Proceedings

First International Symposium on Intelligence in Neural and Biological Systems INBS'95

May 29 - 31, 1995

Herndon, Virginia

Sponsored by

IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence (PAMI)

In cooperation with

IEEE Computer Society
IEEE Computer Society TAI Conference
AAAI Society
AAAS Society
SMC Society
NN Society

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

Self-organized Learning in Multi-layer Networks

Rüdiger W. Brause

J.W.Goethe-University, FB Informatik, D-60054 Frankfurt, FRG
brause@informatik.uni-frankfurt.de

Abstract

We present a framework for the self-organized formation of high level learning by an statistical preprocessing of features. The paper focuses first on the formation of the features in the context of layers of feature processing units as a kind of resource-restricted associative learning. We clame that such an architecture must reach maturity by basic statistical proportions, optimizing the information processing capabilities of each layer. The final symbolic output is learned by pure association of features of different levels and kind of sensorial input.

Finally, we also show that common error-correction learning can be accomplished also by a kind of associative learning.

1 Introduction

In every-day life we can observe the astonishing abilities of a kind of nature-made information processing systems, called "children". As designer of information- processing computer systems which tries to implement good visual and speech- recognition features we have to admit that mother nature has already done better than us: The natural systems do not need (normally!) preprocessed, noise-free selected input or to be adjusted in convergence parameters. Since complex computer systems need such a data fault-tolerant, self organized user interface, we should ask impatiently: How can we implement a system presenting the same features? This paper tries to present the view of some of the questions concerning the fault-tolerant, self-organized processing of features to symbols, but there all still many questions left open. On my opinion, we are still in the beginning of understanding how the brain works, so this disadvantage should be essential for the future research.

Let us first look to known proportions due to the **experimental observations** of natural systems.

1) For the visual system, we know that the information, although intrinsically massivly parallel, is processed sequentially in several areas of the brain, see e.g. [17]. Figure 1.1 shows the raw structure of different stages.

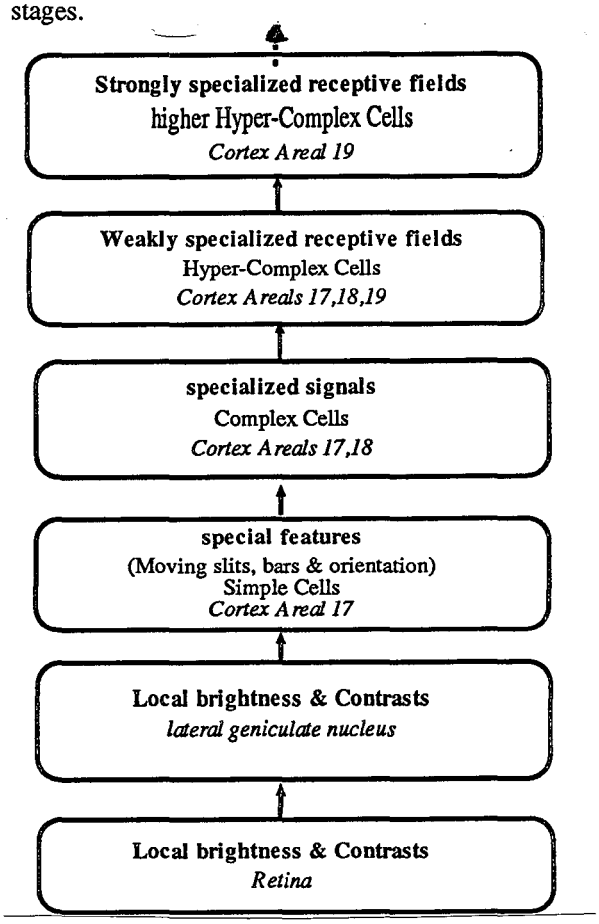


Fig.1.1 The raw visual layer structure

Here, the sensory input is first processed by cells which give simple responses. Then, the responses are tied to more and more complex input patterns. Induced by this, a hypothetical last layer neuron, which is only active when the grandmother comes into sight, is called a "grandmother neuron". It is not reasonable that such a neuron really exists, because it maps a certain event to a single neuron. Since in all living beings neurons die with a certain

rate, an animal which codes an important event by only one neuron might die shortly after the corresponding neuron, enfavouring others who code it by several neurons.

2) For the first layer, according to the experiments of Kuffler [16], we know that the sensory input is processed by neurons weighting their neighboured input by a special weighting function, called "receptive field". Due to its form, one kind is called "Mexican hut" function. Similar receptive fields have been found in the auditory pathway. Daughman showed that the experimental findings for receptive fields in different layers of cat visual cortex can all be modelled by windowed, locally formed Fourier components (e.g. wavelets or Garbor functions) [10].

   The receptive fields of successive layers are enlarged, which can be explained by surjective projections of the neuronal output to the next layer; either by spin-offs of the axons or by the extension of the dendritic tree.

3) The characteristics of the information processing in each layer are quite different. For the input, after a logarithmic intensity encoding stage, we know that the visual processing is simply linear. The following layers are not so well explored. For the second layer, we know that each receptive field of it is streched in a certain direction. Edges which are aligned in parallel to this direction cause a high activity reaction of the neuron. Since there are several directions, the visual information is processed by a set of feature detectors. For every pixel, there is a set of feature detectors, organized in a columnar structure.

4) The whole connection structure is controlled by a maturating process. It is well known that all higher animals are subject to an imprinting stage which takes more or less time. In this stage, lower to higher order abilities ("connections") are formed and, after the end of the imprinting time-out, constantly maintained. Neurophysiological findings for the visual cortex [13] show that in this time the cytosceleton of the lower layer neurons are formed and impede all changes in the synaptic circuitry after that time period.

In general, the further we proceed in the encoding pathway, the less we know about the nature of the encoding. Thus, the main source of ideas lays in simulations and functional models of the information processing. Here, some ideas of systems for technical application of artificial neural networks might help which are described in the next sections.

2 Outline of an information processing model

Let us introduce the model by some propositions, which are not mandatory. Their only purpose is to introduce an information processing system which is consistent to the findings of the previous section. After introducing the assumptions, we will try to fill up the frame with more substantial, mathematically sustained model parts.

Preposition 1:
The main information processing is done in several stages, called "layers", instead of only one giant network.

Remark: This preposition (which is based on observation 1) precludes not the existence of feedback lines. However, theses lines should have orders in magnitude of information stream less than the feedforward lines.

Preposition 2:
Each stage tries to extract the maximal information of the input with the least ressources.

Remark: This preposition needs more evaluation. For instance, we do not know exactly what "least ressources" means. For example, this can be measured by the number of neurons per output bits/sec, by the necessary number of synaptic weights or by an layer activity measure which takes the energy stream (e.g. acetylcholin or oxygen stream, switching current, dissipation heat, etc.) into account.

Preposition 3:
The maturation of the layers starts at the first input layer and effects the higher order layers afterwards, according to correlated activity.

Remark: This generalizes the biological observations, that the ripening process depends on the activation by sensory input and that chemical molecules (e.g. MAP2, see [13]) which are responsible for low-level cytosceleton maturation are also present in the brain parts, used for higher levels of information processing.

Preposition 4:
The maturation is identical to the stationarity of the output pattern probability distribution.

Remark: Propositions 3 and 4 introduce the idea that the system of layers is subject to certain ripening processes. The observed fact that humans can not learn low level primitives after a certain imprinting time can indicate a certain biological sense. On the background of multi-layer simulation experience we can suggest that

this might be the means to provide stable learning to subsequent layers by a stationary input distribution. Otherwise, changes of the distribution in the first layer might cause a complete unstable learning process in higher order layers causing unstable action sequences.

Preposition 5:
Learning in these layers is directed by statistical proportions of associations, not by back-propagated error correction or other direct pattern feedback information.

Remark: This idea excludes all backpropagation learning algorithms. The main reason for this preposition is the fact that, since we do not know the internal behavour of our nervous system, we can not guide it properly by special error patterns. All feedback must be incorperated by slow, general information providing mechanism, not by distinctive patterns.

As a consequence, all learning is provided by associative correlations, see the models in the following sections.

Preposition 6:
After an object seperation process, which is automatically provided by the statistically feature processing stages, the semantic meaning is introduced by an pure associative learning process.

Remark: The association is not limited to features of only one kind. Conversely, the name of an object is an association to the speech recognition parts of the brain which is induced by the famous experiments with splitted brain hemispheres, cutting the corpus callosum.

This was an outline of the whole model.
In figure 2.1 the main system structure is shown as a block diagram. Prepositions 2 and 5 will be evaluated in detail in the next sections.

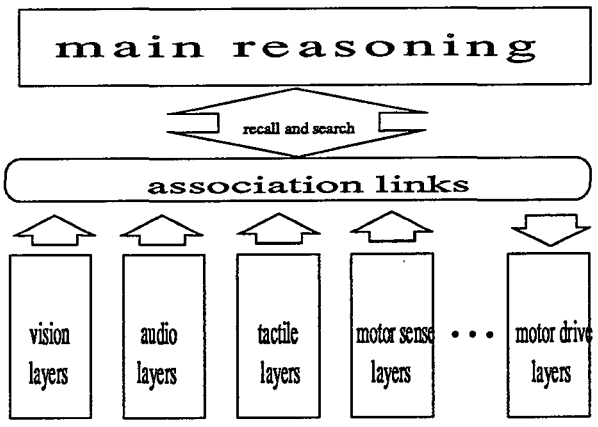


Fig. 2.1 A model for feature processing and semantic associations

3 Parallel information processing

Preposition 2 deals with the optimal information processing capability of each layer. For biological systems, the idea of maximal redundancy reduction [Bar61] or maximal information gain [Lin88], [Hak88] was introduced by several researchers.

Here, we introduce by proposition 2 the additional constraint of limited ressources. After the intuitive introduction of the learning context, let us try in this section to clarify the mathematical conditions for optimal information processing.

3.1 Optimal information processing

One of the most popular information criterion is the maximization of the mutual information or transinformation H_{trans} from the input $\mathbf{x}=(x_1,..,x_n)$ to the output lines $\mathbf{y}=(y_1,..,y_m)$

$$H_{trans} = H(\mathbf{x};\mathbf{y}) = H(\mathbf{x}) + H(\mathbf{y}) - H(\mathbf{y},\mathbf{x}) \qquad (3.1)$$

which, for constant input information $H(\mathbf{x})$ and observed information $H(\mathbf{y})$, heavily depends on the combined source information $H(\mathbf{y},\mathbf{x})$.

One of the most simple layer functions is a linear transformation, obtained by m parallel active neurons, each one with $y_i=\mathbf{w}_i^T\mathbf{x}$ as transfer function, yielding $\mathbf{y}=\mathbf{W}\mathbf{x}$ as layer transformation. With rank(W)=n, the probability density function $p(\mathbf{x})$ which transforms generally by the *Jacobian* $\det(\partial\mathbf{y}/\partial\mathbf{x})=\det(\mathbf{W})$ (the determinant of the matrix of the functional derivatives, see [26]), transforms here with the scaling factor $\det(\partial\mathbf{x}/\partial\mathbf{y}) = \det(\partial\mathbf{y}/\partial\mathbf{x})^{-1} = 1/\det(\mathbf{W})$ of the space volume.
In the linear case we get therefore for the information

$$H(\mathbf{y}) = H(\mathbf{x}) + \log\det(\mathbf{W}) \qquad (3.2)$$

This means e.g. for a Gaussian distributed random variable $\mathbf{x}$ which is transfomed linearly that therandom variable $\mathbf{y}$ is also a Gaussian distributed random variable.

For a scale-invariant transformation (rotation etc) with $\det(\mathbf{W})=1$ also the information H(.) does not change. Because the transinformation is the difference between two transformed random variables, it does not depend on the scaling factor.

An efficient coding of the variables $y_1,...,y_m$ is given when their common information, i.e. the transinformation, becomes very small. Generalizing equation (3.1) we get

$$H(y_1;...;y_m) = H(y_1)+H(y_2)+...+H(y_m) - H(y_1,...,y_m)$$

For general random variables we have

$$p(y_1,...,y_m) = p(y_1)\, p(y_2|y_1) \cdots p(y_n|y_1,...,y_{m-1})$$

and after some algebra we get

$$H(y_1,...,y_m) = H(y_1) + H(y_2|y_1) + ... + H(y_n|y_1,...,y_{m-1})$$

The transinformation becomes very small, when

$$H(y_i) = H(y_i|y_1,...,y_{i-1}) \quad \text{i.e. } p(y_i) = p(y_i|y_1,...,y_{i-1})$$

or $$p(y|x) = p(y) = p(y_1)p(y_2) \cdots p(y_m) \qquad (3.3)$$

Thus, to carry most of the information the output lines must become independent.

For the first layer, we know that the probability distribution of the signal values of each pixel are Gaussian distributed

$$p(x) = A \exp(-(x-x_0)^T C_{xx}^{-1}(x-x_0))$$

with $A=[(2\pi e)^n \det C_{xx}]^{-1/2}$

and $x_0 = \langle x \rangle$, $C_x = \langle (x-x_0)(x-x_0)^T \rangle$ *covariance matrix*

Here, the demand of (3.2) can easily be satisfied by a layer implementing a linear decorrelation with $\langle (y_i-y_{i0})(y_j-y_{j0})^T \rangle = 0$ for $i \neq j$, because with

$$C_{yy} = \langle (y-y_0)(y-y_0)^T \rangle = \begin{pmatrix} \sigma_1^2 & & 0 \\ & \cdots & \\ 0 & & \sigma_m^2 \end{pmatrix}$$

with $y_0 := \langle y \rangle$, $\sigma^2_i := \langle y^2_i \rangle$

we get for the also Gaussian-distributed output y after the linear transformation

$$p(y) = B \exp(-(y-y_0)^T C_{yy}^{-1}(y-y_0))$$

with $B=[(2\pi e)^n \det C_{yy}]^{-1/2}$

$$= B \exp(-\textstyle\sum_i (y_i-y_{i0})^2/\sigma_i^2)$$

$$= B^{1/m} \exp(-(y_1-y_{10})^2/\sigma^2_1) \cdots B^{1/m} \exp(-(y_m-y_{m0})^2/\sigma^2_m)$$

$$= p(y_1) \cdots p(y_m)$$

the condition (3.3) for independent random variables.

What can we deduce by this proportion? From the information point of view, a layer which transfers most of the incoming information, can be purely linear for Gaussian distributed input signals. This is true for pixel statistics or short time speech statistics, i.e. for the primary structures of the incoming information. Therefore, the linear proportions of the first stages of visual perception (see section 1) are sufficient.

4 A model for self-organized input encoding

In the previous section we have seen that the main demand for parallel encoded signal lines is their independence of each other. We have seen that for Gaussian distributed input, this can be achieved by a linear system which decorrelates the signals.

For this reason, let us investigate this idea in more detail for a concrete model for the first layers of one of the column in figure 2.1, where the signals are still Gaussian distributed.

There are several possibilities to obtain a decorrelation by artificial neural networks. The mostly known ones are the networks for principal component analysis (PCA), yielding as principal components the eigenvectors of the crosscorrelation matrix of the input. Many approaches exist which either lead only to an eigenvector subspace with correlated coeefficients, e.g. Oja subspace network [19] and the lateral inhibition network of Földiák [11], or prescribes the formation order of the eigenvectors, e.g. the Sanger decomposition network [24] or the lateral inhibition network of Rubner and Tavan [22].

Contrary to all these approaches, let us use the recent proposal [5] for a fully symmetrical network for PCA, construced by an objective function and implemented by a biological plausible and in VLSI easily realizable network mechanism.

4.1 The model

Let us assume in a first step that we have m neurons which are laterally interconnected as shown in figure 4.1.

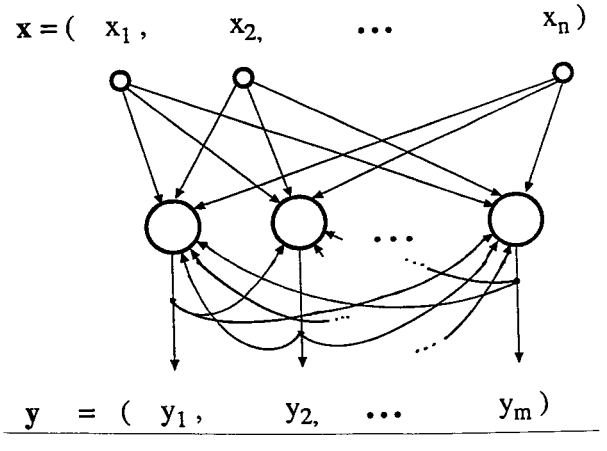


Fig. 4.1 The symmetric, lateral interconnected network model

Each neuron i has a randomly chosen weight vector $\mathbf{w}_i$. After we presented one input pattern $\mathbf{x}$ in parallel to each neuron of the linear system, the output of neurons will result in

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{s} \qquad \mathbf{s} = \mathbf{U}\mathbf{y},\ u_{kk}=0 \quad (4.1)$$

where $\mathbf{s}=(..s_i..)$ denotes the influence by the lateral connections which are weighted by the lateral weights u_{ij}. Rearranging (4.1) leads to

$$\mathbf{y} = \mathbf{A}\mathbf{x} \qquad \text{with } \mathbf{A} = (\mathbf{I}-\mathbf{U})^{-1}\mathbf{W}$$

The input is assumed to be centered. If this is not the case, it can be made by introducing a special threshold weight learned with an Anti-Hebb-rule, see [4].

The learning rule for the weights $\mathbf{a}_i$ is determined by the minimum of a deterministic objective function, composed by the minimal crosscorrelation R_1 and the maximal autocorrelation or variance R_1

$$R(\mathbf{a}_1,..,\mathbf{a}_m) = 1/4\ \beta\Sigma_i\Sigma_{j\neq i}\ (\langle y_i y_j\rangle)^2 - 1/2\ \Sigma_i\ \langle y_i^2\rangle$$
$$= R_1 + R_2 \quad (4.2)$$

and is reached when the weight vectors become the eigenvectors of the correlation matrix $\mathbf{C}$ for $|\mathbf{a}_i|=1$, see [5]; the lateral inhibition weights become zero and the output variance of a neuron becomes the corresponding eigenvalues λ_i. To learn the weight vectors $\mathbf{a}_i$, a gradient descend may be used. Nevertheless, with (4.2) this leads to complicated expressions for $\mathbf{w}_i$ and u_{ij}. Instead, we can use the stochastic algorithm for learning the weights

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \gamma(t)\ \mathbf{x}\ (y_i + \beta\Sigma_{j\neq i}\ u_{ij}y_j)$$
$$= \mathbf{w}_i(t) + \gamma\ \mathbf{x}\ \tilde{y} \quad (4.3)$$

For u_{ij} the temporal floating average of the observed data can be used. It should be noticed that the difference equation converges under with the constraints $\beta > 2/\lambda_{min}$ and $\gamma < 2/\lambda^2_{max}$.

Please note that (4.3) is an associative learning rule. It should be emphasized that the whole associative process converges only because the restriction $|a|=const$ is maintained; otherwise the weights would increase infinitely without directional preference. This is indeed an important constraint which manages a kind of ressource distribution by increasing the weights for active lines and weakens them for passive ones. The constraint corresponds to the "least ressource" demand of preposition 2 and can be explained by a limited molecule flow for the synaptic developement process. For VLSI systems, it can be easily implemented by the Kirchhoff law, see [6].

4.3 Self-organization in a cellular neural network

In this section a self-organized, local formation of the PCA primitives, the eigenvectors (for image data: the eigen images) by the only locally interconnected network of the previous section is presented. This approach is completely new: it combines the optimal PCA properties of the network in the input space with a kind of self-organization in the space of the physical input (and output) layout.

One of the main new ideas of the paradigm of neural networks is the restriction of a neuron to only local data processing, e.g. to a subset of all available input lines. This idea is also supported by many arguments for redundancy removal in biological systems [3] and fits also well to the needs of VLSI design which favours building big systems by the replication of small, modular, local functions. Since the VLSI design is normally implemented on a 2-dim wafer, the approach is well suited for 2-dim sensor fields, e.g. for image processing. Nevertheless, the networks can also principally used in 1-dim or 3-dim design or any other number of neighbourhood dimensions. A typical input layout is shown in figure 4.2. Here, only the sensor elements (disks) and the neurons(rectangangles), but no output lines are shown.

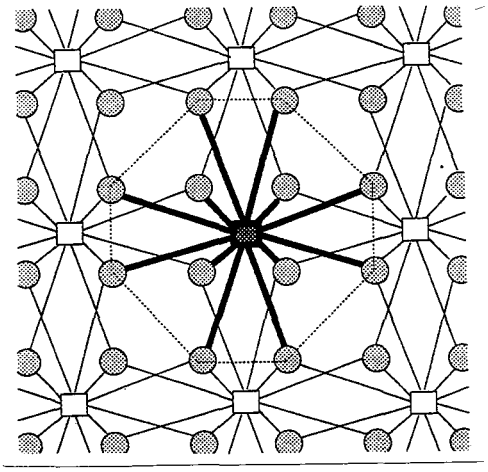


Fig. 4.2 The modularized, 2-dim neural net design

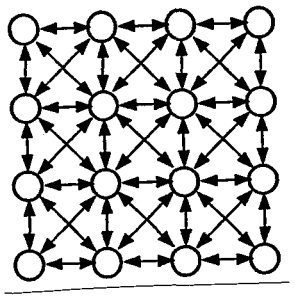
For the *activity phase*, a modular, localized organization of networks has been coined by Leon Chua and his coworkers by the term *cellular neural networks (CNN)* [9]. Since the matrix of the local input connections can be seen as a local picture processing operator which is identical to the operators used in conventional image processing (e.g.[1]) the CNN paradigm has been adopted by an international group of

scientists as a paradigm for a supercomputer for image processing, having a performance of 10^{12}=1000 GIPS (Giga instructions per second) in current available technology [27]. Here, the weights (*template*) $W_{(ij)}$ and $U_{(ij)}$ of a neuronal cell at location (i,j) are set arbitrarily by the user and can be seen as a form of programming.

In this section, we show that the modular organization of the weights in cellular neural networks can be also achieved by a non-supervised, self-organized *learning process phase*. Let us consider a symmetrical, lateral inhibited network as it has been introduced in section 4.1. Additionally, let us assume that we have only a limited radius *r* of inhibition influence as it is defined for CNN's. This corresponds to local windows which have for squared tiles as eigenfunctions two-dimensional sine and cosine waves [8]. For Gaussian type of windows the simulation shows that this results in the same, but Gaussian modulated kind of waves [25]. This means that we are in fact encoding the image signal by a kind of localized Fourier transform with very special basis functions. Assuming a local Fourier transform for the visual cortex, its function can be consistently explained [20].

Now, we want to show that the only locally defined interactions between the neurons imply a self-organizing process. For the simulation we used input patterns of n=36 components, each one set by Gaußean noise with different variance. The input weights for the m=16 neurons, arranged in a 2-dim order (see figure 4.3), are randomly initilized with a fixed vector length $|w_i|=1$, the lateral weights are initialized with zero. The parameters β and γ_0 are set according to convergence condition with decreasing $\gamma(t)$.

For the inhibition radius r=1 each neuron converges to an eigenvector. If we denote the index of the eigenvector (denoted by the descending order of their associated eigenvalues λ_i, i.e. $\lambda_1=\lambda_{max}$) the following configurations can be observed in three runs, see Fig.4.3. The inhibition forces all other neurons within the inhibition radius to converge to eigenvectors with other eigenvalues enabling a self-organized two-dimensional formation of eigenvectors. This is also the case of 1-dim. inhibition arangements, see [7].



run1	run2	run3
3 4 3 1	1 4 1 5	2 3 4 1
2 1 5 2	3 2 3 2	5 1 5 2
4 3 4 3	1 5 4 1	4 3 4 3
1 2 1 2	2 3 2 3	1 2 1 2

Fig. 4.3 The lateral inhibition interactions of m=16 CNN-neurons and the formation of local eigenvector sets

Although in this simulation the whole input is received by all neuronal units, the same results can be attended for systems with also localized input (local receptive fields) if the input statistics are translation-invariant. For most data like speech and image this is the case, because the neighboured data points are more correlated than ones with a longer distance, independent of the absolut position in time or picture coordinates.

Thus, each input sensor point (e.g. each image pixel) is represented by a local linear superposition of a locally changing set of eigenvectors. In (4.3) two sets of run2 are encircled as examples. The image representation can be compared to the 3-dot colour matrix encoding used in colour TV tubes to encode a arbitrary colour by three components. The resolution of such a device is determined by the distance between two eigenvector sets, i.e. two eigenvectors of the same index. If we choose the inhibition radius equal for all neurons, the regular pattern like the one in (4.3) will occur.

Our previous prepositions 5 and 6 assume pure associative, resource-restricted learning, either in an unsupervised, self-organized manner of section 4 or in the classical associative manner, given for example by the correlative matrix memory, see [15]. However, these two learning mechanism do not cover the case where unknown complex patterns w have to be learned according to a general performance criterion.

5.1 Error correction learning

Here, the well-known backpropagation mechanism [23] is successfully used, based on the gradient search

$$w(t) = w(t-1) - \gamma(t)\nabla_w R(w) \qquad (5.1)$$

of the least expected quadratic error R(w,L) between the performance z of the neuron, based on a weight pattern w, and the teacher evaluated goal F

$$R(w,L) := \langle (F(x)-z(x))^2 \rangle_x$$
$$\nabla_w R(w) = -\langle 2(F(x)-z(x))\, \nabla_w z(x) \rangle_x \qquad (5.2)$$

which gives for linear neurons $z(x)=w^T x$ the stochastic approximation

$$w(t) = w(t-1) + \gamma(t)(F(x)-w^T x)\, x \qquad (5.3)$$

with special conditions for the the learning rate $\gamma(t)$.

Unfortunately, for the learning of complex movement patterns, now human being does know the complex derivatives of his internal movement generation mechanism to be used in equation (5.2). Instead, a much simpler mechanism of associative learning can be used instead, described in the next section.

5.2 Evolutionary associative learning

Conventional associative learning mechanism try to associate a given stimulus pattern x with the appropriate response L(x) by a learning rule

$$w(t) = w(t-1) + \gamma(t)\, L(x)\, x \qquad (5.4)$$

This kind of learning might be adequate if the quantities L and x are given, but it does not solve the problem of finding an unknown pattern w which produces L.

To overcome this restriction, let us assume that x is a randomized version of w. This assumes a learning context where a new movement is tried after the old one was not successfull. If we take a constant learning rate (which weights the last events higher and depends less on old, bad samples), the w as an performance weighted average depends highly on the random properties of the pattern x.

This random walk is demonstrated in a simulation, shown in figure 5.1. Here, the squared error if shown during an iteration of 160 samples. Obviously, there is no convergence.

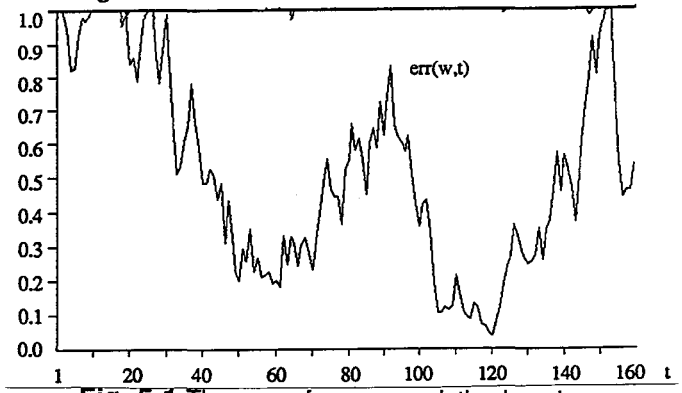


Fig. 5.1 The error of pure associative learning

This brings us to the conclusion that we have include in order to learn something not only the actual pattern performance $R(x(t))=R_t$ but also the former performance $R(x(t-1))=R_{t-1}$. For example, we might correct the current pattern estimation w if the performance has increased $R_t-R_{t-1}>0$, otherwise not

$$w(t) = w(t-1) + \gamma(t)\, L(x)\, x \qquad (5.5)$$

$$\text{with} \quad L(R_t-R_{t-1}) = \begin{cases} 1 & \text{if } R_t-R_{t-1} > 0 \\ 0 & \text{else} \end{cases} \qquad (5.6)$$

$$\text{and} \quad p(x) = A \exp(-x^T C^{-1} x)$$

which is a kind of *evolutionary learning* [21]. In figure 5.2 the error developement of such a learning system is shown.

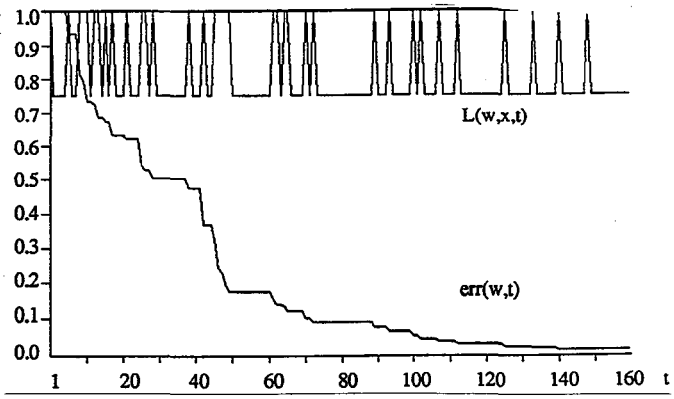


Fig. 5.2 The error of binary evolutionary associative learning

Each improvement x is a random deviation of the pattern w according to a Gaussian distribution with equal width $\sigma=0.3$, i.e. $C^{-1}= I\sigma^{-1}$ and $\gamma=0.2$. The figure shows additionally as a change indicator the function $L'(t)=0.75 + 0.25\, L(t)$ which indicates a change in w by a spike. Obviously, for (5.6) the error can only decrease.

The basic learning equation (5.5) contains a performance function L(t,t-1) of (5.6) which can be very different. Instead of a binary threshold function used in (5.6) we can also consider the linear case

$$L(R_t-R_{t-1}) = R_t-R_{t-1} \qquad (5.7)$$

In figure 5.3 the error developement of (5.5) using (5.7) is shown.

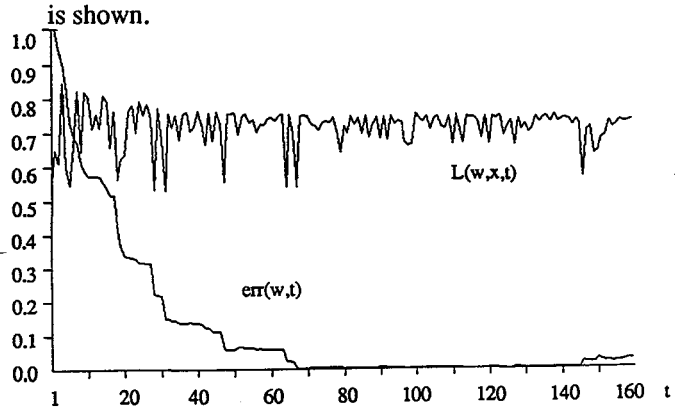


Fig. 5.3 The error of linear evolutionary associative learning

In the upper part of the drawing we see the indicator function L'(t) again. In difference to the performance of

(5.6) we need less iterations to approach the goal, because in the neighbourhood of the goal the step width is automatically reduced, whereas in (5.6) it remains constant. We have to skip more random variations to get a better performance; unfortunately, the random deviations prevent us from stability after reaching the goal. In figure 5.4 the three algorithms are compared due the random walks they produce.

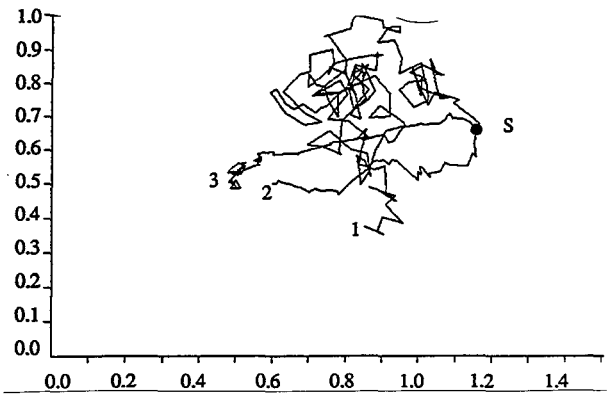


Fig. 5.4 The random walks of evolutionary associative learning

For two-dimensional patterns **w** and **x** a common random starting point S and a goal Δ located at (0.5,0.5) are used. The walks start all at a black dot S and terminate, after 160 patterns have been presented, at the end of the lines, numbered 1,2 and 3 according to the algorithms of (5.4), (5.6) and (5.7). The convergence tendency of the three associative algorithms can be observed using the same parameters as above: the first produces an random walk without apparently approaching the goal, the second one approaches it directly, but slowly and the third one approaches fast (but oscillates around the goal).

An increase in the random component would accelerate the algorithms in the start, but would lead in the final phase to a slower convergence for the algorithm (5.6) and to higher random deviations for (5.7).

REFERENCES

[1] D. H. Ballard, Ch. Brown: Computer Vision; Prentice Hall 1982

[2] H. Barlow: The coding of sensory messages; in: Thorpe,Zangwill(eds.): Current Problems in Animal Behaviour, Cambridge Univ. Press, 1961

[3] T. Bossomaier, A. Snyder: Why Spatial Frequency Processing in the Visual Cortex?; Vision Research, Vol.26, No.8, pp.1307-1309, 1986

[4] R. Brause: The Minimum Entropy Network; Proc. IEEE Tools for Art. Intell. TAI-92, Arlington (1992)

[5] R. Brause: A Symmetrical Lateral Inhibited Network for PCA and Feature Decorrelation; Proc. Int. Conf. Art. Neural Networks ICANN-93, Springer Verlag 1993, pp.486-489

[6] R. Brause: A VLSI-Design of the Minimum Entropy Neuron; in: J.Delgado-Fria, W. Moore(esds.): VLSI for Artificiasl Intelligence and Neural Networks, Plenum Publ. Corp., 1994

[7] R. Brause: Picture Encoding using Self-organized Cellular Neural Nets; Proc. Int. Conf. on Art. Neural Networks ICANN-94, Springer Verlag 1994, pp.1125-1128.

[8] R. Brause: Neuronale Netze, Teubner Verlag 2nd ed., Stuttgart 1995

[9] L. O. Chua, L. Yang: Cellular neural networks: Theory; IEEE Trans. Circuits Syst., Vol. 35, pp.1257-1272, Oct. 1988
and L.O.Chua, L.Yang: Cellular neural networks: Applications; IEEE Trans. Circuits Syst., Vol. 35, pp.1273-1290, Oct. 1988

[11] P. Földiák: Adaptive Network for Optimal Linear Feature Extraction; IEEE Proc. Int. Conf. Neural Networks; pp. I/401-405 (1989).

[13] B. Gordon, E. Allen, P. Trombley: The Role of Norepipene in Plasticity in the Visual Cortex; Progress in Neurobiology, Vol30, No.2/3, pp. 171-191 (1988)

[14] H. Haken: Information and Self-Organization; Springer Verlag Berlin Heidelberg 1988

[15] T.Kohonen: Correlation Matrix Memories; IEEE Transactions on Computers, Vol C21, pp.353-359, (1972)

[16] S.W.Kuffler: Discharge Patterns and Functional Organization of Mammalian Retina; Journal of Neurophys., Vol 16, No1, pp.37-68, (1953)

[17] M. Levine: Vision in man and machine; McGraw Hill 1985

[18] R. Linsker: Self-Organization in a Perceptual Network; IEEE Computer, pp. 105-117, (March 1988)

[19] Erkki Oja: Neural Networks, Principal Components, and subspaces Int. J. Neural Systems, Vol 1/1 pp. 61-68 (1989)

[20] K.Okajima: A Mathematical Model of the Primary Visual Cortex and Hypercolumn; Biol. Cyb. Vol. 54, pp. 107-114 (1986)

[21] Ingo Rechenberg: Evolutionsstrategie; problemata frommann-holzboog, 1973

[22] J. Rubner, P. Tavan: A Self-Organizing Network for Principal-Component Analysis, Europhys.Lett., 10(7), pp. 693-698 (1989).

[23] D.E.Rumelhart, J.L.McClelland: Parallel Distributed Processing; Vol I, MIT press, Cambridge, Massachusetts 1986

[24] T. Sanger: Optimal unsupervised Learning in a Single-Layer Linear Feedforward Neural Network; Neural Networks Vol 2, pp.459-473 (1989)

[25] T. Sanger: Analysis of the Two-Dimensional Receptive Fields Learned by the Generalized Hebbian Algorithm in Response to Random Input; Biol. Cybernetics, Vol63, pp.221-228 (1990)

[26] C.E.Shannon, W.Weaver: The Mathematical Theory of Information; Univ. of Illinois Press, Urbana 1949

[27] Special issue on cellular neural networks, IEEE Transactions on Circuits and Systems I, Vol. 40, No. 3, March 1993