. Experimental studies of the cortex[6] indicate that a large fraction of its neurons are always silent and the general activity levels are quite low, certainly much lower than 50%. Low activity patterns correspond to correlated patterns in the context of the Hopfield network which cannot be learnt using the Hebbian rule. Therefore, to understand the associative memory capabilities of the human cortex it is necessary to study other networks which are capable of handling low activity patterns.

One of the early attempts to model associative memory of low activity patterns is due to Willshaw *et al.*[7]. A network based on their ideas was analysed by Golomb *et al.*[8]. Their conclusion is that the storage capacity of the Willshaw network is somewhat lower than that of the other proposed networks. In particular, the network proposed by Tsodyks and Feigel'man[9] based on the Hebbian principle has been shown to have very high storage capacity when the activity levels are extremely low. A significant aspect of their network is the biologically realistic 0/1 coding instead of the –1/+1 coding.

Most associative memory networks proposed in the literature suffer from the problem of spurious attractors[10–12]. Computer experiments confirm the presence of spurious patterns in the network proposed by Tsodyks and Feigel'man. A generic solution to the problem of spurious patterns in the standard Hopfield network has been proposed earlier[10,11]. With a suitable modification the solution is extended here to the network proposed by Tsodyks and Feigel'man.

One of the important measures of performance of any associative memory network is its noise tolerance. This measure is related to the sizes of basins of attraction of the stored patterns. The larger the basin of a stored pattern, the larger is the allowed noise margin in recalling the stored pattern and smaller is the content size with which to initiate the recall. In the Tsodyks and Feigel'man network if the threshold related to the constant activity level of the patterns to be stored is kept fixed, then the basins of stored patterns are found to be very small. By adapting the threshold to the instantaneous activity in the network it is observed that the basins of attraction are enlarged considerably.

In practice, the activities of patterns to be stored may vary from pattern to pattern though they be somewhat low. To handle such variable activities, the Hebbian learning rule for fixed activity patterns can be generalized in a natural way. This generalization entails adapting the threshold of neural firings as one does not know *a priori* which stored pattern is to be recalled for a given input key. The problem of spurious patterns in the case of storing variable activity patterns can be solved in a way similar to the case of the fixed activity patterns.

An optimal learning rule for patterns of high activity[13–15] can be extended to the case of low activity patterns with 0/1 coding. This rule requires that linear bounds be imposed on the interaction strengths of the neurons in the network. Through computer experiments it has been observed that a very large fraction of the

interaction strengths assume either the maximum permissible negative value or the maximum positive value. This fact suggests that clipping the interaction strengths according to their sign would not alter the performance of the network significantly. Such a clipping would enable one to implement the network in hardware for practical applications. Neural network models are inherently parallel systems and using hardware implementations very high speeds of operation can be achieved. Earlier Moopenn *et al.*[16] have reported an hardware implementation of a network based on the Willshaw model. They also report some method of handling spurious patterns. Compared to their approach for storing low activity patterns without spurious attractors, the approach based on linear programming would be optimal and robust.

The paper is organized as follows. Firstly the network model proposed by Tsodyks and Feigel'man is discussed briefly. This is followed by the description of the method for circumventing spurious patterns in that network. The idea of adaptive threshold is discussed next so as to enlarge the basins of attraction for stored patterns. The Hebbian learning rule of the network is generalized for variable activity patterns. The method for circumventing spurious patterns is extended to the case of storing variable activity patterns. The application of an optimal learning rule for storing low activity patterns is discussed next. An hardware implementation of the optimal rule is suggested based on the examination of the distribution of the interaction strengths. Some related issues and possible extensions of the work reported here are discussed in the last section of the paper.

A network model for storing low activity patterns

Let N denote the number of two-state neurons in the network and s_i the state of the ith neuron. Keeping biological realism in mind, the silent state of a neuron is denoted by $s_i = 0$. The firing state is denoted by $s_i = 1$. We consider a completely connected network. Every neuron interacts with every other neuron in the network. The influence of neuron j on neuron i is denoted by the interaction strength, J_{ij}. All self-interactions, $\{J_{ii}\}$, are set to zero. The following equations define the dynamics of the network.

$$h_i(t) = \sum_{j=1}^{N} J_{ij} s_j(t), \tag{1a}$$

$$s_i(t+\delta t) = \frac{(\text{sign}\,[h_i(t) - \chi] + 1)}{2}, \tag{1b}$$

where $h_i(t)$ denotes the time-dependent local field at the ith neuron and χ denotes the threshold. A positive value for χ as a function of the activity level of the pattern is a must for optimal performance of the network as we shall see later.

The updating of the neural states using eqs (1a) and (1b) is to be carried out for one neuron at a time. The neuron to be updated is picked at random. Such an approach is generally called the asynchronous or serial updating. Alternatively, one can update all the N neurons simultaneously which is called the synchronous or parallel updating. Each scheme has its merits as well as drawbacks. The asynchronous scheme is biologically realistic and, if an energy function exists, would always lead the network to a stable point. The network would not get into limit cycles. However, one problem with the implementation of the asynchronous scheme is that the random choice of neurons for updating may leave one or more specific neurons unupdated for long durations, though the likelihood of this possibility is small. Statistically, updating of all the neurons cannot be ensured in any finite number of individual updates. The synchronous scheme ensures the updating of all N neurons every time, but may sometimes lead to undesired convergence to limit cycles of length two. Also, the synchronous scheme is biologically not plausible as there is no evidence of any synchronizing mechanism in living brains.

By modifying the asynchronous scheme at the implementation level, all the N neurons can be updated in what may be called one block-serial update. Firstly, a random permutation of the neurons is selected. The asynchronous updating of one neuron at a time is carried out according to this random permutation ensuring the update of all N neurons. This amounts to one block-serial update. The next block-serial update is carried out using a new random permutation of the neurons. It is, of course, simpler to keep a constant permutation of neurons for all block-serial updates. However, using a constant permutation is biologically less justifiable than using a new permutation for every block-serial update. Checking for convergence to a steady state in the case of a simple asynchronous scheme is a bit involved. With the block-serial approach, convergence is checked by comparing the states of the network before and after a block-serial update. In the experiments reported in this paper, block-serial scheme is used for updating the networks unless stated otherwise.

With the states of the neurons being represented by 0 or 1, the pattern vectors to be stored also have either 0 or 1 as their elements. Regarding the activity levels of the patterns to be stored, a simplifying assumption of constant activity levels may be made. Thus, denoting the constant fractional activity by f_a, each of the p patterns $\{\zeta_i^\mu : i = 1,.. N; \mu = 1,.., p\}$ to be stored may have exactly Nf_a nonzero elements. The Hebbian learning rule for this model is[9],

$$J_{ij} = \sum_{\mu=1}^{p} \frac{(\zeta_i^\mu - f_a)(\zeta_j^\mu - f_a)}{(1 - f_a)}, \tag{2}$$

$J_{ii} = 0$ for all i.

It is important to note that the p pattern vectors are assumed to be random and are uncorrelated as far as their active elements, that is ones, are concerned. To see if a specific pattern $\{\xi_i^\nu : i = 1,..,N\}$ is stable with respect to dynamics (eq. (1)) a preliminary signal-to-noise ratio analysis can be carried out. Assuming the network to be in the state corresponding to the νth pattern, the local potential at neural site i is calculated as

$$h_i\left(\{\xi_i^\nu\}\right) = \sum_{j=1}^{N} J_{ij}\xi_j^\nu$$

$$= \sum_{j\neq i}\sum_{\mu=1}^{p} \frac{\left(\xi_i^\mu - f_a\right)\left(\xi_j^\mu - f_a\right)\xi_j^\nu}{(1-f_a)} \qquad (3)$$

$$= Nf_a\left(\xi_i^\nu - f_a\right) + \sum_{\mu\neq\nu}\sum_{j\neq i} \frac{\left(\xi_i^\mu - f_a\right)\left(\xi_j^\mu - f_a\right)\xi_j^\nu}{(1-f_a)}.$$

The first term on the right hand side of eq. (3) is the signal and the second term denotes the noise which is approximately Gaussian with a zero mean. The standard deviation is easily computed to be $Nf_a\sqrt{\alpha f_a}/(1-f_a)$, where $\alpha = p/N$. The signal term is biased towards positive values and needs a nonzero threshold to make it swing to the positive and negative sides with equal likelihood. Such a threshold value χ, can be seen as $Nf_a(1-2f_a)/2$. This threshold value would make the silent states as stable as the firing states. A positive threshold is also a biologically plausible entity and can play some useful role in widening the basins of attraction as seen in numerical simulations. With this threshold term, the order of the signal is Nf_a. Therefore, noise is proportional to $\sqrt{\alpha f_a}/(1-f_a)$ and extensive storage of patterns is possible. In fact, it has been shown that the critical value of the memory loading parameter, α, is $1/(2f_a|\ln f_a|)$ when f_a is very low[9].

Incidentally, when $f_a = 0.5$, the threshold reduces to zero and this network reduces to the standard Hopfield network. Firstly, the synaptic connections defined by eq. (2) reduce to those computed by a similar equation for the Hopfield network except for a uniform scale factor. Secondly, the local potential h_i, computed by eq. (1a) is not changed if the following equation is used, instead.

$$h_i(t) = \sum_{j=1}^{N} J_{ij}(s_j(t) - f_a). \qquad (4)$$

This can be proved easily by showing that the contribution due to the additional f_a term in eq. (4) is effectively zero for large N. Eq. (4) is, of course, what one uses in the case of the dynamics of the Hopfield network. This equivalence rests on the fact that when f_a is the fractional activity, exactly Nf_a elements of any pattern are ones.

Avoiding spurious patterns

Numerical simulations confirm that the learning rule (eq. (2)) is not free from the problem of spurious patterns. The solution proposed for the Hopfield network[10] can easily be extended as described below. It is easy to see that for a pattern $\{\xi_i^\mu\}$ to be stable, the following condition must be satisfied for all i.

$$\left(2\xi_i^\mu - 1\right)\left[\sum_{j=1}^{N} J_{ij}\xi_j^\mu - \chi\right] > 0. \qquad (5)$$

As in the case of the Hopfield network, a parameter h_{mincp} is defined as

$$h_{\text{mincp}} = \operatorname*{Min}_{\mu}\left[\operatorname*{Min}_{i}\left(2\xi_i^\mu - 1\right)\left[\sum_{j=1}^{N} J_{ij}\xi_j^\mu - \chi\right]\right]. \qquad (6)$$

If h_{mincp} is positive, then all the p patterns are stable. The larger the value of h_{mincp}, the larger are the basins of attraction for the stored patterns. In the context of the Hopfield network, it has been observed[12] that when the loading level of the network, α, is less than 0.051, the average local potentials of correct patterns are higher than those of the spurious patterns. This critical loading level, $\alpha_c^1 = 0.051$, is applicable to the learning rule (eq. (2)) when f_a assumes a value of 0.5. It has been observed in computer simulations[11] that the critical loading level $\alpha_c^2(f_a)$, beyond which correct patterns become no longer stable, is inversely related to f_a. This observation is consistent with analytical studies[9] as well as with the preliminary signal-to-noise ratio analysis presented in the previous section. Therefore, it is reasonable to expect that $\alpha_c^1(f_a)$ also is likely to be inversely related to f_a.

Computer simulations show that when α is less than 0.051, for a range of f_a from 0.05 to 0.5, the average local potentials of the spurious patterns are lower than those of the correct patterns. Denoting a spurious pattern with 0/1 elements by $\{w_i^\nu : i = 1,\ldots,N\}$ one may compute the parameter h_{maxsp} over all possible spurious patterns as

$$h_{\text{maxsp}} = \operatorname*{Max}_{\nu}\left[\operatorname*{Min}_{i}\left(2w_i^\nu - 1\right)\left[\sum_{j=1}^{N} J_{ij}w_j^\nu - \chi\right]\right]. \qquad (7)$$

With low loading of the network, i.e. with $\alpha < 0.05$, h_{mincp} is expected to be higher than h_{maxsp}. Since $\alpha_c^1(f_a)$ is likely to be inversely related to f_a, its value is likely to be more than 0.051, if f_a is lower than 0.5. As a result, h_{mincp} is likely to be higher than h_{maxsp} even for loading levels beyond 0.051, when the patterns have low activities. The value of the parameter h_{maxsp} can be easily evaluated by a large number of random Monte Carlo recall trials[10].

Once a positive bandgap is obtained between h_{mincp} and h_{maxsp}, a suitable value of h_{self} can be selected in the bandgap and used in the recall dynamics as follows.

$$h_i(t) = -h_{\text{self}}(2s_i(t)-1) + \sum_{j\neq i} J_{ij}s_j(t). \quad (8)$$

Defining $J_{ii} = -2h_{\text{self}}$ one may write eq. (11) as

$$h_i(t) = h_{\text{self}} + \sum_{j=i}^{N} J_{ij}s_j(t). \quad (9)$$

From an examination of eq. (8) it is easy to see that the stability of correct patterns would not be impaired. At the same time one or more neural sites in every spurious pattern would be made unstable due to inhibitory self-interaction.

When the network is not able to recall any valid pattern, it is preferable that it settles in the no-activity state instead of going into a time-dependent state. It is easy to see that the use of inhibitory self-interaction as employed in eq. (8) would not let the network settle in the no-activity state. The following modification of eq. (8) would remove this drawback.

$$h_i(t) = -h_{\text{self}}s_i(t) + \sum_{j\neq i} J_{ij}s_j(t). \quad (10)$$

This equation destabilizes only the active elements in any spurious pattern and leaves the passive elements undisturbed. Therefore, the parameters, h_{mincp} and h_{maxsp} also need to be computed only for those neural sites i which correspond to active states. In the computer experiments reported in the following sections, inhibitory self-interaction is used as indicated in eq. (10).

Numerical simulations were carried out for checking the effectiveness of the inhibitory self-interaction term in eq. (10). The experiments were carried out for a network of size $N = 500$. The loading level α was fixed at 0.05. The activity level of the patterns to be stored was varied from 0.05 to a maximum of 0.5 in steps of 0.05 and all patterns in an experiment were assigned a constant activity selected within that range. Block-serial dynamics was used uniformly. For each set of patterns, h_{mincp}, h_{maxsp}, and the fractional volumes of basins of the correct and the spurious patterns, namely f_c and f_s were evaluated. For evaluating h_{maxsp}, 1,000 experiments were carried out in each case. The fractional volumes were evaluated both with and without inhibitory self-interaction. The activity of the random input patterns was kept the same as that of the stored patterns in each case. The results are presented in Table 1. In all the ten cases of random, low activity patterns, a positive bandgap has been observed. Without the self-interaction term recall experiments sometimes converge to spurious patterns.

Once the self-interaction term is switched on, the network always converges to the correct patterns. If a correct recall is not possible it goes into the no-activity state or to a limit cycle.

Adaptive threshold

The data in Table 1 indicate that the fractional volumes of correct patterns are mostly zeros when the activity levels are low. The patterns to be stored actually correspond to stable points of dynamics in eqs (1a) and (1b). But their basins of attraction seem to be very small when f_a is small. The source of the problem actually lies with the constant positive threshold χ. Only when the network is near a correct pattern, there is enough signal strength to overcome the large threshold value $\chi = Nf_a(1 - 2f_a)/2$. If either a small content of a stored pattern or a highly corrupted version of it is presented as input, the neurons are not able to overcome the large positive threshold.

A remedy to this problem is to scale the threshold with the instantaneous activity of the network. That is, using the current activity

$$a(t) \equiv \sum_i s_i(t), \quad (11)$$

the threshold χ should be set as

$$\chi = a(t)(1 - 2f_a)/2. \quad (12)$$

When the network is at one of the stored patterns, $a(t)$ would equal Nf_a; therefore, the stability of the stored patterns is not affected. With either a small percentage of the contents of the stored patterns or their noisy versions as input, the strength of the signal is expected to be small at the neural sites. However, since the threshold is also proportional to the level of the signal, the neurons which need to go into the active state so as to recall an intended pattern would be enabled to do so. To verify this observation, the set of computer experiments whose results are presented in Table 1 were repeated with the adaptive threshold. Here also the activity of random input patterns in the Monte Carlo recall experiments were kept the same as that of the stored patterns. The results are presented in Table 2.

The significant improvement in the fractional volumes of the stored patterns following the use of adaptive threshold is very clear from the data in Table 2. The fractional volume f_c is quite close to one in several cases. Also, once the inhibitory self-interaction is switched on, the few recalls of spurious patterns also disappear. The data also point to another factor of performance. The fractional volume f_c gradually decreases with increasing activity levels. This implies that low activity encoding is more tolerant to noise during recall.

For a visual comparison of performances with and without adaptive threshold, the data relating to f_c in

Table 1. Results of experiments on learning constant activity patterns for a network of size 500

f_a		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
h_{mincp}		7.96	13.28	15.6	9.5	12.41	18.07	20.48	31.65	18.11	22.5
h_{maxsp}		0.0	0.0	0.0	0.0	0.0	4.14	5.46	6.66	6.84	8.5
$J_{ii}=0$	f_c	0.0	0.0	0.0	0.0	0.002	0.193	0.504	0.298	0.210	0.195
	f_s	0.0	0.0	0.0	0.0	0.0	0.083	0.469	0.702	0.790	0.805
$J_{ii}' \neq 0$	f_c	0.0	0.0	0.0	0.0	0.006	0.247	0.76	0.746	0.334	0.358
	f_s	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

The constant fractional activity, f_a, was varied from 0.05 to 0.5. In each case h_{mincp} and h_{maxsp} were evaluated. The fractional volumes of correct and spurious patterns, f_c and f_s, were evaluated with and without self-interaction. For evaluating h_{maxsp} and the fractional volumes 1,000 recall experiments were carried out in each case.

Table 2. Results of experiments on learning constant activity patterns for a network of size 500

f_a		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
h_{mincp}		7.96	13.28	15.6	9.5	12.41	18.07	20.48	31.65	18.11	22.5
h_{maxsp}		0.0	0.324	0.728	2.11	4.21	4.55	5.57	6.06	7.44	7.88
$J_{ii}=0$	f_c	0.715	0.962	0.995	0.977	0.856	0.673	0.472	0.296	0.21	0.186
	f_s	0.0	0.001	0.002	0.02	0.133	0.313	0.504	0.668	0.779	0.814
$J_{ii} \neq 0$	f_c	0.713	0.977	0.999	0.996	0.988	0.956	0.768	0.764	0.289	0.358
	f_s	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

The experiments were carried out with adaptive threshold. The meanings of the parameters are the same as in Table 1.

Tables 1 and 2 are presented in a graphical form in Figure 1. These data correspond to the case of $J_{ii}=0$. At lower values of f_a the difference in performance is remarkable. This difference reduces to zero, as f_a increases towards 0.5.

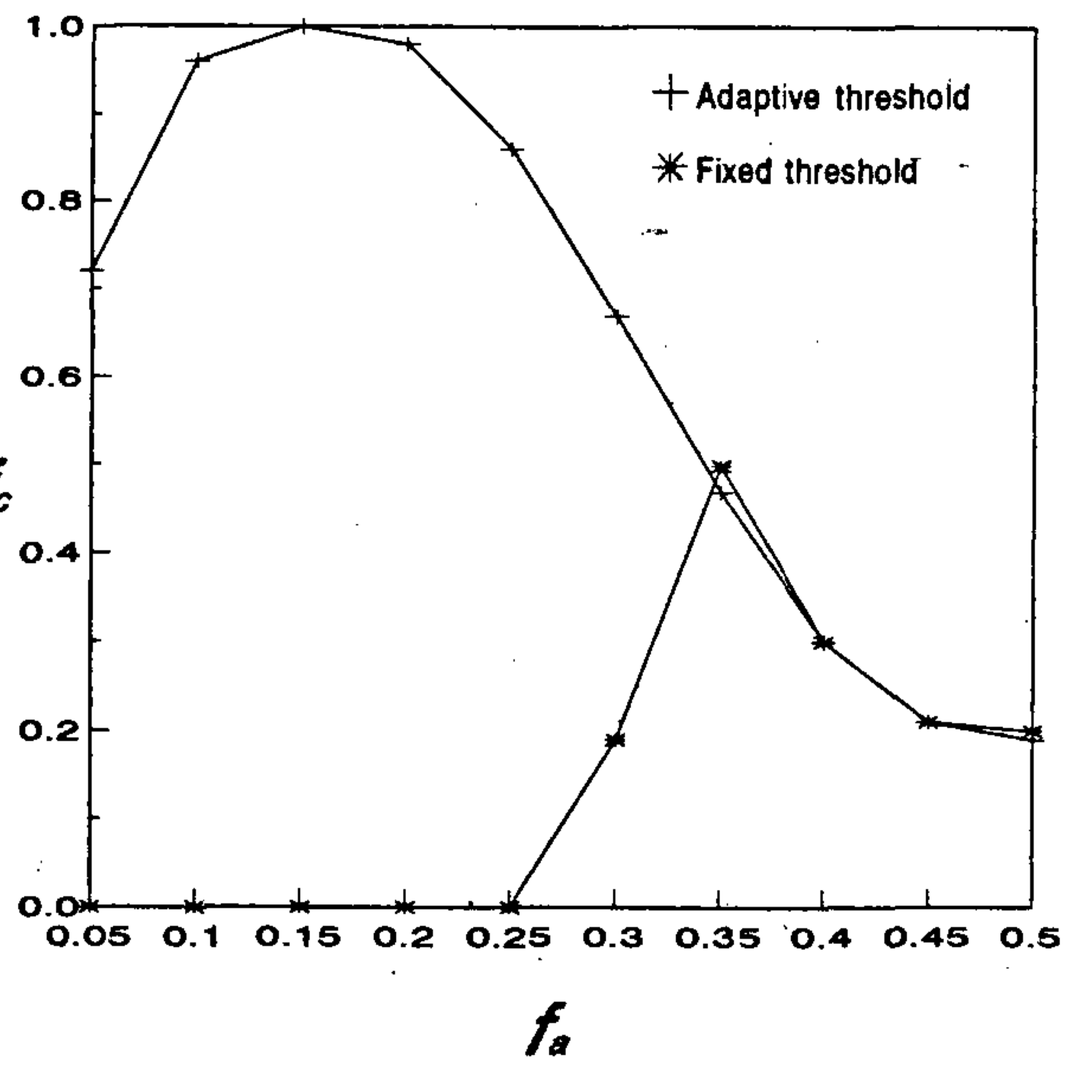

Figure 1. The fractional volumes of basins of correct patterns, f_c, with and without adaptive threshold are shown here as a function of the activity level, f_a, of the patterns. These data correspond to the case of $J_{ii}=0$.

Variable activity patterns

The learning rule (eq. (2)) would work satisfactorily only if all the p patterns to be stored have approximately Nf_a non-zero elements. In practice, one often needs to handle patterns whose activities might greatly differ from one another. Denoting the fractional activity of the μth pattern by f_a^μ the learning rule may be modified as below so that it can handle variable activity patterns also.

$$J_{ij} = \sum_{\mu=1}^{p} \frac{\left(\xi_i^\mu - f_a^\mu\right)\left(\xi_j^\mu - f_a^\mu\right)}{\left(1-f_a^\mu\right)}. \tag{13}$$

The patterns have to be uncorrelated in their active elements for eq. (13) to work. Using a preliminary signal-to-noise ratio analysis as shown earlier one can see that

$$h_i\left(\{\xi_i^\nu\}\right) = N f_a^\nu\left(\xi_i^\nu - f_a^\nu\right) + \sum_{\mu\neq\nu}\sum_{j\neq i} \frac{\left(\xi_i^\mu - f_a^\mu\right)\left(\xi_j^\mu - f_a^\mu\right)\xi_j^\nu}{\left(1-f_a^\mu\right)}. \tag{14}$$

As before, the first term is the signal and the second is a Gaussian noise term with a zero mean. To make the signal term swing to positive and negative values equally, the threshold χ should be $Nf_a^\nu(1-2f_a^\nu)/2$. Since one does not know the activity of the pattern to be actually recalled in an experiment *a priori*, the threshold has to become perforce adaptive. As discussed in the previous section, adaptive threshold is necessary for larger basins of attraction. Here one sees that it is also necessary for the storage and recall of variable activity patterns. One small problem is that f_a^ν occurs in two places in the expression for χ. If f_a^ν at both these places are scaled

with the instantaneous fractional activity, $f_a(t)$, it is possible that frequently the network may settle in the state where all neurons are active. Computer experiments corroborate this observation. This is due to the fact that as $f_a(t)$ increases beyond 0.25, the threshold value actually decreases leading to the activation of more and more neurons till all of them become active. Inhibitory signals are overcome by the ever decreasing threshold. This kind of behaviour of the network is not useful. Using the current activity $a(t)$, a solution to this problem is to set the threshold as

$$\chi(t) = a(t)(1 - 2f_a(t))/2, \tag{15}$$

where $f_a(t)$ equals $a(t)/N$ but is clipped to remain within the range of activities of the stored patterns. This new form of $\chi(t)$ prevents the network from going to the state where all neurons would be active.

The method of circumventing the spurious patterns discussed earlier is applicable to the case of variable activity patterns as well. However, if the range of activities of the stored patterns is somewhat large, it is possible that a spurious pattern with high activity may have larger local potentials compared to some correct patterns with lower activities. This may frequently lead to a negative bandgap between h_{mincp} and h_{maxsp} even when one operates in the low α range. A natural solution to this problem is to compute h_{mincp} and h_{maxsp} for unit activity of the patterns. That is, one may define h_{umincp} and h_{umaxsp} as

$$h_{\text{umincp}} = \operatorname*{Min}_{\mu}\left[\operatorname*{Min}_{i}\left[\left(2\xi_i^\mu - 1\right)\sum_{j\neq i} J_{ij}\xi_j^\mu - \chi\left(\xi_i^\mu\right)\right]\right] / Nf_a^\mu, \tag{16}$$

$$h_{\text{umaxsp}} = \operatorname*{Max}_{\nu}\left[\operatorname*{Min}_{i}\left[\left(2w_i^\nu - 1\right)\sum_{j\neq i} J_{ij}w_j^\nu - \chi\left(w_i^\nu\right)\right]\right] / \sum_j w_j^\nu, \tag{17}$$

where $\{w_i^\nu\}$ denotes a spurious pattern indexed by ν. The threshold $\chi(.)$ is to be computed for the correct or spurious patterns suitably using eq. (15).

These unit activity local fields are more likely to have the required bandgap with h_{umincp} being greater than h_{umaxsp}. The self-interaction factor, h_{uself}, can be selected between h_{umincp} and h_{umaxsp} and used in the computation of local potentials in the following manner.

$$h_i(t) = -a(t)h_{\text{uself}}s_i(t) + \sum_{j\neq i} J_{ij}s_j(t). \tag{18}$$

It is easy to see that using eq. (18) would not affect the stability of the correct patterns whereas the spurious patterns would become unstable due to the self-interaction term.

Computer experiments were carried out to verify the effectiveness of the unit activity-based self-interaction factor. Two network sizes $N = 500$ and 1,000 were considered. In each case α was set as 0.05. Four ranges of fractional activities of patterns to be stored, viz. 0.05–0.2, 0.06–0.21, 0.07–0.22, and 0.08–0.23 were used. In each case h_{umincp} and h_{umaxsp} were calculated. To evaluate h_{umaxsp} 1,000 random recall experiments were carried out. The results are presented in Table 3. The bandgap, $h_{\text{umincp}} - h_{\text{umaxsp}}$, is positive in all the eight cases.

As a byproduct of computing h_{umaxsp}, the fractional volumes of basins of stored patterns and the spurious patterns, f_c and f_s, were also computed without self-interaction. The results are presented in Table 4. With the use of adaptive threshold, the values of f_c turn out to be very good, being close to one. When the self-interaction term as defined in eq. (18) was switched on, the few spurious recalls also disappeared as was observed in computer experiments. In the presence of inhibitory self-interaction, f_c actually increases slightly.

The fractional volume of basins of correct patterns improves dramatically with the use of adaptive threshold as the simulation results show. This improvement is obtained both with sets of constant and variable activity patterns. With f_c close to one, any input key having an activity within the range of activities of the stored patterns would result in a recall of some stored pattern. Only if the space of input clues is known to be free of noise and clues of unknown or new patterns, very high f_c is desirable. Otherwise, when the input key has inadequate content for a valid recall, it is preferable that the network goes into a no-activity state rather than recall something meaningful. Such a performance can be obtained by limiting the adaptive threshold to a lower bound.

Table 3. The values of h_{umincp} and h_{umaxsp} for $N = 500$ and 1000

	rf_a^μ	0.05–0.2	0.06–0.21	0.07–0.22	0.08–0.23
$N = 500$	h_{umincp}	0.256	0.193	0.143	0.127
	h_{umaxsp}	0.033	0.056	0.058	0.079
$N = 1000$	h_{umincp}	0.089	0.148	0.110	0.137
	h_{umaxsp}	0.027	0.003	0.052	0.013

The loading level of the network, α, was set at 0.05 for the experiments. The parameter rf_a^μ denotes the range of fractional activities of the stored patterns.

Table 4. Fractional volumes of basins of correct patterns and spurious patterns, for two sizes of the network

	rf_a^μ	0.05–0.2	0.06–0.21	0.07–0.22	0.08–0.23
$N = 500$	f_c	0.967	0.979	0.987	0.980
	f_s	0.005	0.009	0.008	0.010
$N = 1000$	f_c	0.997	0.996	0.993	0.996
	f_s	0.002	0.004	0.007	0.003

The parameter rf_a^μ denotes the range of activities of the patterns to be stored.

Linear programming rule

To store p patterns of low and variable activities, $\{\xi_i^\mu : i = 1,\ldots,N;\ \mu = 1,\ldots,p\}$, possibly correlated in their active elements, the following condition must be satisfied for all μ and i.

$$\left(2\xi_i^\mu - 1\right)\sum_{j\neq i} J_{ij}\xi_j^\mu > k, \tag{19}$$

where k is a positive parameter to be maximized. Linear bounds on the $\{J_{ij}\}$ are to be imposed in order to keep the maximum bounded. That is, the following constraints have to be added to eq. (19) before solving for $\{J_{ij}\}$.

$$|J_{ij}| < J_{\max}. \tag{20}$$

The problem of maximizing k subject to the constraints eqs (19) and (20) is a linear programming problem and is solved using the standard simplex algorithm[17]. The $\{J_{ij}\}$ are obtained as a byproduct.

Recall can be carried out using dynamics, i.e. eqs (1a) and (1b). As in the case of earlier learning rules, spurious patterns may be created by the present rule also. By computing h_{mincp} and h_{maxsp} using the eqs (6) and (7) respectively and using inhibitory self-interaction, the spurious patterns can be circumvented easily. One may also use asymmetric dilution of the synapses and improve the sizes of basins for stored patterns[10].

Clipping the synapses for hardware realization

A final task is to see if this network can be implemented in hardware. Otherwise, one of the main advantages of the neural approach, which is its high operational speeds, cannot be realized in practice. Earlier, Moopenn *et al.*[16] studied a hardware implementation of the Willshaw learning rule[7]. The elements of $\{J_{ij}\}$ computed using the Willshaw learning rule turn out to be either 1 or 0; therefore, a hardware implementation becomes practical. The Hebbian rules eqs (2) and (13) do not give rise to a $\{J_{ij}\}$ matrix with 0/1 elements. Clipping the $\{J_{ij}\}$ matrix elements to +1 or −1 is possible depending on their signs but that would result in a degradation in performance. Moreover, the Hebbian rules are not optimal anyway.

An examination of the distribution of $\{J_{ij}\}$ computed by optimizing k in constraint eq. (19) was made in order to see if clipping would be acceptable. It revealed that more than about 80% of the J_{ij}s are either $+J_{\max}$ or $-J_{\max}$. This is mainly due to the imposition of linear bounds on the magnitudes of J_{ij}s. This fact suggests that clipping $\{J_{ij}\}$ to either +1 or −1 depending on their signs would not affect the stability of the stored patterns very much. After clipping, h_{mincp} can be computed using the new $\{J_{ij}\}$ to see if all the p patterns are stable or not. Computer experiments with $N = 200$ and 300 confirm that clipping does not affect the value of h_{mincp} significantly. The fractional count of $\{J_{ij}\}$ which are either $+J_{\max}$ or $-J_{\max}$ were calculated from these experiments for four cases of activity levels. In each case ten different sample sets of random and uncorrelated patterns were used for finding the mean and the standard error of the fractional counts. The results are presented in Table 5.

The data in Table 5 suggest that clipping the $\{J_{ij}\}$ would be acceptable so that a hardware realization as outlined by Moopenn *et al.* can be tried. Combined with the feasibility of storing low activity patterns and circumventing spurious ones, the clipped $\{J_{ij}\}$ computed by the linear programming rule would be more efficient and robust than that of the Willshaw learning rule.

Discussion

The solution to the problem of spurious patterns proposed in this paper is simple to implement in practice. After the computation of the synaptic strengths, the h_{mincp} can be computed easily. The h_{maxsp} can be estimated by a Monte Carlo method. The spurious pattern associated with the value of h_{maxsp} is expected to have a large basin of attraction. Therefore, in a finite number of Monte Carlo recall experiments, it is likely to be visited at least once leading to the correct estimation of h_{maxsp}. If h_{mincp} is greater than h_{maxsp}, the spurious patterns can be avoided completely. The results of computer simulations presented in this paper demonstrate the effectiveness of this new solution. With low loading levels and low activities of stored patterns, $h_{\text{mincp}} - h_{\text{maxsp}}$ turns out to be positive mostly.

Most neural networks when used as associative memories would suffer from the problem of spurious patterns in some form or the other. The solution proposed in this paper can be easily adapted to all such networks for circumventing the spurious patterns. The parameters analogous to h_{mincp} and h_{maxsp} need to be computed first. Once a positive bandgap, $h_{\text{mincp}} - h_{\text{maxsp}}$, is available, inhibitory self-interaction of suitable magnitude can be added to the neurons to make them either unstable or inactive near any spurious pattern.

Computer simulations demonstrate that the basins of stored patterns are enlarged considerably when the neural thresholds are scaled with instantaneous activity. The fractional volume of basins of attraction for stored patterns in fact tends towards the maximum value of one for a range of N from 500 to 2000. This is desirable in a situation where every input key is expected to recall some stored pattern, regardless of the inadequacy of the pattern content in the key. However, sometimes one may want the network to recognize a pattern only if a certain

Table 5. Fractional counts of synaptic strengths ($+J_{max}$, $-J_{max}$) for various values of f_a, α, and N

	f_a	0.05	0.1	0.2	(0.05 – 0.2)
$N = 200$	$\alpha = 0.1$	0.990 ± 0.0013	0.968 ± 0.0016	0.921 ± 0.0008	0.969 ± 0.0074
	$\alpha = 0.2$	0.975 ± 0.0014	0.904 ± 0.0013	0.843 ± 0.0011	0.888 ± 0.0099
$N = 300$	$\alpha = 0.1$	0.987 ± 0.0008	0.951 ± 0.0015	0.914 ± 0.0001	0.953 ± 0.0072

In each case the mean value of the fractional count and the standard error were computed using ten different sets of random patterns. In the first three columns the fractional activity level of the patterns is chosen to be fixed and it is varied in the last column within the range shown.

minimum content of it is presented as input key. For any input key having a lower content of the pattern, the network should perhaps go to the zero activity state, signalling the fact that either the input consists of insufficient content or the network is seeing a new pattern not stored earlier. This kind of performance can be obtained from the network by limiting the adaptive threshold to a lower bound. The value of this lower bound can be easily worked out from the minimum amount of content of any stored pattern which is stipulated to be necessary for a valid recall.

The associative memory network discussed in this paper recalls patterns auto-associatively. That is, given a part of a stored pattern, the complete pattern is recalled. In many applications one may need to recall cross-associatively, i.e. given a part of a pattern in one class to recall fully an associated pattern belonging to another class[18]. The perceptron networks, both the single layer and multi-layer types, are generally designed for cross-associative applications. The perceptron networks, however, do not use any dynamics explicitly and the recall is done in one step. They do not use any feedback of their output and are actually known as feed-forward networks. One generalization of the single layer perceptron network due to Kosko[19] uses feedback from the output layer to the input and brings relaxation dynamics into play. Patterns belonging to one class on the input layer would induce an associated one from another class on the output layer. Depending on the requirement either Hebb rule or some optimal rule can be used for ensuring stability of associations in Kosko's bidirectional associative network. As in the case of auto-associative dynamical networks, this bidirectional network also would suffer from the problem of spurious cross-associations. To solve this problem one can try the notion of self-interaction proposed in this paper.

One of the aims of studying associative memory models based on neural networks is to see if the storage and recall of patterns in the human brain can be understood. The patterns which a normal human brain handles such as faces of individuals, tunes, and words, to name a few classes, are highly correlated with each other. For instance, many spoken words may have common phonemes and written words may have common substrings of letters. The Hebbian learning rule eq. (2), which is consistent with known neurophysiological processes, however, cannot ensure the storage of correlated patterns. It is possible to construct local iterative learning rules to handle correlated patterns though they will not be biologically justifiable due to the need for multiple presentations of the patterns. Thus the problem of modelling human associative memory still remains to be solved completely.

1. Amit, D. J., *Modelling Brain Functions*, Cambridge Univ. Press, Cambridge, 1989.
2. van Hemmen, J. L. and Kuhn, R., in *Models of Neural Networks* (eds Domany, E., van Hemmen, J. L. and Shulten, K.), Springer-Verlag, Berlin, 1991, pp. 1–105.
3. Hopfield, J. J., Proceedings of National Academy of Sciences, USA, 1982, **79**, 2554–2559.
4. Hebb, D. O., *Organization of Behaviour*, Wiley, New York, 1949, p. 62.
5. Hertz, J., Krogh, A. and Palmer, R. G., *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company, USA, 1991.
6. Abeles, M., *Local Cortical Circuits*, Springer-Verlag, Berlin, 1982.
7. Willshaw, D. J., Buneman, O. P. and Longuet-Higgins, H. C., *Nature*, 1969, **222**, 960.
8. Golomb, D., Rubin, N. and Sompolinsky, H., *Phys. Rev.*, 1990, **A41**, 1843.
9. Tsodyks, M. V. and Feigel'man, M. V., *Europhys. Lett.*, 1988, **6**, 101–105.
10. Athithan, G. and Dasgupta, C., *IEEE Trans. Neural Networks*, 1997, **8**, 1483–1491.
11. Athithan, G., Ph D Thesis, Indian Institute of Technology, Mumbai, India, 1997.
12. Amit, D. J., Gutfreund, H. and Sompolinsky, H., *Ann. Phys.* 1987, **173**, 30–67.
13. Krauth, W. and Mezard, M., *J. Phys.*, 1987, **A20**, L745–L752.
14. Verleysen, M., Sirletti, B., Vendemeulebroecke, A. and Jespers, P. G. A., *IEEE Trans. Circuits Systems*, 1989, **36**, 762.
15. Athithan, G., *Pramana – J. Phys.*, 1995, **45**, 569–582.
16. Moopenn, A., Lambe, J. and Thakoor, A. P., *IEEE Trans. Systems, Man Cybernetics*, 1987, **17**, 325.
17. Gass, S. T., *Linear Programming*, McGraw-Hill, New York, 1969.
18. Widrow, B. and Lehr, M. A., *Proc. IEEE*, 1990, **78**, 1415.
19. Kosko, B., *IEEE Trans. Systems, Man, Cybernetics*, 1988, vol. SMC-18, 49–60.

Received 6 August 1998; revised accepted 22 December 1998.