Biol. Cybern. 62, 107–111 (1989)

Biological Cybernetics



# A New Learning Algorithm for Synergetic Computers

H. Haken, R. Haas, and W. Banzhaf

Institut für theoretische Physik und Synergetik der Universität, Pfaffenwaldring 57/IV, D-7000 Stuttgart 80, Federal Republic of Germany

**Abstract.** We introduce a Lyapunov function which allows us to calculate the vectors, by which the synaptic strengths are determined, by means of a gradient dynamics. The approach does not require any back-propagation, nor simulated annealing, nor computations on a freely running (unclamped) network. The approach is applicable to noiseless patterns as well as to sets of noisy patterns defined by their second and fourth order moments.

## 1 Introduction

The development of adequate learning algorithms for associative nets and neurocomputers is a problem of great current interest (cf. for instance Kohonen 1987; Ackley et al. 1985; Rumelhart and McClelland 1986; Haken 1988b; Banzhaf and Haken 1988; and conference proceedings such as Personnaz and Dreyfus 1989). In this paper we present a learning algorithm for synergetic computers. This algorithm does not need any backpropagation, nor simulated annealing, nor computations on freely running (unclamped) networks which are usually very time-consuming. In previous papers (Haken 1979, 1987, 1988a) it has been shown that there exists a duality between pattern formation by systems far from thermal equilibrium and pattern recognition. This has led us to the proposal of a synergetic computer (Haken 1987, 1988a). The applicability of the corresponding algorithm has been demonstrated by explicit examples dealing with the recognition of faces (Fuchs and Haken 1988a, b, 1989). In this formalism the $M$ prototype patterns are encoded as vectors $\mathbf{v}_k$, $k=1,2,\ldots,M$. We shall assume that for each pattern also its negative is an acceptable pattern so that the moments

$$\langle \mathbf{v} \rangle = 0 \tag{1.1}$$

and

$$\langle v_i v_j v_k \rangle = 0 \tag{1.2}$$

vanish. We shall use the adjoint vectors $\mathbf{v}_k^+$ which have the property

$$(\mathbf{v}_k^+ \mathbf{v}_{k'}) = \delta_{kk'} . \tag{1.3}$$

We shall assume that the adjoint vectors are represented as superpositions of the transposed of the vectors $\mathbf{v}_k$

$$\mathbf{v}_k^+ = \sum_{k'} g_{kk'} \mathbf{v}_{k'}^T . \tag{1.4}$$

When a test pattern vector $\mathbf{q}(0)$ is offered to the system, a dynamics was constructed according to which this initially given test pattern is eventually pulled into the prototype vector $\mathbf{v}_k$ to which it had been nearest

$$\mathbf{q}(0) \to \mathbf{q}(t) \to \mathbf{v}_k . \tag{1.5}$$

The dynamics is described by an equation of the form

$$\begin{aligned} \dot{\mathbf{q}} &= \sum_k \lambda_k \mathbf{v}_k (\mathbf{v}_k^+ \mathbf{q}) - B \sum_{k \neq k'} (v_{k'}^+ \mathbf{q})^2 \, (\mathbf{v}_k^+ \mathbf{q}) \mathbf{v}_k \\ &\quad - C \sum_{kk'} (\mathbf{v}_{k'}^+ \mathbf{q})^2 (\mathbf{v}_k^+ \mathbf{q}) \mathbf{v}_k , \\ \dot{\mathbf{q}} &= - \frac{\partial V}{\partial \mathbf{q}^+} \end{aligned} \tag{1.6}$$

where $V$ is a potential function defined by

$$\begin{aligned} V = &- \tfrac{1}{2} \sum_k \lambda_k (\mathbf{v}_k^+ \mathbf{q})^2 + \tfrac{1}{4} B \sum_{k \neq k'} (v_k^+ \mathbf{q})^2 (\mathbf{v}_{k'}^+ \mathbf{q})^2 \\ &+ \tfrac{1}{4} C \sum_{kk'} (\mathbf{v}_{k'}^+ \mathbf{q})^2 (\mathbf{v}_k^+ \mathbf{q})^2 . \end{aligned} \tag{1.7}$$

Here we have defined the adjoint vector $\mathbf{q}^+$ by means of the relations

$$\mathbf{q} = \sum_k \xi_k \mathbf{v}_k + \mathbf{w} \tag{1.8}$$

and

$$\mathbf{q}^+ = \sum_k \xi_k \mathbf{v}_k^+ + \mathbf{w}^+ . \tag{1.9}$$

The equations resulting from (1.6) and (1.7) with the help of (1.8) and (1.9) can be written quite generally in the form

$$\dot{q}_i = \sum_j \lambda_{ij} q_j + \sum_{jlm} \lambda_{ijlm} q_j q_l q_m . \tag{1.10}$$

As has been shown elsewhere (Haken 1987), the coefficients $\lambda_{ij}, \lambda_{ijlm}$ can be interpreted as synaptic strengths in a neural net where for instance $\lambda_{ij}$ is given by

$$\lambda_{ij} = \sum_k \lambda_k v_{ki} v_{kj}^+ \tag{1.11}$$

and a similar expression holds for $\lambda_{ijlm}$. When our algorithm is realized on a three-layer network, the synaptic strengths between cell $i$ of the input layer and cell $k$ of the middle layer is directly given by the vector components $v_{ki}^+$, while the synaptic strengths between cell $k$ of the middle layer and cell $l$ of the output layer is given by the vector components $v_{kl}$.

## 2 Construction of the Lyapunov Function

We now wish to construct a Lyapunov function by which the $\mathbf{v}_k$ and $\mathbf{v}_k^+$ can be determined by a straight forward procedure. To this end in a first step we introduce a number of patterns to be learned

$$\mathbf{q}_j, \; j = 1, \ldots, M . \tag{2.1}$$

We further introduce the vectors $\mathbf{v}_k$ for $k = 1, \ldots, K$

$$\mathbf{v}_k, \; k = 1, \ldots, K \tag{2.2}$$

where we shall decide later whether $K = M$ or $K < M$. For what follows we consider (1.7) as a function of the vectors $\mathbf{v}_k^+$

$$V(\mathbf{v}_k^+, \mathbf{q}_j) \tag{2.3}$$

from which we construct the potential function $V_1$

$$V_1 = \sum_j V(\mathbf{v}_k^+, \mathbf{q}_j) \tag{2.4}$$

where the sum runs over the patterns offered to the system and to be learned by it. In case the patterns are noisy, a suitable average has to be taken over the distribution function $f(\mathbf{q})$ of these patterns. Denoting the corresponding average by $\langle \ldots \rangle$, we define

$$V_1 = \langle V(\mathbf{v}_k^+, \mathbf{q}) \rangle . \tag{2.5}$$

We now want to show that $V_1$ is a Lyapunov function at least in the case where the $\mathbf{q}_j$'s are noiseless and linearly independent. In case of noisy patterns, $V_1$ will be used to find optimal vectors $\mathbf{v}_k^+$. We shall assume that $\mathbf{v}_k$ and $\mathbf{v}_k^+$ lie in the $\mathbf{q}_j$-space. We wish to determine the $\mathbf{v}_k^+$ by minimizing $V_1$

$$V_1 = \min ! \tag{2.6}$$

Let $\mathbf{v}_k^+$ be given at an initial time $t_i$

$$\mathbf{v}_k^+(t_i) . \tag{2.7}$$

We then wish to show that there is a dynamics according to which $\mathbf{v}_k^+$ is pulled at a final time $t_f$ into $\mathbf{v}_k^+$

$$\mathbf{v}_k^+(t_f) \equiv \mathbf{v}_k^+ \tag{2.8}$$

which just minimizes (2.6) and belongs to the test pattern $\mathbf{q}_j$. To this end we assume that according to the formalism of the synergetic computer $\mathbf{v}_k^+$ had been fixed so that $V$ (1.7) is minimal for $\mathbf{q}_j$. This assumes that the prototype pattern vectors $\mathbf{v}_k$ and the test pattern vectors $\mathbf{q}_j$ in the final state are linearly independent, respectively. Then we may write

$$\mathbf{v}_k^+(t_i) = \mathbf{v}_k^+ U ; \tag{2.9}$$

where $U$ is a matrix.

Forming the scalar product, we obtain the relations

$$\mathbf{v}_k^+(t_i)\mathbf{q}_j = \mathbf{v}_k^+ U \mathbf{q}_j = \mathbf{v}_k^+ \mathbf{q}_j(t_i) , \tag{2.10}$$

where the last equation defines $\mathbf{q}_j(t_i)$. According to (1.6), (1.7) $\mathbf{q}$ can be chosen such that $V$ approaches its minimal value at $\mathbf{q}_j$

$$\mathbf{q}_j(t_i) \to \mathbf{q}_j(t) \to \mathbf{q}_j . \tag{2.11}$$

By means of the matrix $W$ we may express this time evolution in the form

$$\mathbf{q}_j(t) = W(t)\mathbf{q}_j , \tag{2.12}$$

where $\mathbf{q}_j$ is the final state just mentioned. Quite evidently the relation

$$W(t) \to 1 \quad \text{for} \quad t \to \infty \tag{2.13}$$

holds. The scalar product between $\mathbf{v}_k^+$ and $\mathbf{q}_j(t)$ can be written in the form

$$\mathbf{v}_k^+ \mathbf{q}_j(t) = \mathbf{v}_k^+ W(t) \mathbf{q}_j . \tag{2.14}$$

This relationship can be interpreted such that $\mathbf{v}_k^+$ is subject to a time evolution according to

$$\mathbf{v}_k^+ W(t) = \mathbf{v}_k^+(t) . \tag{2.15}$$

Quite evidently we have

$$W(t_i) = U \tag{2.16}$$

and

$$\mathbf{v}_k^+(t_f) \equiv \mathbf{v}_k^+ . \tag{2.17}$$

In this way we have shown that starting from any initially given $v_k^+$ (2.7), we may construct the dynamics so that the $V$s tend to lower values and approach their minima. Thus $V_1$ is a Lyapunov function.

In this argument we have used the constraints that $\mathbf{v}_k$ lies in the space spanned by $\mathbf{q}_k$. For computational

reasons it is more advantageous to introduce a potential function $V_2$ which is minimal if this requirement is met. Therefore we introduce this function by

$$V_2 = C_2 \sum_j \|(1-P)\mathbf{q}_j\|^2 \tag{2.18}$$

where the projection operator $P$ is defined by

$$P = \sum_k \mathbf{v}_k \cdot \mathbf{v}_k^+ . \tag{2.19}$$

More explicitely (2.18) can be expressed by

$$V_2 = C_2 \sum_j ((1-P)\mathbf{q}_j)^T (1-P)\mathbf{q}_j , \tag{2.20}$$

where $T$ means the transposed vector.

Finally, we require that the relation (1.4) holds. To achieve this in the final state, we introduce the potential function $V_3$ by means of

$$V_3 = C_3 \sum_k \left\| \left( \mathbf{v}_k^+ - \sum_{k'} g_{kk'} \mathbf{v}_{k'}^T \right)^2 \right\| \tag{2.21}$$

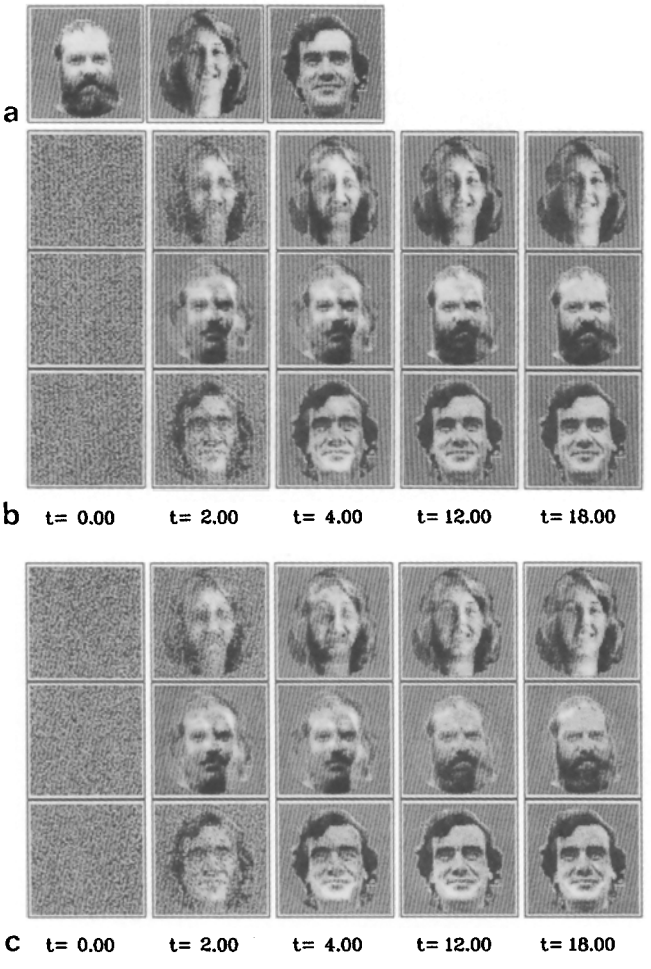


**Fig. 1.** **a** The patterns to be learned simultaneously. **b** Time evolution of the vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$. The initial state was chosen random. **c** Time evolution of the vectors $\mathbf{v}_1^+, \mathbf{v}_2^+, \mathbf{v}_3^+$. The initial state was chosen as $\mathbf{v}_k^+ = \mathbf{v}_k^T$

or written more explicitely by

$$V_3 = C_3 \sum_k \left( \mathbf{v}_k^+ - \sum_{k'} g_{kk'} \mathbf{v}_{k'}^T \right)^T \left( \mathbf{v}_k^+ - \sum_{k'} g_{kk'} \mathbf{v}_{k'}^T \right). \tag{2.22}$$

If the coefficients $C_2$ and $C_3$ are sufficiently large, the constraints (2.18)=0 and (2.21)=0 are practically automatically fulfilled. We now require that

$$V_{\text{tot}} = V_1 + V_2 + V_3 = \min! \tag{2.23}$$

holds. The gradient dynamics is now applied with respect to the variables

$$v_{k,i};\ v_{k,i}^+;\ g_{kk'} . \tag{2.24}$$

In our practical calculations it has turned out that we may save a good deal of computing time when we first form

$$(\mathbf{v}_k^+ \mathbf{q}_j) \tag{2.25}$$

and then perform all the other steps. This is also true if $\mathbf{q}_j$ is presented again and again in the case of noisy patterns. On the other hand from the purely theoretical point of view our approach is equivalent to one in which the sums over $j$ are replaced by the average values

$$\sum_j q_{ji} q_{jl} \to \langle q_i q_l \rangle \tag{2.26}$$

and correspondingly

$$\langle q_i q_j q_l q_m \rangle . \tag{2.27}$$

In this way we may state that our algorithm allows us to recover the prototype vectors $\mathbf{v}_k$ and their adjoints, if the second and fourth order moments of patterns are given. Figure 1a–c and Table 1 show how the system

**Table 1.** Time evolution of the matrix elements $g_{kk'}$

| | (1,1) (2,1) (3,1) | (1,2) (2,2) (3,2) | (1,3) (2,3) (3,3) |
|---|---|---|---|
| t=0.00 | 1.000 | 0.000 | 0.000 |
| | 0.000 | 1.000 | 0.000 |
| | 0.000 | 0.000 | 1.000 |
| $t$=6.00 | 0.745 | −0.022 | −0.142 |
| | −0.017 | 0.667 | −0.014 |
| | −0.144 | −0.009 | 0.836 |
| $t$=10.00 | 0.751 | −0.002 | −0.169 |
| | 0.005 | 0.619 | −0.055 |
| | −0.171 | −0.051 | 0.887 |
| $t$=14.00 | 0.833 | 0.024 | −0.189 |
| | 0.030 | 0.683 | −0.113 |
| | −0.190 | −0.110 | 0.964 |
| $t$=18.00 | 0.924 | 0.035 | −0.197 |
| | 0.038 | 0.807 | −0.183 |
| | −0.198 | −0.182 | 1.026 |

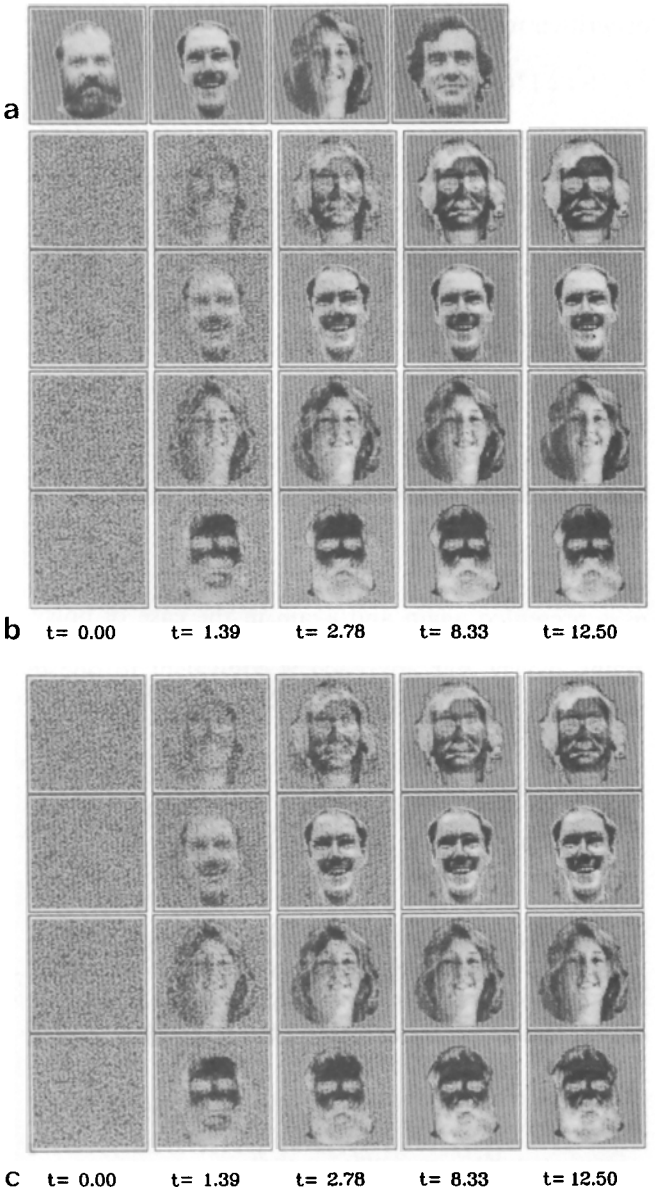


**Fig. 2.** **a** The pattern to be learned simultaneously. **b** Time evolution of the vectors $\mathbf{v}_k$, $k=1,\ldots,4$. The initial state was chosen random. **c** Time evolution of the vectors $\mathbf{v}_k^+$, $k=1,\ldots,4$. The initial state was chosen as $\mathbf{v}_k^+ = \mathbf{v}_k^T$

learns three patterns offered to it. The pattern vectors are represented here as two-dimensional arrays which allows us at the same time to visualize the working of the algorithm. As is seen here, the whole formalism works very well with vectors having $60 \times 60$ pixles. This is certainly no more a toy problem. Figure 2a–c and Table 2 present our results for the simultaneous learning of four patterns. Two of the faces are reproduced by their negatives, which is allowed by our formalism. Finally, Figs 3a–c and Table 3 show our results when noisy patterns are offered. These results

**Table 2.** Time evolution of the matrix elements $g_{kk'}$

| | (1,1) | (1,2) | (1,3) | (1,4) |
|---|---|---|---|---|
| | (2,1) | (2,2) | (2,3) | (2,4) |
| | (3,1) | (3,2) | (3,3) | (3,4) |
| | (4,1) | (4,2) | (4,3) | (4,4) |
| t=0.00 | 1.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 1.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 1.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 1.000 |
| $t=1.39$ | 0.997 | 0.006 | 0.002 | −0.007 |
| | 0.005 | 0.992 | −0.001 | 0.008 |
| | 0.004 | −0.002 | 1.005 | 0.001 |
| | −0.003 | 0.005 | 0.000 | 0.998 |
| $t=2.78$ | 0.941 | 0.025 | −0.029 | −0.014 |
| | 0.027 | 0.960 | −0.006 | 0.050 |
| | −0.025 | −0.004 | 0.953 | 0.025 |
| | −0.015 | 0.053 | 0.029 | 0.973 |
| $t=8.33$ | 0.751 | 0.081 | −0.045 | −0.055 |
| | 0.083 | 0.968 | −0.032 | 0.228 |
| | −0.039 | −0.032 | 0.844 | 0.090 |
| | −0.058 | 0.229 | 0.092 | 0.914 |
| $t=12.50$ | 0.761 | 0.110 | −0.056 | −0.080 |
| | 0.111 | 1.023 | −0.038 | 0.292 |
| | −0.051 | −0.038 | 0.882 | 0.106 |
| | −0.082 | 0.293 | 0.107 | 0.971 |

**Table 3.** Time evolution of the matrix elements $g_{kk'}$

| | (1,1) | (1,2) | (1,3) |
|---|---|---|---|
| | (2,1) | (2,2) | (2,3) |
| | (3,1) | (3,2) | (3,3) |
| t=0.00 | 1.000 | 0.000 | 0.000 |
| | 0.000 | 1.000 | 0.000 |
| | 0.000 | 0.000 | 1.000 |
| $t=2.50$ | 0.937 | 0.085 | −0.013 |
| | 0.084 | 0.907 | 0.010 |
| | −0.018 | 0.018 | 0.925 |
| $t=5.00$ | 0.942 | 0.155 | −0.053 |
| | 0.155 | 0.873 | 0.040 |
| | −0.056 | 0.046 | 0.857 |
| $t=15.00$ | 1.041 | 0.318 | −0.050 |
| | 0.318 | 1.035 | 0.102 |
| | −0.049 | 0.102 | 0.945 |
| $t=22.50$ | 1.097 | 0.390 | −0.027 |
| | 0.390 | 1.105 | 0.130 |
| | −0.027 | 0.131 | 0.972 |

demonstrate how our formalism can be used for the reconstruction of noisy patterns.

As just mentioned, the algorithm allowed us to determine the prototype vectors simultaneously. The formalism can also be applied to a sequential learning. Here for instance first a series of a single noisy pattern

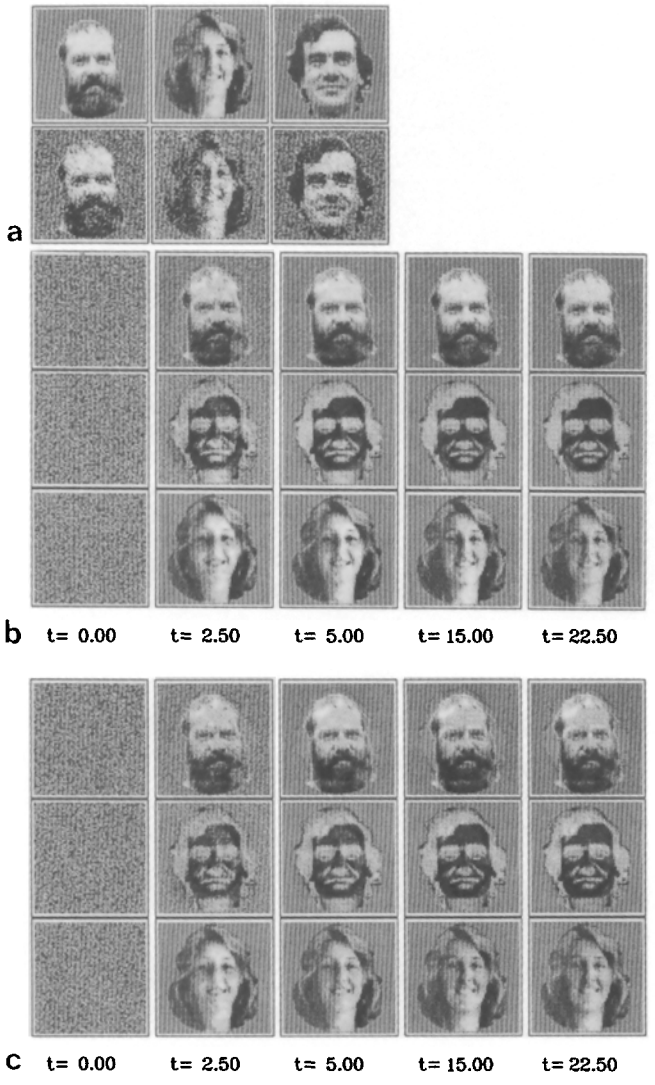


**Fig. 3.** **a** Noisy patterns offered to the computer. Upper part: noiseless patterns. Lower part: a typical noisy pattern. **b** Time evolution of $\mathbf{v}_k$. **c** Time evolution of $\mathbf{v}_k^+$

is given and only one $\mathbf{v}$ is used. Then in the next step one we may offer the computer the same $\mathbf{v}$ again and a new set of noisy patterns which are supposed to belong to a new class so that a new additional vector $\mathbf{v}_2$ is determined. In this way we may proceed stepwise. Our approach can be generalized in various ways, e.g. we may require a minimum angle between the prototype vectors or prescribe a limited number of $\mathbf{v}$'s in order to perform a classification. The results of this analysis will be published elsewhere.

*Acknowledgement.* We wish to thank the Volkswagenwerk foundation for financial support.

## References

Ackley DH, Hinton GE, Sejnowski TJ (1985) A learning algorithm for Boltzmann machines. Cogn Sci 9:147–152

Banzhaf W, Haken H (1988) A network for recognition and classification of continuous patterns. First Annual Meeting of INNS, Boston

Fuchs A, Haken H (1988a) Pattern recognition and associative memory as dynamical processes in a synergetic system, part I. Biol Cybern 60:17–22

Fuchs A, Haken H (1988b) Pattern recognition and associative memory as dynamical processes in a synergetic system, part II. Biol Cybern 60:107–109

Fuchs A, Haken H (1989) Pattern recognition and associative memory as dynamical processes in a synergetic system. Biol Cybern 60:476

Haken H (1979) Pattern formation and pattern recognition – an attempt at a synthesis. In: Haken H (ed) Pattern formation by dynamical systems and pattern recognition. Springer, Berlin Heidelberg New York

Haken H (1987) Synergetic computers for pattern recognition and associative memory. In: Haken H (ed) Computational systems, natural and artificial. Springer, Berlin Heidelberg New York

Haken H (1988a) Synergetics in pattern recognition and associative action. In: Haken H (ed) Neural and synergetic computers. Springer, Berlin Heidelberg New York

Haken H (1988b) Information and selforganization. Springer, Berlin Heidelberg New York

Kohonen T (1987) Selforganization and associative memory. Springer, Berlin Heidelberg New York

Personnaz L, Dreyfus G (1989) Neural networks from models to applications. I.D.S.E.T., Paris

Rumelhart DE, McClelland JL (1986) Parallel distributed processing. MIT Press, Cambridge, Mass

Received: June 19, 1989
Accepted: August 7, 1989

Prof. H. Haken
Institut für theoretische Physik
und Synergetik der Universität
Pfaffenwaldring 57/IV
D-7000 Stuttgart 80
Federal Republic of Germany