

A NETWORK OF MULTISTATE UNITS CAPABLE OF ASSOCIATIVE MEMORY AND PATTERN CLASSIFICATION

Wolfgang BANZHAF

Institut für theoretische Physik und Synergetik der Universität Stuttgart, Pfaffenwaldring 57/IV D-7000 Stuttgart 80, Fed. Rep. Germany

Received 8 December 1987

Revised manuscript received 29 August 1988

Communicated by R. Temam

We consider a model of multistable units acting together in a network. We modify the landscape algorithm of spinglass-like neural nets to cope with new conditions. Collective capabilities such as associative memory function or pattern classification are demonstrated using the simplest possible learning rule of Hebb.

1. Motivation

Last years have shown an enormous convergence of seemingly different research areas. Brain research and computer simulations of neural nets are bringing closer and closer together neurophysiology, cognitive science, computer science, biology and physics.

Since the early days of artificial intelligence one of the key issues is the following question: What are the fundamentals of “intelligent” behaviour of organisms and how can one realize them with the materials to the disposal of mankind?

The efforts of AI have uncovered huge problems one encounters in copying intelligent behaviour at the high level of abstraction as, say, the cognitive level. Now interest has turned again to a deeper level of natural realization: the neural level which was already studied more than twenty years ago. In present days there is substantial hope to understand intelligent behaviour by deciphering the self-organization processes going on in networks of neurons. In the fifties McCulloch and Pitts [1] have modeled neurons as all-or-none units firing if its inputs exceed a certain threshold and otherwise

being silent. These ideas were followed by the development of models of neural networks for associative memory and pattern recognition tasks [2–4]. Typical models are spinglass-like neural networks [5–10] which are in addition well suited for simulation on digital computers.

Neurophysiological studies, however, have brought up the idea of frequency coding of signals in the nervous system [11,12]. This means that neurons are working in an analog fashion rather than a digital one at least in the frequency domain. No wonder: The majority of environmental observables we can measure at macroscopic level, i.e. the level sensory neurons are receptive, are continuous-valued functions to which special neurons are optimally adapted. Thus it seems to be reasonable to study models which incorporate this aspect as well as simple dynamics known from models of 2-state neurons.

Here we want to present a network model motivated by analog coding. The units we consider can achieve one from a finite set of states. Memory function is understood as being able to store and retrieve arbitrary collections of these states lumped together to state vectors. Being not binary and not

continuous in state space this model constitutes a sort of compromise between different requirements. We study this network with respect to its pattern recognition abilities.

Section 2 outlines the general ideas whereas section 3 gives the detailed model. Section 4 shows simulations of the network done on a serial computer.

2. The general principles

The most important aspect of spinglass-like models of neural networks designed to memorize patterns x is its action in a high-dimensional landscape $E(x)$ of a generalized energy-function. Stored patterns are local minima of this function whereas its attractor basins coincide with the valleys surrounding these minima.

Recalling dynamics can be described as an overdamped motion downhill in this high-dimensional landscape. Thereby, a starting pattern x changes its form as long as it is distinct from the pattern x^α constituting the (presumably) nearest local minimum. x has recalled x^α the same way $x + \delta x$ would have recalled x^α .

Typically, this function is used as an auto-associative memory. But we want to remark here that it can be used for hetero-associative memory and even for pattern classification as well. One can see this as follows: Suppose your stored patterns x^α are composed of two parts

$$x^\alpha = \begin{pmatrix} y^\alpha \\ z^\alpha \end{pmatrix}, \quad (1)$$

then you can regard y^α as key or stimulus pattern and z^α as a corresponding response pattern. Now you constrain retrieval dynamics to the z -part, i.e. you fix y to, say, y^γ (or $y^\gamma + \delta y$), then (after a while) dynamics should give you z^γ .

In other words, dynamics has to look for a global minimum of the energy landscape under the constraints given by fixed y . Since a simple

gradient search will not do the job, a suitable dynamics has to be chosen (see below). Of course, z^γ could be the same for more than one stored pattern

$$z^\gamma = z^\beta = z^\alpha.$$

Thus, we have a data compression or classification in that – constraining dynamics to the z -part – one z may result from a set of y , $y \in \{y^\alpha, y^\beta, y^\gamma, \dots\}$, whereas another z' may result from a different set of patterns $y \in \{y^\delta, y^\epsilon, \dots\}$.

If, on the other hand, dynamics is constrained to the y -part of patterns under a given $z = z$, it depends on the starting state in the y -domain which of the previously learned patterns will be chosen. More specifically, the system will relax to that example y of the corresponding class which had the maximum overlap just from the beginning, since only motions downhill in the energy landscape are allowed by the dynamics. If two examples have the same initial overlap or an unbiased null vector $y = \theta$ is the starting state, a random fluctuation of the dynamics will decide. The entire operation may be termed “inverse” classification giving an example of the class z .

In addition, if we could relax the restriction of a binary valued state space, we would have still more possibilities. These include classification of patterns according to intrinsic criteria like e.g. the most significant feature different in all stored patterns or according to criteria totally at will. In general this will require a projection to a subspace which is not binary.

To summarize, what has to be done is to constrain dynamics to an arbitrary subspace of the state space of patterns where things additionally are going on faster.

Since spinglass-like models are good examples of associative memory based on a dynamics, it is near at hand to try to generalize the underlying principles in such a way as to ensure storage, retrieval and classification abilities of the model in a denser state-space.

To give an overview we list here what we keep from these models:

- The overdamped motion in a high-dimensional Lyapunov-function landscape downhill to local minima.
- The mechanism of storing information by changing synaptic weights (connections), i.e. by modifying the interaction between processing elements, with a linear learning rule.

Compared to spinglass-like models we change:

- The state space density: Each dimension (component of x) ranges from $-m, \dots, +m$ in integer steps, in our case $m = 5$.
- The connection matrix J_{ij} comprise additionally self-coupling terms J_{ii} .
- The dynamics will be driven by fluctuations leading to local equilibrium states.
- The kinds of motion: Any realizable constrained dynamics on subspaces will be allowed.
- New non-linearities are introduced and the processing of patterns needs non-local operations (e.g. computation of absolute values of the state variables over all components).

Let us make some comments: Clearly, noise in J_{ij} will destroy our memory capabilities at a critical value $\alpha_c = p_c/N$. Thus, the used learning rule forces us to store orthogonal prototype patterns or at least weakly correlated ones with randomly chosen components. Although this is not a real world problem, the simplest learning rule for patterns ξ^α (Hebb-like)

$$J_{ij} = \sum_{\alpha} \xi_i^{\alpha} \xi_j^{\alpha} \quad (2)$$

- which means just the superposition of the dyadic products of prototype patterns - was applied here for the sake of clarity. Another choice for a learning rule - though not considered in the present paper - is a connection matrix including the Moore-Penrose pseudoinverse which amounts to a complete orthonormalization of prototype patterns.

Fluctuation driven dynamics is chosen to study an alternative kind of motion in state space which

realizes a constraint satisfaction network (cf. [13]). It uses noise to proceed on the way downhill. One could conjecture that the brain works similarly: it presumably uses noise in a very efficient way rather than struggling against it.

Note that the Boltzmann-machine algorithm [13] was designed following the same line of thoughts. The difference may be that it was also inspired by statistical mechanics whereas our approach tends to be more like natural evolution proceeding by some kind of mutation and selection. We suppose that evolution-like algorithms can be applied very efficiently to network dynamics as well as to learning dynamics. This is, however, beyond the scope of the present paper and will be subject to another contribution [14].

The necessary change with respect to spin models was the introduction of absolute value or norm of a state. In contrast to binary models for patterns from $\{-1, +1\}$, the norm here is a non-constant variable characterizing each state vector together with its angles (corresponding to the overlaps) to the stored patterns. The former energy function

$$E(x) = - \sum_{ij} x_i J_{ij} x_j \quad (3)$$

has to be modified to compensate for the effects of J_{ij} on the length of a state vector. A new energy function can be written down as

$$E(x) \equiv - \sum_{ij} x_i J_{ij} x_j \frac{1}{|x|} \frac{1}{|Jx|}, \quad (4)$$

where

$$|x| = \sqrt{\sum_i x_i^2}, \quad (5a)$$

$$|Jx| = \sqrt{\sum_i (Jx)_i^2}. \quad (5b)$$

Computing $E(x)$ means

1. mapping x by J into the subspace L_{ξ} spanned by ξ^{α} , $\alpha = 1, \dots, p$;

Binary patterns (orthogonal to stored ones)

n-ary patterns (orthogonal to stored ones)

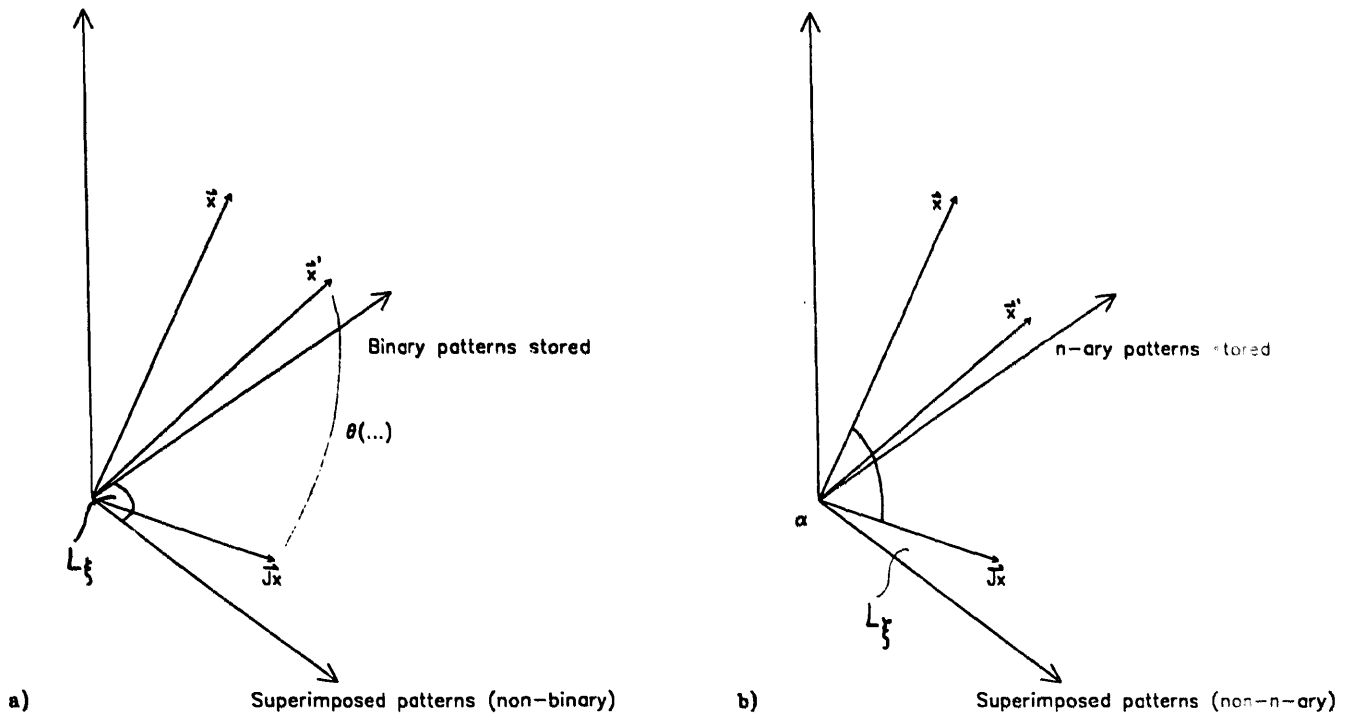


Fig. 1. Dynamical succession of two states: (a) With 2-state neurons. After projecting arbitrary binary patterns x to the subspace L_ξ of known patterns a non-linear threshold operation Θ takes place to restore the binary structure of pattern vectors. In general one can observe convergence toward the axis of stored patterns. (b) With multistate units. After projection and normalization resulting in an energy proportional to the angle α a new x is looked for which has smaller α . This pattern vector is nearer to the L_ξ -space. Search ends in the direct neighborhood of the axis of stored n -ary patterns.

2. comparison of the result Jx with the original x on the common basis of normalized vectors. As will be shown below, E has indeed minima if x is one of the stored patterns x^α .

By $|x|$ and $|Jx|$ we have introduced new non-linearities to be able to compare x and Jx . The situation can be illustrated according to fig. 1. Whereas in the binary model a simple threshold operation θ will restore the constant length of pattern vectors, a more complicated operation N of normalization is needed in non-binary models. This ensures an energy function proportional to $\cos \alpha$, α being the angle between a presented pattern and its mapping to the subspace of stored patterns.

The idea of introducing new non-linearities can be generalized and leads to a new class of dynamical models for pattern recognition [15]. Normalizing factors such as $|x|$ constitute non-local

non-linearities in the sense that the result of the Jx (mapping) operation at every site i firstly has to be (quadratically) summed over all sites i and then secondly has to be propagated back to every unit. One mechanism to perform this operation would be a second net connecting all processing elements which runs alternatively and proliferates every unit with the result of this normalization.

3. The detailed model

Now it is necessary to give a more detailed description of the behaviour of the proposed algorithm. Suppose we want to store patterns x^α , $\alpha = 1, \dots, p$, $x_i \in \{-5, -4, \dots, +4, +5\}$, $i = 1, \dots, N$, in the network. After subtraction of the average vector m , $m = (1/p)\sum_\alpha x^\alpha$ the new $x^{\alpha'}$ are normalized $\xi^\alpha = x^{\alpha'} / |x^{\alpha'}|$ and superimposed to

form the matrix

$$J_{ij} = \frac{1}{p} \sum_{\alpha} \xi_i^{\alpha} \xi_j^{\alpha}. \quad (6)$$

This matrix determines the energy landscape of eq. (4). In [16] we have shown the behaviour of the system for orthogonal patterns ξ^{α} . Here we generalize the treatment to non-orthogonal patterns by introducing correlations $\Delta^{\alpha\beta}$ between prototype patterns ξ^{α} , ξ^{β} . We show that

- i) $E(\mathbf{x})$ has local minima for $\mathbf{x} \parallel \mathbf{x}^{\alpha}$,
- ii) the \mathbf{x}^{α} are attractor states for its neighborhood.

i) Any pattern \mathbf{x} can be written as

$$\mathbf{x} = a_{\beta} \mathbf{x}^{\beta} + \mathbf{v}^{\beta}, \quad (7)$$

where \mathbf{x}^{β} corresponds to one of the stored patterns ξ^{β} , $\beta = 1, \dots, p$ and a_{β} is the scalar product $\mathbf{x}^{\beta} \cdot \mathbf{x}$. Thus, it is understood as composed of a known pattern plus some noise vector which is composed of contributions from the other known patterns and pure noise:

$$\mathbf{v}^{\beta} = \sum_{\gamma \neq \beta} a_{\gamma} \mathbf{x}^{\gamma} + \mathbf{n}. \quad (8)$$

By application of J_{ij} on \mathbf{x} , \mathbf{x} is mapped into the subspace L_{ξ} of all stored patterns. Note, however, that J_{ij} is not a projection operator in the usual sense, since its components ξ^{α} are not orthogonal. This may be cured by introduction of adjoint patterns which are appropriately defined. Let the deviation of ξ^{α} from orthogonality be formulated as

$$\frac{1}{p} \sum_i \xi_i^{\alpha} \xi_i^{\beta} = \delta^{\alpha\beta} + \Delta^{\alpha\beta}, \quad \Delta^{\alpha\beta} \ll 1. \quad (9)$$

Then the mapped state vector \mathbf{x}_J can be written as

$$\mathbf{x}_J \equiv J\mathbf{x} = a_{\beta} \mathbf{x}^{\beta} + \mathbf{h}^{\beta} + \mathbf{k}^{\beta}, \quad (10)$$

where

$$\mathbf{h}^{\beta} = J\mathbf{v}^{\beta}, \quad \text{and} \quad (11)$$

$$\mathbf{k}^{\beta} = a_{\beta} |\mathbf{x}^{\beta}| \sum_{\alpha} \xi^{\alpha} \Delta^{\alpha\beta}. \quad (12)$$

Now E reads

$$\begin{aligned} E &= - \frac{\mathbf{x}}{|\mathbf{x}|} \cdot \frac{\mathbf{x}_J}{|\mathbf{x}_J|} \\ &= - (a_{\beta} \mathbf{x}^{\beta} + \mathbf{v}^{\beta}) \cdot (a_{\beta} \mathbf{x}^{\beta} + \mathbf{h}^{\beta} + \mathbf{k}^{\beta}) \\ &\quad \times \frac{1}{|\mathbf{x}| |\mathbf{x}_J|} \geq -1, \end{aligned} \quad (13)$$

which is minimal for

$$1. \quad \mathbf{v}^{\beta} = 0, \quad \mathbf{h}^{\beta} = 0. \quad (14a)$$

$$2. \quad \mathbf{v}^{\beta} = \sum_{\gamma \neq \beta} \mathbf{x}^{\gamma} a_{\gamma} \in L_{\xi}, \quad \mathbf{h}^{\beta} = \mathbf{v}^{\beta} + \mathbf{l}^{\beta}$$

$$\text{with } \mathbf{l}^{\beta} = \sum_{\alpha\gamma \neq \beta} \xi^{\alpha} \Delta^{\alpha\gamma} a_{\gamma}. \quad (14b)$$

$$3. \quad \mathbf{v}^{\beta} \text{ eigenvector of } J, \quad \mathbf{h}^{\beta} = \lambda \mathbf{v}^{\beta}. \quad (14c)$$

The probabilities of solutions 2 and 3 are small and could even be reduced by a small amount of noise added to \mathbf{x} . Note that \mathbf{k}^{β} is constant as long as we are in the basin of \mathbf{x}^{β} and hence does not contribute differently to variations of \mathbf{x} .

ii) Let \mathbf{x}, \mathbf{x}' be two patterns which fulfill the following additional conditions:

$$\left. \begin{aligned} \mathbf{x} &= \mathbf{x}^{\gamma} + \mathbf{v}^{\gamma} \\ \mathbf{x}' &= \mathbf{x}^{\gamma} + \mathbf{v}'^{\gamma} \end{aligned} \right\} \gamma \in \{1, \dots, p\} \quad \left. \begin{aligned} |\mathbf{v}^{\gamma}| &< |\mathbf{v}^{\beta}| \\ |\mathbf{v}'^{\gamma}| &< |\mathbf{v}'^{\beta}| \end{aligned} \right\} \quad (15)$$

$$\gamma \neq \beta,$$

$$v'_i = v_i, \quad i \neq k, \quad |v'_k| > |v_k|.$$

(Note that we set for simplicity $a_{\gamma} = 1$.) Then it is

straightforward to show that

$$E = - \frac{1}{|x|} \frac{1}{|x_j|} \sum_i (x_i^\beta + v_i^\beta)(x_i^\beta + h_i^\beta + k_i^\beta), \tag{16a}$$

$$E' = - \frac{1}{|x'|} \frac{1}{|x'_j|} \sum_i (x_i^\beta + v_i^\beta)(x_i^\beta + h_i^\beta + k_i^\beta). \tag{16b}$$

With

$$x \cdot x' \frac{1}{|x|} \frac{1}{|x'|} = 1 - \frac{1}{2} \left(\frac{x}{|x|} - \frac{x'}{|x'|} \right)^2$$

we get

$$|v^\beta - h^\beta| < |v'^\beta - h'^\beta|. \tag{17}$$

Now $v'^\beta = v^\beta + \delta v$, $h'^\beta = h^\beta + J\delta v$ and

$$|v^\beta - h^\beta| < |v^\beta - h^\beta + c|, \tag{18a}$$

$$c = \delta v - J\delta v. \tag{18b}$$

This is the case since $c \neq v - h$ after triangle inequality. Therefore, search for an x' near x which has a smaller energy leads nearer and nearer to a

stored pattern x^α . This search can be done by a variety of algorithms one of which (driven by fluctuations) we have chosen for the simulations below.

4. Simulations

Before we demonstrate the network operation a few words about the chosen dynamics.

The kind of random dynamics we apply could be characterized as an evolutionary optimization strategy [17, 18] where optimization means search for the state with minimal energy. To be more concrete: It is a (1 + 1) strategy of the class of $(\mu + \lambda)$ strategies (μ : # of parents, λ : # of descendants). This strategy is simple and well suited for problems in finite sets such as that represented by $x_i \in \{-5, -4, \dots, +5\}$.

A descendant state x' is generated from the parent state x by a random flip in one component according to the possibilities. The energy $E(x')$ is calculated. On this basis it is decided whether x or x' is the new parent state.

We now want to demonstrate the following two functions of the network: Autoassociative memory, in the following mode 1, and pattern classifi-

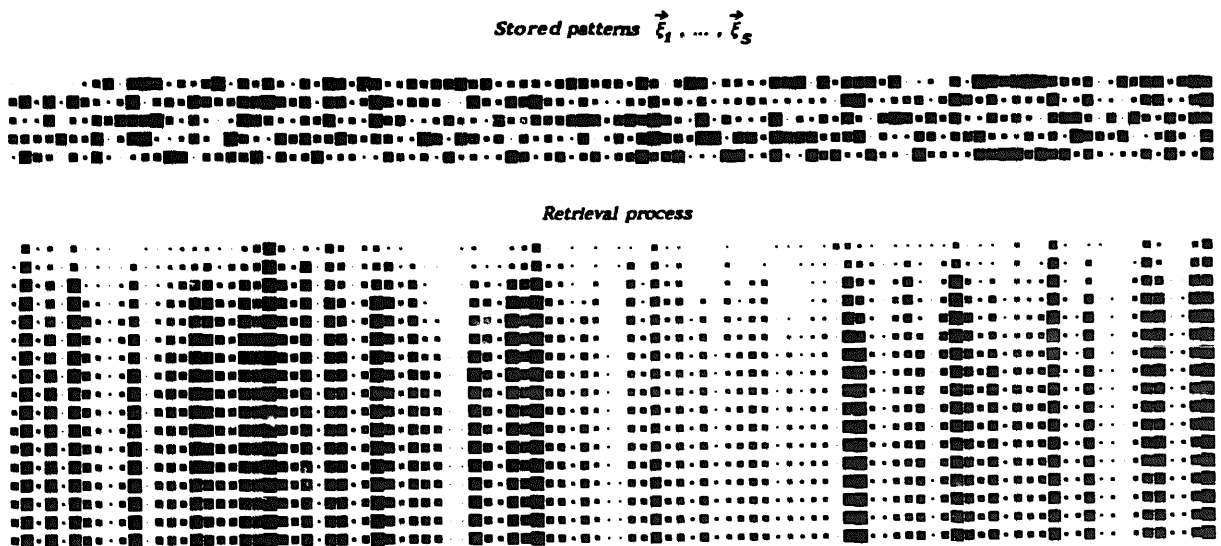


Fig. 2. Example of a pattern vector development. Network of $N = 100$ units each of eleven states $-5, -4, \dots, +4, +5$, coded as the extension of black squares: $\bullet \hat{=} -5, \dots, \blacksquare \hat{=} +5$. $p = 5$ random patterns (non-orthogonal) stored. Dynamics begins with a very perturbed version of pattern 2 and converges finally to pattern 2.

cation, hereafter mode 2. A more detailed description is given in [16]. We use random patterns, since the learning algorithm (2), (6) is very sensible to correlations – as is well known from other work [5–10].

Fig. 2 illustrates mode 1 by an example. The dynamical process changes x as long as $E(x)$ is non-minimal. This, however, depends on step size of the flips which was fixed to $\Delta x = \mp 1$. Fig. 3 shows the decrease in state energy and the corresponding increase in overlap of x to the nearest stored pattern. These curves are well described by $f(T) = e^{-\beta T}$ respectively, $1 - e^{-\beta T}$ laws. Although it could not be applied here, a gradient search strategy would exhibit a similar behaviour. Since Δx is fixed, the convergence velocity scales roughly with the number of dimensions of patterns. As in [19] convergence is said to be reached if the overlap with a stored pattern exceeds 97%. Mode 2 of network operation is shown in fig. 4 where components $c_1 + 1 \dots c_2$ are dedicated to class description and components $c_0 \dots c_1$ are those of different sample patterns. The gain in convergence velocity is considerable, cf. fig. 5.

The capacity of the network can be examined by presenting patterns including more and more noise and observing the recognition capability of the net. As a consequence of the Hebb-like learn-

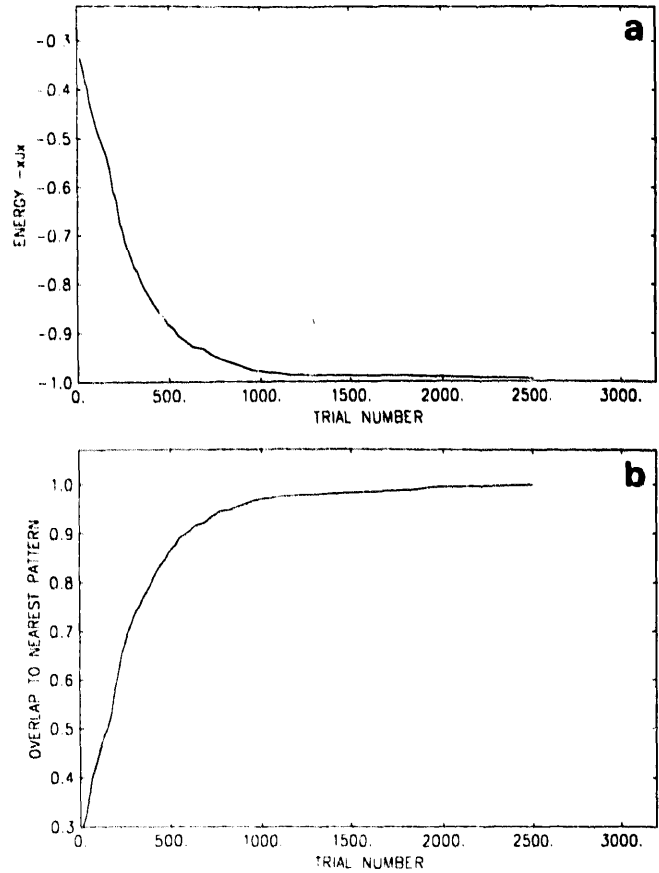


Fig. 3. (a) Energy decrease accompanying the convergence process of fig. 2. (b) Corresponding increase in overlap to pattern 2. After 2500 steps convergence is reached. The exponential decay of perturbations is evident.

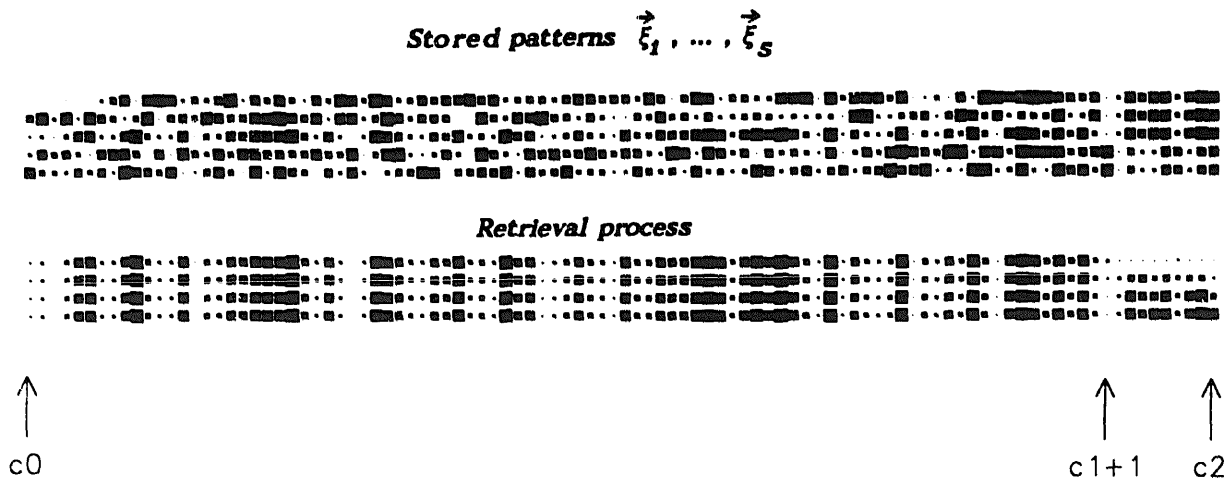


Fig. 4. Pattern classification. Same conditions as in fig. 2: $N = 100$, $p = 5$. Components $c_0 = 1 \dots c_1 = 90$ are different for every random pattern, whereas components $91 \dots 100$ represent classes and are the same for 3 resp. 2 patterns. In the example, a given pattern 3, the class of which is unknown, can be processed very fast to converge to class 1.

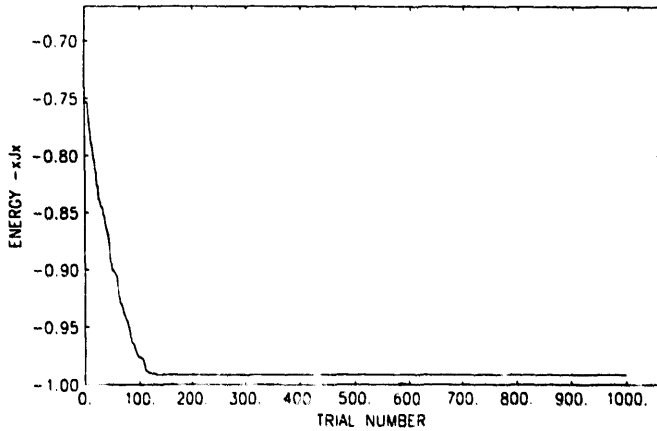


Fig. 5. Convergence in energy is reached very fast (after 130 trials) due to the fact that dynamics was constrained seriously to components 91...100.

ing rule a kind of phase transition should be expected when $\alpha = p/N$ exceeds a certain value. Some results can be seen in Figs. 6(a,b) and 7 respectively.

Fig. 6(a) demonstrates the behaviour of a sample net where different degrees of noise are added to prototype patterns. The amount of noise is measured by the Euclidean distance from the original prototype. Although quite fluctuating, a different choice for α results in a more or less secure retrieval characteristics under the perturbation conditions shown. Fig. 6(b) gives a more refined sketch for $\alpha = 0.05$ with better statistics. One can see a continuous degradation of the retrieval reliability.

Finally, in fig. 7 the influence of α on the retrieval reliability is shown explicitly. One can see a sigmoid-like decrease of retrieval with growing α . The smooth shape of this curve should be contrasted with the sharp falling-off found in Hopfields net [5]. The reason may be that the transition is washed out by choosing an Euclidean distance measure.

5. Conclusion

An example of a new class of associative memory models was presented here, equipped with the simplest possible learning rule of Hebb. Evolu-

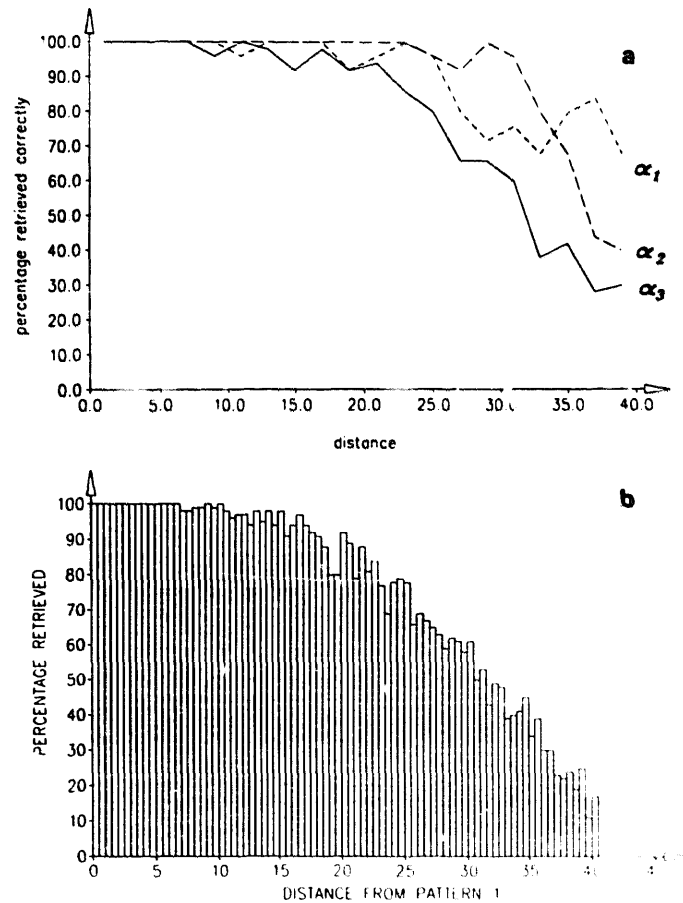


Fig. 6. (a) 25 perturbed patterns are processed for Euclidean distances between 1 and 40. The percentage of correct retrieval (overlap m_c after convergence $m_c > 0.97$) is shown for three different $\alpha = p/N$. (b) 100 perturbations of pattern 1 with Euclidean distance between 1 and 40 are processed in a net with $p = 5$, $N = 100$, $\alpha = 0.05$. Up to distances around 7 retrieval is sure ($m_c > 0.97$).

tionary dynamics was applied since it is well suited for constrained dynamics like this. Consequently, a natural parallelization of the algorithm would be to search on different routes simultaneously.

It should, however, be emphasized that only very simple evolution strategies will do the desired job. In general, more complicated evolution strategies will tend to the global minimum of the Lyapunov function of $E(x)$ [14], whereas memory content is stored in local minima of E . The advantages of further constraining dynamics by using reliable knowledge on parts of the subject were exemplified. At least the possibility of using noise with profit in memory models was indicated.

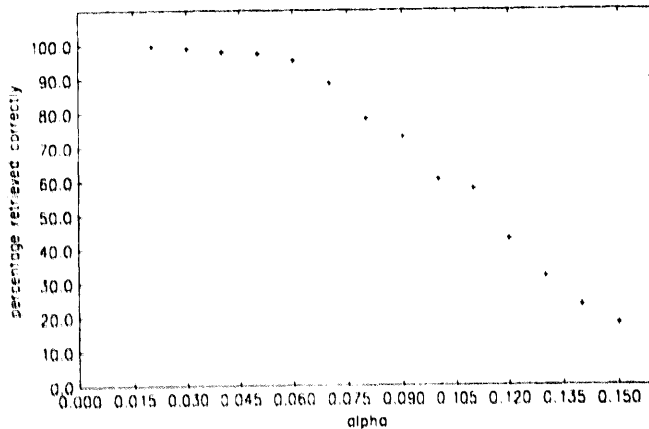


Fig. 7. Dependence of the retrieval on $\alpha = p/N$. Every cross corresponds to a different α and was obtained by averaging over 5 runs of 100 random perturbations each of stored pattern 1 with a constant Euclidean distance of 10.

Examples of application of the kind of network we considered here are given by image or sound processing. In general, all physical data coming into a system in an analog fashion could be mapped more efficiently to a network of multistable units than to one consisting of bistable units.

We have seen the necessity to introduce new non-linearities due to the non-invariance of the pattern vector length in models using multistable units. Note that nature provides us with an immense richness of different kinds of non-linearities. There is no doubt that the diversity in natural phenomena is closely related to this arbitrariness. It may well be that intelligent behaviour as the adaptability to different environmental conditions is only possible in systems with a high degree of freedom to choose internal non-linearities for proper representation of the environment.

Acknowledgements

This work was supported by "Stiftung Volkswagenwerk". I am grateful to the director of the Institute of Theoretical Physics and Synergetics, Prof. H. Haken, for useful discussions and his support. I also want to thank the unknown referees for valuable comments.

References

- [1] W.S. McCulloch and W. Pitts, *Bull. Math. Biophys.* 5 (1943) 115.
- [2] K. Steinbuch, *Kybernetik* 1 (1961) 36.
- [3] G. Palm, *Biol. Cyb.* 36 (1980) 19.
- [4] T. Kohonen, *Selforganization and Associative Memory* (Springer, Berlin, 1983).
- [5] J. Hopfield, *Proc. Natl. Acad. Sci. USA* 79 (1982) 2554; 81 (1984) 3088.
- [6] W. Kinzel, *Z. Phys. B* 60 (1983) 205.
- [7] D.J. Amit, H. Gutfreund and H. Sompolinski, *Phys. Rev. A* 32 (1985) 1007.
- [8] L. Personnaz, I. Guyon and G. Dreyfus, *J. Phys. Lett. (Paris)* 46 (1985) L-359; *Phys. Rev. A* 34 (1986) 4217.
- [9] J.P. Nadal, G. Toulouse, S. Dehaene and J.P. Changeux, *Europhys. Lett.* 1 (1986) 535.
- [10] F. Fogelman Soulie and G. Weisbuch, *J. Phys. (Paris) Lett.* 46 (1985) L-623.
- [11] J. Eccles, *The Understanding of the Brain* (McGraw-Hill, New York, 1973).
- [12] T. Bullock, A. Grinnell and R. Orkland, *Introduction to Nervous Systems* (Freeman, San Francisco, 1977).
- [13] Ackley, G.E. Hinton and T.J. Sejnowski, *Cogn. Science* 9 (1985) 147.
- [14] In preparation.
- [15] H. Haken, in: H. Haken, ed. *Computational Systems, Natural and Artificial*, Proc. Elmau Intern. Symp. on Synergetics 1987 (Springer, Berlin, 1987).
- [16] W. Banzhaf, in: M. Caudill and C. Butler, eds. *IEEE Proc. 1st Intern. Conf. on Neural Networks*, San Diego, 1987, vol. II, p. 223.
- [17] I. Rechenberg, *Evolutionsstrategien* (Frommann-Holzboog, Stuttgart, 1973).
- [18] H. Schwefel, *Numerical Optimization of Computer Models* (Wiley, New York, 1981).
- [19] D.J. Amit, H. Gutfreund and H. Sompolinsky, *Ann. Phys.* 173 (1987) 30.