Physics Letters A 170 (1992) 95–98
North-Holland

PHYSICS LETTERS A

Maximum overlap neural networks for associative memory

E. Elizalde, S. Gómez and A. Romeo [1]

Departament d'Estructura i Constituents de la Matèria, Facultat de Física, Universitat de Barcelona, Diagonal 647, 08028 Barcelona, Spain

Received 29 April 1992; revised manuscript received 7 July 1992; accepted for publication 28 August 1992
Communicated by A.P. Fordy

The possibility of achieving optimal associative memory by means of multilayer neural networks is explored. Three original different solutions which guarantee maximal basins of attraction and storage capacity are found and their main characteristics are outlined.

The basic problem of associative memory is to store a set of binary patterns $\{\xi^\mu\}_{\mu=1,\dots,p}$ in such a way that the network, when presented with an input pattern ξ, retrieves the stored one which is closest to it. The distance between two patterns ξ and ξ^μ is given by their mutual overlap

$$O^\mu(\xi) \equiv \sum_{k=1}^{N} [\xi_k^\mu \xi_k + (1-\xi_k^\mu)(1-\xi_k)] , \qquad (1)$$

which grows as ξ and ξ^μ resemble each other more and more – for mathematical convenience the activation values of the input units have been supposed to be 0 and 1.

Fully connected artificial neural networks achieve this task pretty well if proper interconnection strengths are chosen [1–3]. For instance, the Hebb rule works better if the patterns to be retained are uncorrelated, while the pseudo-inverse rule [4] is preferable if this is not the case. The quality of a particular performance is measured by the size of the basins of attraction: the wider they are, the larger the number of inputs that will access memories in the right way. We consider as *optimal* those solutions for which the pattern retrieved is always the nearest stored one, at most with the exception of those inputs equidistant from two or more of them. Indeed, optimality is equivalent to the possession of the largest possible basins of attraction. In the present Letter we are going to show how multilayer neural networks may be used as multilayer associative memory optimal networks (MAMONets).

The underlying idea behind our solution is the parallel computation of all the overlaps $O^\mu(\xi)$, followed by the identification of the largest one – say $O^\alpha(\xi)$ – and the reconstruction of the pattern ξ^α itself, where the whole process is implemented through ordinary units distributed in layers. The first of these steps has already been taken in ref. [5]. However, the authors of ref. [5] assume the existence of activation functions capable of finding the largest of two numbers and of picking its index, thus sidestepping the harder problem of doing so by means of "available" types of neurons.

First we have to give some meaning to the expression "recognition of the index". The easiest α-index neural representation is the unitary one: $S^\mu(\xi) = \kappa\delta_\alpha^\mu$, $\mu = 1, \dots, p$, $\kappa > 0$, where δ_α^μ is the usual Kronecker symbol. After some work, the following three different schemes have been found to be suitable procedures for this index identification.

(1) *Binary units.* Considering an intermediate layer of units

$$\sigma^{\mu\nu}(\xi) = \Theta(O^\mu(\xi) - O^\nu(\xi))$$

$$= \Theta\left(\sum_{k=1}^{N} \omega_k^{\mu\nu}\xi_k - \theta^{\mu\nu}\right), \quad \mu < \nu , \qquad (2)$$

[1] Present address: Dipartimento di Fisica, II Università di Roma, Via E. Carnevale, I-00173 Rome, Italy.

where $\omega_k^{\mu\nu}=2(\xi_k^\mu-\xi_k^\nu)$ and $\theta^{\mu\nu}=\sum_{k=1}^N(\xi_k^\mu-\xi_k^\nu)$, Θ being the logistic function

$$\Theta(x)\equiv 0 \quad \text{if } x\leqslant 0\,,$$
$$\equiv 1 \quad \text{if } x>0\,, \tag{3}$$

it is easy to show that

$$S^\lambda(\boldsymbol{\xi})=\Theta\Bigg(\sum_{\substack{\mu,\rho\\ \mu<\rho}}\omega^{\lambda,\mu\rho}\sigma^{\mu\rho}(\boldsymbol{\xi})-\theta^\lambda\Bigg) \tag{4}$$

does the job nicely if the weights and thresholds are given by

$$\omega^{\lambda,\mu\rho}=\delta^{\lambda\mu} \quad \text{if } \rho>\lambda\,,$$
$$=-\delta^{\lambda\rho} \quad \text{if } \mu<\lambda\,,$$
$$\theta^\lambda=p-\lambda-\epsilon\,,\quad 0<\epsilon<1\,. \tag{5}$$

Unfortunately, the number of units grows quadratically with the number of patterns. The other drawback is its inability to cope with input patterns which are equidistant from two or more of the $\boldsymbol{\xi}^\mu$'s.

(2) *Decreasing thresholds.* If we knew in advance the value of $O^\alpha(\boldsymbol{\xi})$, it would suffice to choose a common threshold $\theta=O^\alpha(\boldsymbol{\xi})-\epsilon$, $0<\epsilon<1$, and compute

$$S^\mu(\boldsymbol{\xi})=\Theta(O^\mu(\boldsymbol{\xi})-\theta)$$
$$=\Theta\Bigg(\sum_{k=1}^N(2\xi_k^\mu-1)\xi_k-\Bigg(\sum_{k=1}^N\xi_k^\mu-N+\theta\Bigg)\Bigg)\,. \tag{6}$$

Since this is not the case, it is possible to use a threshold large enough and decrease it by time steps, until one of the overlaps be above it and the rest be below. Taking into account that, according to definition (1), all the overlaps can only be integers between 0 and N, a good threshold to start with is

$$\theta(0)=N-\epsilon\,,\quad 0<\epsilon<1\,. \tag{7}$$

At every step the same input pattern will be reprocessed, i.e.

$$\boldsymbol{\xi}(t+1)=\boldsymbol{\xi}(t)\,, \tag{8}$$

and an additional unit, say c, will take care of checking whether the end condition is satisfied or not. We define the state of this control unit as

$$c(\boldsymbol{S})=\Theta\Bigg(\sum_{\mu=1}^p S^\mu\Bigg)\,. \tag{9}$$

Therefore, the updata rule for the variable threshold must be

$$\theta(t+1)=\theta(t)-[1-c(\boldsymbol{S})]\,. \tag{10}$$

When $c=1$, θ repeats its previous value and the network becomes stable.

Unlike the previous scheme, this set-up allows for the recognition of all the stored patterns at minimal distance from the input $\boldsymbol{\xi}$, even if $O^\alpha(\boldsymbol{\xi})$ is not unique.

(3) *Quasilinear units.* The so-called MaxNet algorithm was conceived for the purpose of picking winning units in neuron clusters for competitive learning [6]. We suggest here to use it as the way of finding the largest *normalized* overlap $(1/N)O^\alpha(\boldsymbol{\xi})$. Taking the activation to be the *quasilinear function*

$$f(x)=0 \quad \text{if } x<0\,,$$
$$=x \quad \text{if } 0\leqslant x\leqslant 1\,,$$
$$=1 \quad \text{if } x>1\,, \tag{11}$$

we initialize the state of a fully interconnected layer with

$$S^\mu(t=0)=f\Bigg(\sum_{k=1}^N\frac{1}{N}(2\xi_k^\mu-1)\xi_k$$
$$-\Bigg(\frac{1}{N}\sum_{k=1}^N\xi_k^\mu-1\Bigg)\Bigg)\,, \tag{12}$$

which later on will be synchronously updated through

$$S^\mu(t+1)=f\Bigg(\sum_{\rho=1}^p\omega^{\mu\rho}S^\rho(t)\Bigg)\,, \tag{13}$$

where

$$\omega^{\mu\rho}=1 \quad \text{if } \rho=\mu\,,$$
$$=-\epsilon \quad \text{if } \rho\neq\mu\,,$$
$$0<\epsilon\leqslant 1/(p-1)\,. \tag{14}$$

$\boldsymbol{S}(t)$ reaches a stable configuration if the largest overlap is unique, taking less than $T\equiv\log N/\log(1+\epsilon)+1$ iterations. Otherwise, $\boldsymbol{S}(t\geqslant T)$ singles out the indices of all the $\boldsymbol{\xi}^{\alpha'}$s.

Once the α-index has been identified by means of a unary representation in a layer $\boldsymbol{S}$, the "visual" retrieval of $\boldsymbol{\xi}^\mu$ is trivially achieved with the help of a new layer

Table 1
Average frequencies for the simulation corresponding to the example $N=10$, $p=5$, by the four methods quoted in the text. The numbers of iterations listed were those necessary for obtaining a largest relative increase below 10^{-3}.

	Hebb rule	Q^{-1} method	AdaTron	MAMONet
global retrieval	186.39	124.85	137.05	709.04
spurious	434.84	899.14	302.61	43.71 fake
unstable	402.78	0.01	584.35	271.25 hesitant
number of iterations	1002	1002	1003	333

Table 2
Global retrieval rates obtained by each procedure for $N=10$ and for different values of p/N. The estimated error is 5×10^{-2} for the first three methods and less than 10^{-2} for the MAMONet figures.

p/N	Hebb rule	Q^{-1} method	AdaTron	MAMONet
0.2	0.35	0.36	0.35	0.82
0.3	0.22	0.32	0.18	0.75
0.4	0.25	0.20	0.17	0.71
0.5	0.18	0.12	0.13	0.69
0.6	0.21	0.07	0.20	0.67
0.7	0.20	0.03	0.18	0.65
0.8	0.18	0.03	0.19	0.63
0.9	0.18	0.02	0.17	0.62
1	0.19	0.02	0.18	0.61

$$\zeta_k(\xi)=\Theta\left(\sum_{\mu=1}^{p}\xi_k^\mu S^\mu(\xi)\right). \tag{15}$$

However, if there are more than one maximally overlaping patterns, the network can pick several indices. These configurations correspond to states of the sort

$$S=(0,...,0,\overset{\alpha_1}{\Delta},0,...,0,\overset{\alpha_r}{\Delta},0,...,0)\,, \tag{16}$$

where Δ is some positive constant. In this case the outcome reproduces the OR-sum of patterns $\xi^{\alpha_1},...,\xi^{\alpha_r}$, which may coincide with one of the stored ξ^μ's. Such patterns would then be *fakely* retrieved. otherwise, the result becomes a *hesitant* configuration.

In order to assess the performance of our methods, we have carried out numerical simulations of some examples comparing the third MAMONet scheme with Hopfield networks for three different prescriptions for the weights: the Hebb rule, the pseudo-inverse matrix and the optimal stability solution (the AdaTron algorithm [7]).

The result of the comparison is shown in tables 1 and 2. The first example lists the average number of fake and hesitant states produced by the MAMONet method as well as the spurious and other non-retrieval configurations arising from the Hebb rule and its variants. Table 2 displays just the different global retrieval rates obtained by the four methods and for varying numbers of stored patterns. While the Q^{-1} and AdaTron methods yield rates of the same order as the standard Hebb rule, the MAMONet figures are definitely higher in all cases as expected.

Summing up, we have provided three different MAMONet solutions to the problem of associative memory, all of them sharing unlimited storage capacity, perfect recall and the removal of spurious minima and unstable states. Their retrieval power is optimal in the sense that the network's answer is selected by the maximal overlap criterion. The original contribution of these solutions has been the realization of such designs without introducing types of units different from those currently used in most neural network architectures.

This work has been partly supported by the Dirección General de Investigación Científica y Técnica (DGICYT) project no. PB90-0022. SG and AR thank the Spanish Ministry of Education and Science for FPI and FPIE fellowships.

References

[1] J.J. Hopfield, Proc. Natl. Acad. Sci. USA 79 (1982) 2554; 81 (1984) 3088.
[2] W.A. Little, Math. Biosci. 19 (1974) 101.

[3] D. Amit, H. Gutfreund and H. Sompolinsky, Phys. Rev. A 32 (1985) 1007; Phys. Rev. Lett. 55 (1985) 1431.
[4] I. Kanter and H. Sompolinsky, Phys. Rev. A 35 (1987) 380.
[5] E. Domany and H. Orland, Phys. Lett. A 125 (1987) 32.
[6] R.P. Lippmann, IEEE ASSP Mag. (April, 1987) 4.
[7] J.K. Anlauf and M. Biehl, Europhys. Lett. 10 (1989) 687.