Convergence of recursive associative memories obtained using the multilayered perceptron

1990 J. Phys. A: Math. Gen. 23 4039

(http://iopscience.iop.org/0305-4470/23/17/032)

J. Phys. A: Math. Gen. 23 (1990) 4039-4053. Printed in the UK

Convergence of recursive associative memories obtained using the multilayered perceptron

P J G Lisboa† and S J Perantonis‡

† Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3BX, UK

‡ Department of Applied Mathematics and Theoretical Physics, University of Liverpool, Liverpool L69 3BX, UK

Received 23 October 1989

Abstract. We describe two types of associative memory which are obtained by iterating a multilayered perceptron recursively after training it to auto-associate. The convergence of the associative memories with real-valued output and additional weights connecting the input and output layers directly is demonstrated analytically. Computational results are presented to illustrate the key concepts used in the proof, and also to characterise the capacity of these networks, comparing them with the Hopfield model and an alternative formulation of the multilayered auto-associative memory with thresholding to binary outputs. The capacity of the real-valued memory exceeds that of the alternative formulations when training with noise is used.

1. Introduction

Auto-association is one of the simplest ways of storing a bit pattern and it has been used extensively to illustrate the regenerative capacity of distributed memories using neural networks. At the same time, the multilayered perceptron has emerged as a network of potentially unlimited capacity. It is therefore natural to consider using it to form an auto-associative content addressable memory.

The network can be trained to autoassociate through the gradient back propagation algorithm (Le Cun 1985, Rumelhart *et al* 1986, Rumelhart and McLelland 1986). Recollection of the stored patterns can be achieved by the following process (algorithm A): A pattern is presented as input and recursively iterated through the network, until (hopefully) convergence upon one of the nominated patterns occurs (Fogelman Soulie *et al* 1987, Wieland and Leighton 1987) (figure 1). A variant of this algorithm (algorithm B) involves thresholding of the output after each iteration to make it binary (Gallinari, Fogelman Soulie and Thiria 1987). The network is then expected to stabilise after just a few iterations. Algorithm A has the advantage that small changes in the output can accumulate gradually to eventually flip the corrupted bits and may thus be expected to perform better than algorithm B. However, although the nominated patterns are fixed points of algorithm A by construction, it is not obvious that they are attractors of the recursive iteration.

In this paper we prove that the nominated patterns are attractors of algorithm A with non-zero basins of attraction for a multilayered neural network with top-to-bottom connections. We present numerical results which illustrate key steps of the proof and suggest that a similar result is also true even when the top-to-bottom synaptic connections are absent. Finally, we present further numerical results to estimate the capacity

0305-4470/90/174039+15$03.50

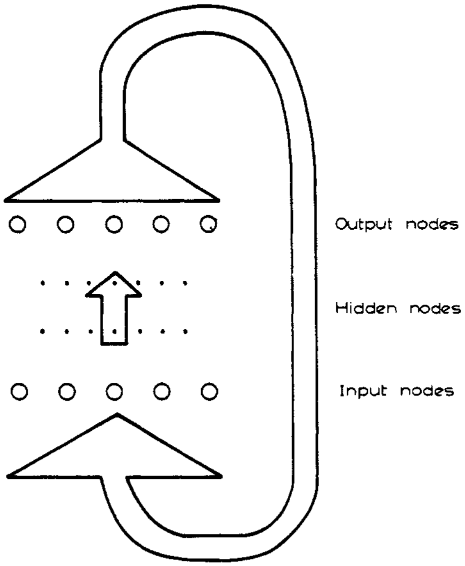

Figure 1. Direct feedback of the output of a multilayer network to its input.

of the recursive associative memory for the two algorithms, and to compare it with the capacity of the Hopfield model (Hopfield 1982). The effect of training with noise on the activity of the hidden units in multilayered networks is also discussed.

2. The associative memory

The multilayered perceptron is commonly used to perform classification of a set of input patterns into classes which are selected by exciting nodes in the output layer (Lippman 1987). Clearly, if the number of nodes in the input and output layers is the same, the network can be required to produce an exact copy of the input patterns at the output nodes. In this paper we shall use input patterns represented by vectors of real numbers (I_i^α, $i = 1, \ldots, N$, $\alpha = 1, 2, \ldots, P$). The subscripts i and α label the input nodes and the nominated patterns, respectively. The network nodes are arranged in layers, with no interlayer connections, as described in figure 2. The input into each

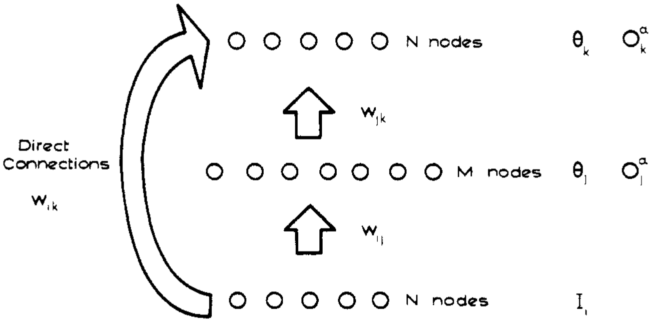

Figure 2. Architecture of a three-layer network with direct connections from the input to the output nodes.

node is a weighted sum of the outputs from all the nodes in the previous layers, to which a bias term is added to set the threshold level for that particular node. The output from this node is of the form

$$O_m = \left[1 + \exp\left(-\sum_n w_{mn} O_n - \theta_m\right)\right]^{-1} \tag{1}$$

where O_n are the outputs from the nodes in the previous layers, w_{mn} are the synaptic weights and θ_m is the bias term. The network is trained by adjusting the values of the weights and bias terms in response to the desired classification of the nominated patterns, with the aim of reducing the value of the cost function

$$E = \tfrac{1}{2} \sum_{i\alpha} (t_i^\alpha - O_i^\alpha)^2. \tag{2}$$

In the case of auto-association, the target values are set equal to the input patterns themselves I_k^α, with the effect that when the network outputs O_k^α reach their target values then the nominated patterns are self-reproducing. The network parameters w_{mn} and θ_m are adjusted by a gradient descent method, using the back-propagation algorithm.

We are mainly interested in the storage and recollection of patterns for which the I_i^α are close to the saturation values of the sigmoid function $f(x) = 1/[1+\exp(-x)]$, i.e. $I_i^\alpha = \delta$ or $I_i^\alpha = 1 - \delta$, where δ is small. If the network is taught in this way and then connected back-to-back so that each node in the output layer excites the corresponding input node directly (figure 1) then the stored patterns are expected to be stable when repeated forward passes are applied, as we propose to show. This is the motivation for using the network as an associative memory.

In contrast with an alternative approach (Gallinari *et al* 1987) we retain the outputs from the network in real valued form when they are fed back into the input layer instead of rounding them off to the nearest integer (0 or 1). The pattern is iterated during recall until convergence is achieved and only then is a threshold filter applied to compare the final output with the stored patterns. In this way the output from the network is allowed to change gradually at each iteration, accumulating changes until stability is reached.

3. Proof of convergence

Under the gradient back-propagation algorithm, each parameter of the network, generically called Q, changes after each iteration by

$$\Delta Q = -\eta \frac{\partial E}{\partial Q} \tag{3}$$

where E is the cost function in equation (2). We shall use throughout the notation in figure 2, and note that in our formulation we have included a set of weights which connect the nodes in the input and output layers directly, in parallel with the usual multilayered perceptron (Rumelhart and McLelland 1986, Ackley *et al* 1985). We shall use the indices i, j and k to label the input, hidden and output nodes respectively.

Consider

$$|t_k^\alpha - O_k^\alpha| = \varepsilon_k^\alpha \tag{4}$$

and assume that the back-propagation algorithm is converging toward the global minimum of equation (2) where the errors corresponding to all nominated patterns are equal to zero. The target values are taken to be precisely 0 to 1, although the conclusions apply also to target values of δ and $1-\delta$ with δ sufficiently small, provided that ε is replaced by $\varepsilon' = \varepsilon + \delta$.

Let

$$\varepsilon = \max_{k,\alpha} (\varepsilon_k^\alpha). \tag{5}$$

Once the parameters of the network are fixed through using the back-propagation algorithm to achieve a small value of ε, the input states are mapped onto the output states through a vector function $\boldsymbol{G}(\boldsymbol{x})$. For a network with one layer of intermediate nodes, and putting $f(x) = 1/(1+\exp(-x))$, we have

$$G_k(\boldsymbol{x}) = f\left[\theta_k + \sum_i w_{ik}x_i + \sum_j w_{jk} f\left(\sum_i w_{ij}x_i + \theta_j\right)\right]. \tag{6}$$

Let us consider the successive iterations of an input pattern $\boldsymbol{x}$ through the network

$$\begin{aligned} &\boldsymbol{G}: && D \subset \mathbb{R}^N \to \mathbb{R}^N \\ &\boldsymbol{G}: && \boldsymbol{x}(r) \to \boldsymbol{x}(r+1) = \boldsymbol{G}(\boldsymbol{x}(r)) \qquad r = 0, 1, 2, \ldots \end{aligned} \tag{7}$$

where D denotes the unit hypercube in $\mathbb{R}^N$. A theorem due to Ostrowski states that if $\boldsymbol{G}$ has a fixed point

$$\boldsymbol{x}^* \in D: \quad \boldsymbol{x}^* = \boldsymbol{G}(\boldsymbol{x}^*) \tag{8}$$

such that the spectral radius (maximum eigenvalue) ρ of the Jacobian matrix $\boldsymbol{G}'$ with elements $\partial G_k/\partial x_i$ obeys the relation

$$|\rho[\boldsymbol{G}'(\boldsymbol{x}^*)]| < 1 \tag{9}$$

then $\boldsymbol{x}^*$ is a point of attraction for the iterative process defined by equation (7) with a finite basin of attraction in $\mathbb{R}^N$. $\boldsymbol{G}'(\boldsymbol{x}^*)$ is easily constructed using the chain rule, to yield

$$\partial G_k/\partial x_i|_{x=x^*} = O_k^*(1-O_k^*)\left[\sum_j w_{jk}O_j^*(1-O_j^*)w_{ij} + w_{ik}\right]. \tag{10}$$

The nominated patterns $\boldsymbol{I}^\alpha$ represent fixed points of $\boldsymbol{G}$ by construction. Therefore, their coordinates obey the following set of coupled equations:

$$\begin{aligned} &\sum_i w_{ik}I_i^\alpha + \sum_j w_{jk}O_j^\alpha + \theta_k = \ln[I_k^\alpha/(1-I_k^\alpha)] \\ &\sum_i w_{ij}I_i^\alpha + \theta_j = \ln[O_j^\alpha/(1-O_j^\alpha)]. \end{aligned} \tag{11}$$

These equations are, in general, underdetermined, the more so if their solution is to yield a genuinely distributed representation of the data i.e., one where the corruption of a small number of connections will not significantly alter the fixed points of the system.

The heuristic basis of the proof is that as $I_k^\alpha = \varepsilon \to 0$ or $I_k^\alpha = 1 - \varepsilon \to 1$, the size of the weights and bias terms is regulated by the logarithms in equations (11) and consequently

$$\left.\begin{matrix} O_k^*(1-O_k^*)w_{ik} \\ O_k^*(1-O_k^*)w_{jk} \end{matrix}\right\} \to O_k^*(1-O_k^*)\ln[I_k^\alpha/(1-I_k^\alpha)] \to 0. \tag{12}$$

The spectral radius of $\boldsymbol{G}'$ will then satisfy inequality (9), and hence the nominated patterns, in their realisation as self-replicating states with excitations near the saturation values of the sigmoid function will be stable in $\mathbb{R}^N$, with non-vanishing basins of attraction. Therefore, it is the flatness of the sigmoid function near its saturation values which ensures that the fixed points in $\mathbb{R}^N$ are stable under multiple iterations of the network without any need for thresholding after each separate iteration. This result is illustrated in figure 3 for a network with one input and one output node.

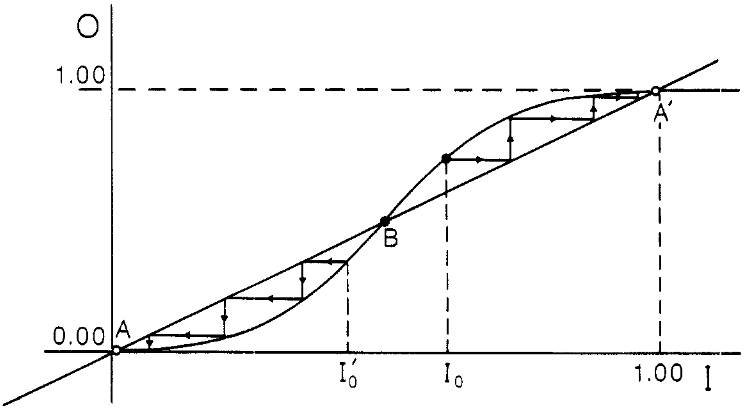

Figure 3. Plot of the output O against the input I for a network with a single input and a single output node acting as an associative memory. The two fixed points A, A′, where the slope of the response function is less than 1, are attractors, while the fixed point B, where the slope of the response function is greater than 1, is a repellor. Two starting inputs, I_0 and I_0', are shown.

It is nevertheless easy to show that if equations (11) are underdetermined, one can construct unstable solutions for which inequality (9) is not obeyed. It remains to prove that such unstable solutions cannot be reached through the gradient back-propagation algorithm for a multilayered network with direct top-to-bottom connections when the nominated patterns have components near enough to 0 or 1.

We first estimate the change in the weights after just one iteration of the back-propagation algorithm, with gain η:

$$|\Delta w_{ik}| \leqslant \eta \sum_\alpha |t_k^\alpha - O_k^\alpha| O_k^\alpha(1-O_k^\alpha) I_i^\alpha \leqslant \eta P \varepsilon^2 \tag{13}$$

$$|\Delta w_{jk}| \leqslant \eta \sum_\alpha |t_k^\alpha - O_k^\alpha| O_k^\alpha(1-O_k^\alpha) O_j^\alpha \leqslant \eta P \varepsilon^2 \tag{14}$$

$$\begin{aligned} |\Delta w_{ij}| &\leqslant \eta \sum_{k,\alpha} |t_k^\alpha - O_k^\alpha| O_k^\alpha(1-O_k^\alpha)|w_{jk}| O_j^\alpha(1-O_j^\alpha) I_i^\alpha \\ &\leqslant \frac{\eta P \varepsilon^2}{4} \sum_k |w_{jk}| \end{aligned} \tag{15}$$

where P represents the total number of nominated patterns.

Similarly for the bias terms:

$$|\Delta\theta_k| \leqslant \eta P\varepsilon^2 \tag{16}$$

$$|\Delta\theta_j| \leqslant \frac{\eta P\varepsilon^2}{4}\sum_k |w_{jk}|. \tag{17}$$

Next we must estimate a lower bound for the rate of change of the cost function in equation (2). In the limit of small gain, the discrete steps can be replaced by a continuous evolution in time, $\eta \to \mathrm{d}\eta$. We obtain

$$\begin{aligned}\frac{\mathrm{d}E}{\mathrm{d}\eta} &= \sum_{\text{all } w}\frac{\partial E}{\partial w}\frac{\mathrm{d}w}{\mathrm{d}\eta} + \sum_{\text{all }\theta}\frac{\partial E}{\partial \theta}\frac{\mathrm{d}\theta}{\mathrm{d}\eta}\\ &= -\sum_{\text{all } w}\left(\frac{\partial E}{\partial w}\right)^2 - \sum_{\text{all }\theta}\left(\frac{\partial E}{\partial \theta}\right)^2.\end{aligned} \tag{18}$$

Considering the weights connecting each input node to the corresponding output node and the output layer biases, we obtain

$$\left|\frac{\mathrm{d}E}{\mathrm{d}\eta}\right| \geqslant \sum_i \left\{\left(\sum_{\alpha_+}(1-O_i^\alpha)^2 O_i^\alpha\right)^2 + \left(\sum_{\alpha_+}(1-O_i^\alpha)^2 O_i^\alpha - \sum_{\alpha_-} O_i^{\alpha^2}(1-O_i^\alpha)\right)^2\right\} \tag{19}$$

where the sums run over the states α_+ with unit excitation at the relevant output and the states α_- with zero excitation respectively. Assuming that all outputs converge to their target values we can find for a given λ $(0<\lambda<1)$ a certain iteration of the gradient back-propagation algorithm such that for all subsequent iterations we have

$$\left|\frac{\mathrm{d}E}{\mathrm{d}\eta}\right| \geqslant \lambda \sum_i \left\{\left(\sum_{\alpha_+}(\varepsilon_i^\alpha)^2\right)^2 + \left(\sum_{\alpha_+}(\varepsilon_i^\alpha)^2 - \sum_{\alpha_-}(\varepsilon_i^\alpha)^2\right)^2\right\}. \tag{20}$$

Using the inequalities

$$x^2 + (x-y)^2 \geqslant \tfrac{1}{5}(x+y)^2 \qquad \text{and} \qquad \left(\sum_{i=1}^{N} x_i\right)^2 \leqslant N\sum_i (x^i)^2 \tag{21}$$

we obtain a lower bound for the rate of change of the cost function

$$\left|\frac{\mathrm{d}E}{\mathrm{d}\eta}\right| \geqslant KE^2 \qquad \text{where } K = \frac{4\lambda}{5N}. \tag{22}$$

From relations (14) and (22) we conclude that

$$\left|\frac{\mathrm{d}w_{jk}}{\mathrm{d}E}\right| = \frac{|\mathrm{d}w_{jk}/\mathrm{d}\eta|}{|\mathrm{d}E/\mathrm{d}\eta|} \leqslant \frac{P\varepsilon^2}{KE^2} \leqslant \frac{2P}{KE}. \tag{23}$$

It follows that

$$|w_{jk}| - |w_{jk}^0| \leqslant \left|\int_{w_{jk}^0}^{w_{jk}} \mathrm{d}|w_{jk}|\right| \leqslant \frac{2P}{K}\ln(E_0/E) \tag{24}$$

and similarly for w_{ik} and θ_k.

The relationship described by relation (24) therefore allows an estimate to be obtained for the maximum change in the value of the weights to the output layer, in terms of reference values for the weights and cost function, namely w_{jk}^0 and E_0.

These values must be frozen after the onset of the perturbative regime leading to final convergence to the global minimum of the cost function. We conclude that

$$\begin{aligned}|w_{ik}| &\leqslant \frac{2P}{K}\ln(R_1/E)\\ |w_{jk}| &\leqslant \frac{2P}{K}\ln(R_2/E)\\ |\theta_k| &\leqslant \frac{2P}{K}\ln(R_3/E)\end{aligned} \tag{25}$$

for some constants R_l, $l = 1, 2, 3$.

In order to complete our analysis of the terms in the Jacobian given in equation (10) we need also to analyse the weights to the intermediate layer.

From relations (15), (2), (4) and (25) we obtain

$$\left|\frac{\mathrm{d}w_{ij}}{\mathrm{d}\eta}\right| \leqslant \frac{P^2NE}{K}\ln(R_2/E). \tag{26}$$

Therefore, by relation (22)

$$\left|\frac{\mathrm{d}w_{ij}}{\mathrm{d}E}\right| \leqslant \frac{P^2N}{K^2E}\ln(R_2/E) \tag{27}$$

whence

$$|w_{ij}| \leqslant \frac{P^2N}{2K^2}[\ln^2(R_2/E)+\Lambda] \tag{28}$$

where Λ is a fourth reference constant.

All of the terms in equation (10) can now be estimated, giving

$$\left|\frac{\partial O_k^\alpha}{\partial I_i^\alpha}\right| \leqslant \frac{\varepsilon(1-\varepsilon)P}{2K}\left\{\ln(R_1/E)+\frac{MP^2N}{2K^2}\ln(R_2/E)[\ln^2(R_2/E)+\Lambda]\right\}. \tag{29}$$

The second term is the effect of the hidden layer, therefore it is scaled by the number of nodes there, M.

We readily conclude that

$$\left|\frac{\partial O_k^\alpha}{\partial I_i^\alpha}\right| \xrightarrow[E\to 0]{} 0. \tag{30}$$

Hence all elements in the Jacobian tend to zero and inequality (9) is eventually satisfied after sufficient steps of the algorithm. This completes our proof.

In practice, this result means that the nominated patterns are stable under multiple iterations of a multilayered perceptron with feedback provided that the real valued components of the output are taught close enough to the saturation levels of the sigmoid, 0 and 1. In our simulations we have used target values of 1.0 and 0.0 and taught to an accuracy per component per input pattern of 0.05.

The proof requires that the cost function converges to zero at a rate given by relation (22). In practice, the worst case condition applies—i.e. the equality—asymptotically as $\varepsilon\to 0$. This was verified numerically for a variety of different auto- or hetero-associative tasks, with or without top-down connections, and trained initially with or without noisy inputs.

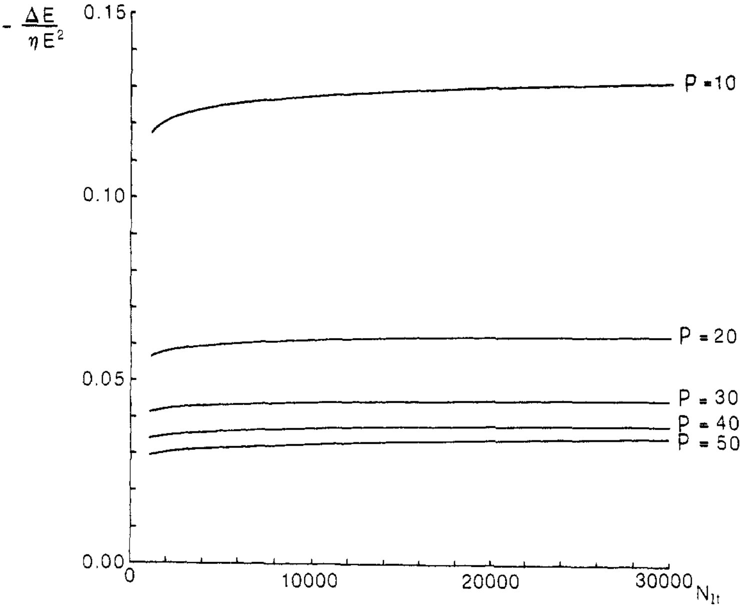

Figure 4. The quantity $\sigma = -\Delta E/\eta E^2$ plotted against the number of iterations $N_{\rm It}$ of the gradient back-propagation algorithm for auto-associative tasks with different numbers P of nominated patterns. The network has $N = 32$ input and output nodes, $M = 16$ intermediate nodes and direct top-to-bottom connections. In all cases σ approaches a non-zero value as $N_{\rm It} \to \infty$. As P increases, this value approaches $4/5N = 0.025$ (equation (22)).

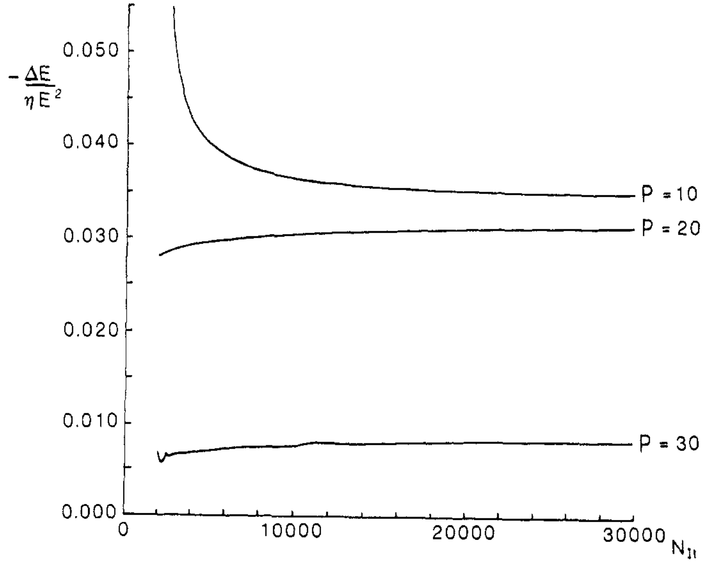

Figure 5. As in figure 4, but the network has no direct connections and is trained with noisy inputs for an initial period of 2000 iterations (not shown in the graph). A similar behaviour to that of figure 4 is observed as $N_{\rm It} \to \infty$.

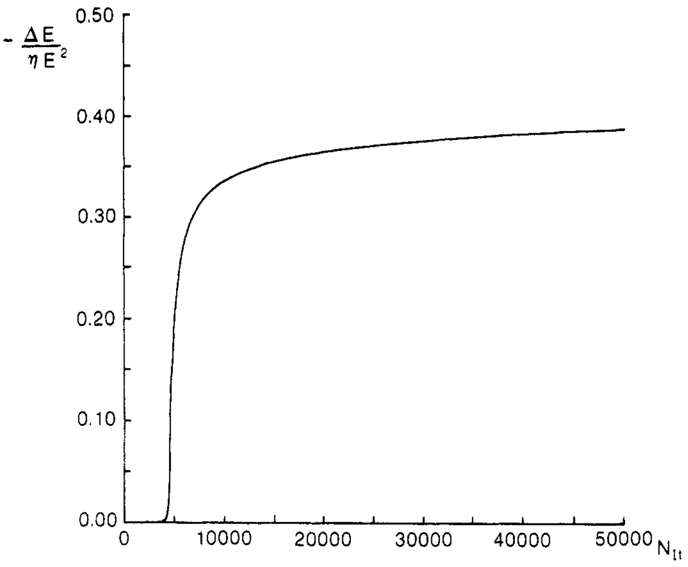

Figure 6. The quantity $\sigma = -\Delta E/\eta E^2$ plotted against the number of iterations of the gradient back-propagation algorithm for a network with two input, two intermediate and one output node which solves the exclusive OR problem. Similar behaviour to that of figures 4 and 5 is observed as $N_{It} \to \infty$.

The numerical results are presented in figures 4–6 respectively, for auto-associativity with the external connections, and using the ordinary two-layered perceptron, and for the exclusive OR problem. These results demonstrate some of the insight into the nature of the convergence of the multilayered perceptron using the back-propagation algorithm which arose during the construction of our earlier argument.

Finally, note that the proof given here applies also to fully connected single-layer networks, and it is easily extended to an arbitrary number of layers and nodes per layer provided a set of external weights exists in parallel with the multilayered perceptron. The numerical results indicate that the convergence result applies also in the absence of the external connections, and all that is required to complete the proof in this case is to show analytically the validity of relation (22).

4. Performance of the associative memory

We begin our investigation by studying the properties of the single-layer perceptron under recursive associative memory recall. Figure 7 shows the ratio of correctly recalled patterns over the total number of recall trials as a function of α, which is the ratio of the number of nominated patterns to the number of input nodes. The figures shown involve recalling the nominated patterns with 20% of their pixels reversed at random. Each point represents an average over 50 presentations of each nominated pattern corrupted with noise, averaged also over five separate training runs each with a fresh set of nominated random patterns.

The performance of the single layer perceptron compares favourably with the results from a study by Forrest (1988) of the Hopfield model with symmetric synapse weights. Forrest uses a training algorithm which ensures very strong alignment of the spin

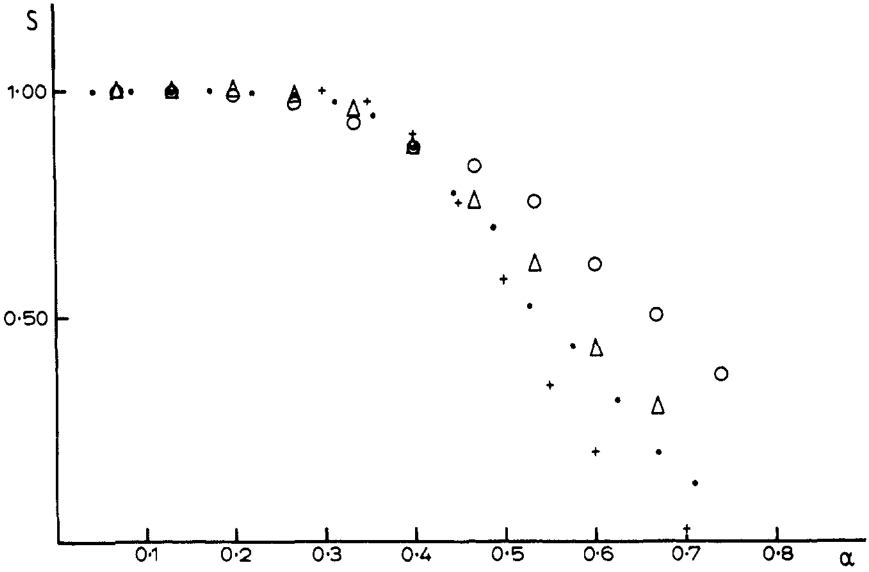

Figure 7. The success rate of recall against α (the ratio of the number of patterns over the number of output nodes N) for a network without hidden nodes, for: $N = 15$ (○); 30 (△); 45 (•); 100 (+). The nominated patterns are presented with 20% of the pixels flipped at random.

variables to their local fields and in that sense is similar to our training algorithm of the single layer perceptron. For example, for 20% of the pixels reversed (corresponding to an overlap of 0.6 in Forrest 1988), the biggest network considered (100 nodes) shows a 60% success rate during recall for $\alpha = 0.5$ compared with 1% for a 256 node net with the strongest spin-local field alignment attempted by Forrest. The non-symmetrical synapse connections in our model may account for this advantage.

Next, we characterise the performance of two-layered networks, now for a range of α between 0 and 2. We considered a network with 50 input and output nodes and investigated the effect of adding hidden nodes and external connections, and training with noise. The results are presented in figure 8. This time, the success rate during recall concerns nominated patterns presented with 10% and 20% of the pixels corrupted. The statistics are the same as before. Note that the network with hidden nodes and direct top-to-bottom connections performs only marginally better than the single layer network. This is not surprising. Indeed, we have found that deleting the connections to the hidden nodes results in a network which is still capable of reproducing a recognisable version of the taught patterns, in the sense that the original high and low bits are recalled correspondingly above and below 0.5. It is clear that the external connections are short circuiting the hidden nodes when the network is trained simply to auto-associate.

The performance improves when the network is trained with the external connections deleted, but most of all when it is trained with noise (Le Cun 1985, Wallace 1987). This conclusion is in accordance with published results (Fogelman Soulie *et al* 1987; Gallinari, Fogelman Soulie and Thiria 1987). Our training schedule involved a small number of iterations without noise which were used to 'anchor' the fixed attractors of the system to the nominated patterns following a long period of training with noise. In general, we find that the rate of success is an increasing function of the number of

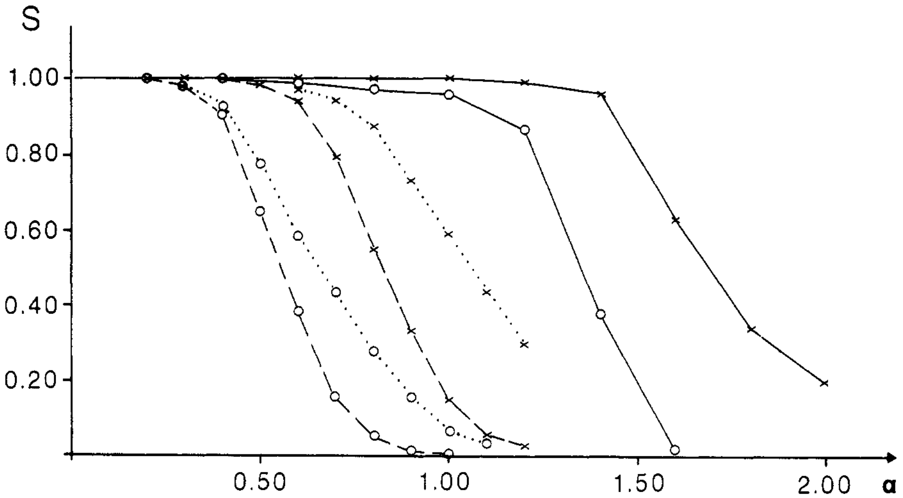

Figure 8. The success rate of recall for networks with 50 input and 50 output nodes as a function of α. Recall takes place from noisy versions of the nominated patterns with 5 (×) and 10 (○) pixels flipped. Broken lines: hidden 50-node layer with direct top-to-bottom connections, training without noise. Dotted lines: hidden 50-node layer with no direct connections, training without noise. Full lines: hidden 50-node layer with direct connections, fully connected, training with 2000 iterations of noisy inputs.

intermediate nodes and of the period of training with noise, as illustrated in figures 9 and 10. We also find that networks with top-to-bottom connections generally learn in a smaller number of iterations than networks without them. Moreover, networks without direct connections often encounter local minima of the cost function during training, especially in the region $\alpha \geqslant 1.2$. The performance of networks without direct connections for relatively small values of α ($\alpha \leqslant 1.0$) has been studied by Fogelman Soulie *et al* (1987). Here we concentrate on networks with an equal number of input,

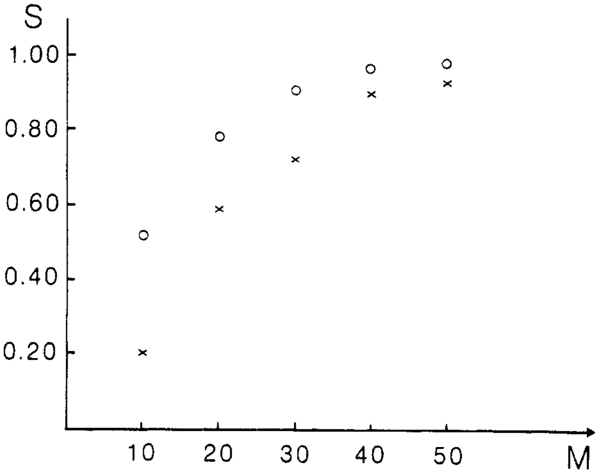

Figure 9. Success rate for recall as a function of the number M of hidden nodes for a network with a 50-M-50 architecture and direct top-to-bottom connections trained with noisy inputs (10 pixels flipped, 2000 iterations), for: $\alpha = 0.8$ (○); $\alpha = 1.0$ (×).

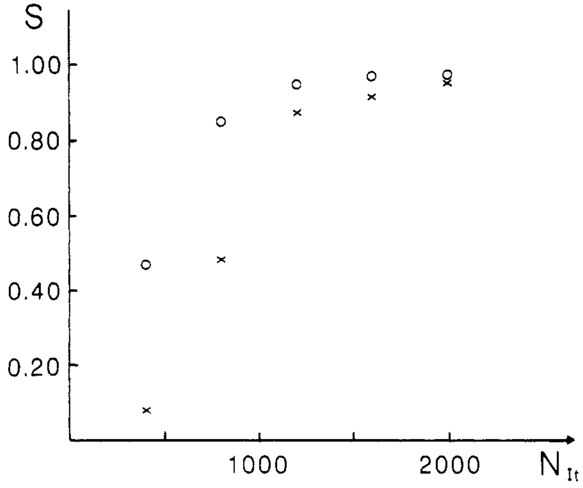

Figure 10. Success rate for recall as a function of the number $N_{\rm It}$ of training cycles with noisy inputs. The results are from networks with a 50-50-50 architecture and direct connections. The noise during training and recall corresponds to 10 flipped pixels, for: $\alpha = 0.8$ (○); $\alpha = 1.0$ (×).

intermediate and output nodes and direct top-to-bottom connections and train them with a relatively large number of noisy iterations (2000) in the simulations to follow. We show the performance in figure 8 as a function of α. Note that training with noise applied to a two-layered network with full external connections provides non-trivial basins of attraction even at $\alpha = 2.0$, which is the limit of the capacity of single layer networks (Gardner and Derrida 1987, Baldi and Venkatesh 1987).

It is clear that teaching with noise puts the intermediate nodes to good use. In fact the excitations of the hidden nodes in response to the nominated patterns tend to approach the critical values of 0 and 1 more often than when training without the use of noise. Further numerical experiments using structured data suggest that the basins of attraction thus produced are not only larger, as evidenced by the results in figure 8, but also more isotropic, the more so the larger the number of nodes in the intermediate layer. Further evidence of the increased role of the intermediate nodes is that it is no longer possible to delete the intermediate connections after training and still expect the system to be able to reproduce the stored patterns at all.

It is instructive also to compare the performance of the recursive associative algorithm with real valued and with binary output. The results shown in figure 11 indicate that the gradual changes in the network output after each iteration accumulating over a large number of iterations plays a role in improving the performance of the real valued memory over that of the binary valued network, where spin flips must always take place in a single iteration.

We conclude our performance evaluation with a brief discussion of the ability of networks to retrieve taught patterns from uncertain information about them (pattern completion task). If a fraction of the pixels is not known, we set the corresponding inputs to the intermediate value 0.5 and iterate through the network as before. A three-state variant of the Hopfield model designed to solve the pattern completion problem has been studied numerically (Meunier *et al* 1988) and shown to achieve high success rates in the region of $\alpha < 0.15$ for the pattern completion problem with a

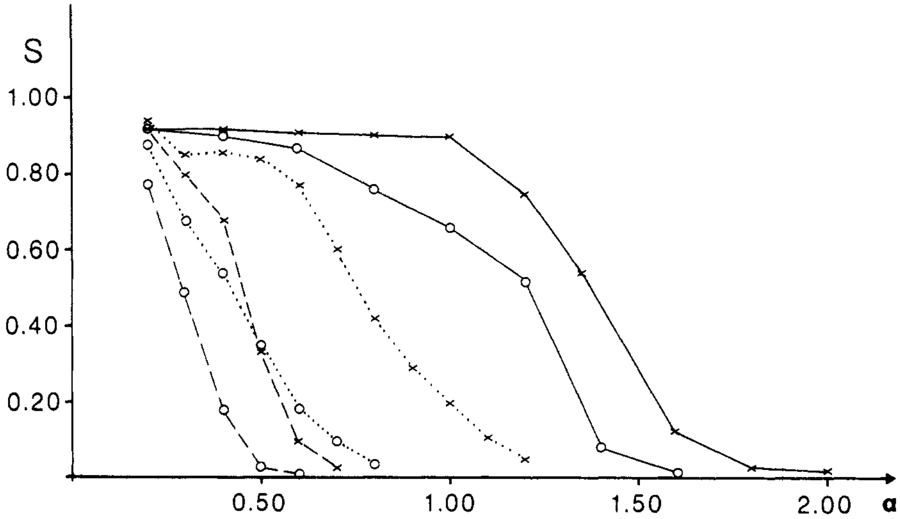

Figure 11. Success rate for recall from noisy versions of the nominated patterns as a function of α for networks with 50 input and 50 output nodes. The network output is thresholded to integer values after each iteration of the recall algorithm. The notation is the same as in figure 8.

fraction of uncertain pixels of up to 0.8. In figure 12(a) we show the recall success rate for the augmented multilayered perceptron, trained by back-propagation. Adding an intermediate layer of nodes and training with incomplete versions of the nominated patterns, again, greatly improves the performance. Note that a success rate larger than 95% is achieved even when most of the pixels (70%) of a nominated pattern are lost for $\alpha \leqslant 0.6$. When a small proportion (10%) of the patterns is lost, high success rates are achieved even for $\alpha \simeq 2.0$. Training specifically for the recall task is obviously an expedient way of improving the performance of a neural network, although this does not necessarily indicate any amount of generalisation of the acquired memory to different, even if related, tasks. Hence, training with noise does not generally significantly improve the performance in the pattern completion problem, and vice-versa.

With this proviso, the improvement in the pattern completion task is more pronounced for larger fractions of uncertain pixels. Figure 12(b) also illustrates that if the output during recall is thresholded after each iteration, then information is irretrievably lost with a consequent deterioration of the capacity of the network. In this sense, there are tasks where the real valued nature of the network output, and its convergence after multiple iterations, play an important part.

5. Conclusion

The convergence of recursive associative memories consisting of a multilayered perceptron augmented with an external set of weights was proved. The results of numerical simulations indicate that the quantitative behaviour of these networks during learning and their consequent stability under recursion are more general and apply to all multilayered perceptrons with arbitrary connectivity or topology.

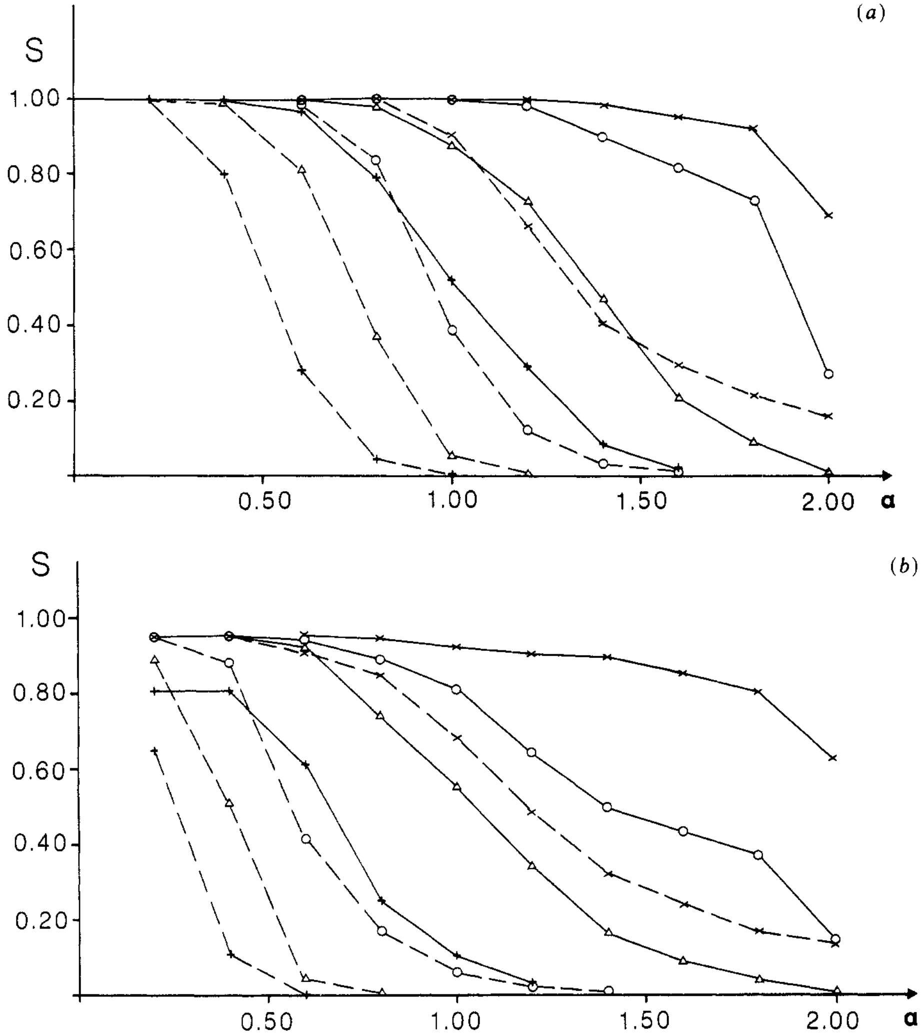

Figure 12. (*a*) The success rate of recall from incomplete versions of the nominated patterns with a fraction r_1 of their pixels missing for networks with 50 input, 50 intermediate and 50 output nodes. The full lines denote training with 2000 incomplete versions of the nominated patterns with a fraction r_2 of their pixels missing, while the broken lines correspond to training with the nominated patterns themselves. (×) $r_1=0.10$, $r_2=0.30$; (○) $r_1=0.30$, $r_2=0.30$; (△) $r_1=0.50$, $r_2=0.50$; (+) $r_1=0.70$, $r_2=0.50$. (*b*) As in (*a*) but the output is thresholded after each iteration of the recall algorithm.

The performance of these associative memories is improved when training with noise is used, which is in agreement with known results. This form of training was shown to enhance the role of the nodes in the intermediate layer and it renders the capacity of networks augmented with even a fully connected set of external weights completely non-trivial.

The capacity of the single-layer network compares favourably with that of the iterative Hopfield model. The performance of the two-layered network, with external connections, was found to exceed the theoretical limit for the capacity of single-layer networks only when noise is applied during training.

Further work is in progress to determine whether there are any limitations to the capacity of multilayered recursive associative memories trained by the back-propagation algorithm, when non-trivial basins of attraction are required.

Acknowledgments

We wish to thank the late Elizabeth Gardner for valuable discussions. SJP acknowledges the financial support of SERC and is thankful to Chris Michael for the use of his grant of computer time on the RAL CRAY.

References

Ackley D H, Hinton G E and Sejnowski T J 1985 *Cog. Sci.* **9** 147

Baldi P and Venkatesh S 1987 *Phys. Rev. Lett.* **58** 913

Fogelman Soulie F, Gallinari P, Le Cun Y and Thiria S 1987 *Proc. 1st IEEE Conf. on Neural Networks (San Diego, 1987)* vol II (Piscataway, NJ: IEEE) p 653

Forrest B M 1988 *J. Phys. A: Math. Gen.* **21** 245

Gallinari P, Le Cun Y and Thiria S 1987 *Automata Networks in Computer Science, Theory and Applications* (Manchester: Manchester University Press/Princeton, PA: Princeton University Press) pp 133–86

Gardner E and Derrida B 1988 *J. Phys. A: Math. Gen.* **21** 271

Hopfield J J 1982 *Proc. Natl Acad. Sci. USA* **79** 2554

Le Cun Y 1985 *Proc. Cognitiva '85* p 599

Lippman R P 1987 *IEEE ASSP Mag.* **4** 4

Meunier C, Hansel D and Verga A 1988 Information processing in three-state neural networks *Preprint* Ecole Polytechnique A865.1188

Rumelhart D E, Hinton G E and Williams R J 1986 *Nature* **323** 533

Rumelhart D E and McLelland J L 1986 *Parallel Distributed Processing* (Cambridge, MA: MIT Press)

Wallace D 1987 *Computational Physics (Scottish Universities Summer School in Physics 32)* ed R D Kenway and G S Pawley (Edinburgh: SUSSP Press)

Wieland A and Leighton R 1987 *Proc. 1st IEEE Conf. on Neural Networks (San Diego, 1987)* vol III (Piscataway, NJ: IEEE) p 385