# SELF-ORGANIZED LEARNING IN MULTI-LAYER NETWORKS

RÜDIGER W. BRAUSE
*J. W. Goethe-University, Computer Science Dep., NIPS*
*D-60054 Frankfurt., Germany*
*email: brause@informatik.uni-frankfurt.de*

**ABSTRACT**

We present a framework for the self-organized formation of high level learning by a 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 multiresolution 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 for motor skills can be accomplished also by non-specific associative learning.

*Keywords:* feedforward network layers, maximal information gain, restricted Hebbian learning, cellular neural nets, evolutionary 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 designers of information processing computer systems which try 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 giving rise to the same features?

This paper tries to present the view for some of the questions, especially concerning the fault-tolerant, self-organized processing of features to symbols; but there all still many questions left open for future research.

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

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

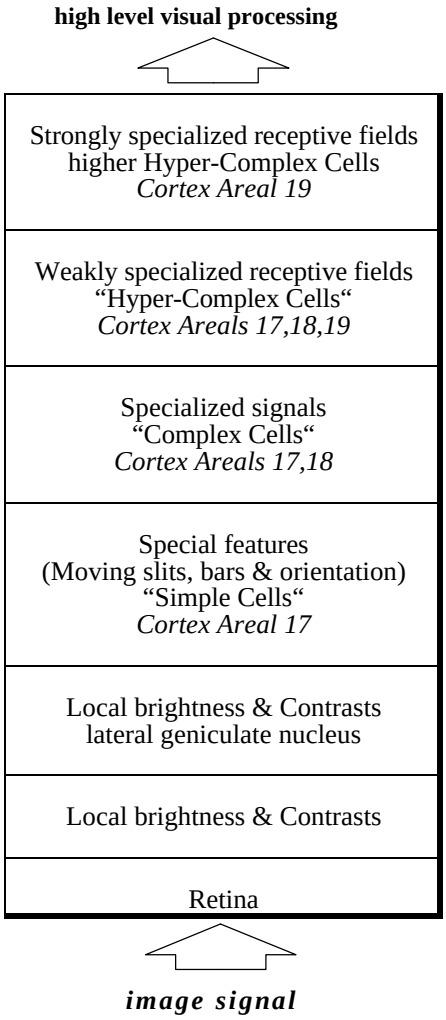


Fig. 1 Information processing in the raw visual layer structure

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 view, a hypothetical last layer neuron should exist which is only active when e.g. the grandmother comes into sight and is therefore 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 only by one neuron might die shortly after the corre-

sponding neuron, enfavouring others who code it by several neurons.

2) For the first layer, according to the experiments of Kuffler [2], we know that the visual sensory input is processed by neurons weighting their neighbored input by a special weighting function, called "receptive field". 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 dendrite tree. Due to its form, the receptive field function is called "Mexican hat" function. Similar receptive fields have also been found in the auditory pathway.
   Daughman showed in [3] that the experimental findings for receptive fields in different layers of cat visual cortex can all be modeled by windowed, locally computed Fourier components (e.g. wavelets or Gabor functions).
   In [4] Okajima showed that biological visual system can be interpreted by local Fourier transforms which are organized in sets of frequency components, each one forming one hypercolumn on the visual cortex. In this model, also shift and deformation invariance of visual recognition is supported.
   Nevertheless, both authors do not say how these Fourier or Gabor transforms evolve in the cortex.

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 stretched 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 period, constantly maintained. Neurophysiological findings for the visual cortex [5] 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 system. Here, some ideas for the technical application of artificial neural networks might help us 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, completely connected network.

*Remark:*
This preposition (which is based on observation 1) precludes not the existence of feedback lines within and between layers. However, the feedback lines between layers 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 resources.

*Remark:*
This preposition needs more evaluation. For instance, we do not know exactly what "least resources" means. For example, this can be measured by the number of neurons per output bit per second of a layer, by the necessary number of synaptic weights or by a 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 [5]) 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 mature state 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 might have 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 providing a stationary input distribution to them. 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 behavior of our nervous system, we can not guide it properly by special error patterns. All feedback must be incorporated by slow, very general feedback information (e.g. attention level, emotional level etc.), not by distinctive patterns. As for preposition 1, this does not exclude feedback mechanism on the same (low) level as for instance the spinal motor reflex. It only prohibits specific error correction pattern feedback from high to low level.

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

**Preposition 6:**
After an object separation 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 Fig. 2 the main system structure is shown as a block diagram. Prepositions 2 and 5 will be evaluated in detail in the next sections.

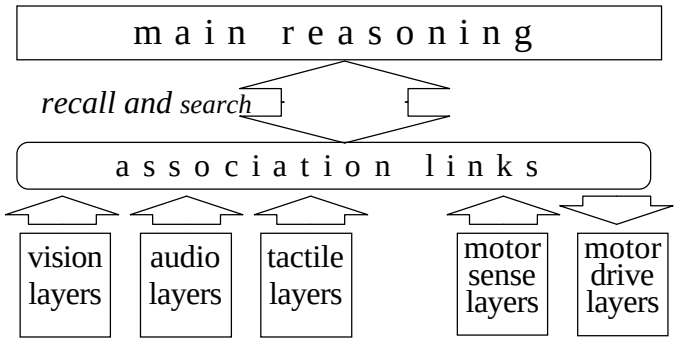


Fig. 2 A model for feature processing and semantic associations

## 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 [6] or maximal information gain [7], [8] was introduced by several researchers.

Here, we introduce by preposition 2 the additional constraint of limited resources.

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 $H_{trans}$ (see [9]) from the input $\mathbf{x}=(x_1,..,x_n)$ to the output lines $\mathbf{y}=(y_1,..,y_m)$

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

which, for constant input information H(x) and observed information H(y), heavily depends on the compound source information H(x,y).

One of the most simple layer functions is a linear transformation, obtained by *m* parallel neurons, each one with the transfer function $y_i = \mathbf{w}_i^T\mathbf{x}$, yielding $\mathbf{y} = \mathbf{W}\mathbf{x}$ as layer transformation. With rank($\mathbf{W}$)=n, the probability density function p($\mathbf{x}$) which transforms generally by the Jacobian $\det(\partial y/\partial x)=\det(\mathbf{W})$ (the determinant of the matrix of the functional derivatives, see [9]), transforms here with the scaling factor $\det(\partial x/\partial y) = \det(\partial y/\partial x)^{-1} = 1/\det(W)$ of the space volume.
In the linear case we get therefore

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

This means e.g. for a Gaussian distributed random variable *x* which is transformed linearly that the random variable *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

$$H(y_1;...;y_m) = H(y_1)+H(y_2)+...+H(y_m) - H(y_1,...,y_m) \stackrel{!}{=} \min$$

For general random variables we have

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

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

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

$$\text{or } p(y|x) = p(y) = p(y_1)p(y_2)\cdots p(y_m) \qquad \textit{independence condition} \quad (3.3)$$

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

$$p(\mathbf{x}) = A \exp(-(\mathbf{x}-\mathbf{x}_0)^T C_{xx}^{-1}(\mathbf{x}-\mathbf{x}_0))$$

$$\text{with } A=[(2\pi e)^n \det C_{xx}]^{-1/2} \text{ and } \mathbf{x}_0 = \langle \mathbf{x} \rangle, \quad C_{xx} = \langle (\mathbf{x}-\mathbf{x}_0)(\mathbf{x}-\mathbf{x}_0)^T \rangle$$

*covariance matrix*

Here, the demand of (3.3) can easily be satisfied by a layer implementing a linear decorrelation with $\langle (y_i-y_i^0)(y_j-y_j^0)^T \rangle = 0$ for $i \neq j$ , because with

$$C_{yy} = \langle (y-y_0)(y-y_0)^T \rangle = \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m^2 \end{pmatrix} \text{ with } y_0 := \langle y \rangle, \ \sigma^2_i := \langle y^2_i \rangle$$

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

$$\begin{aligned} p(\mathbf{y}) &= B \exp(-(\mathbf{y}-\mathbf{y}_0)^T C_{yy}^{-1}(\mathbf{y}-\mathbf{y}_0)) \qquad \text{with } B=[(2\pi e)^n \det C_{yy}]^{-1/2} \\ &= B \exp(-\Sigma_i (y_i-y_{i0})^2/\sigma^2_i) \\ &= B^{1/m} \exp(-(y_1-y_{10})^2/\sigma^2_1) \cdots B^{1/m} \exp(-(y_m-y_{m0})^2/\sigma^2_m) \\ &= p(y_1) \cdots p(y_m) \end{aligned}$$

the condition (3.3) for independent random variables.

What can we deduce by this proportion? From the information point of view, a layer which encodes the information in parallel signals best, 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 proportion of the first stages of visual perception (see section 1) is 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 Fig. 2, 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 coefficients, e.g. Oja subspace network [10] and the lateral inhibition network of Földiák [11], or prescribes the formation order of the eigenvectors, e.g. the Sanger decomposition network [12] or the lateral inhibition network of Rubner and Tavan [13].

Contrary to all these approaches, let us use the recent proposal [14] for a fully symmetrical network for PCA, constructed 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 Fig. 3.

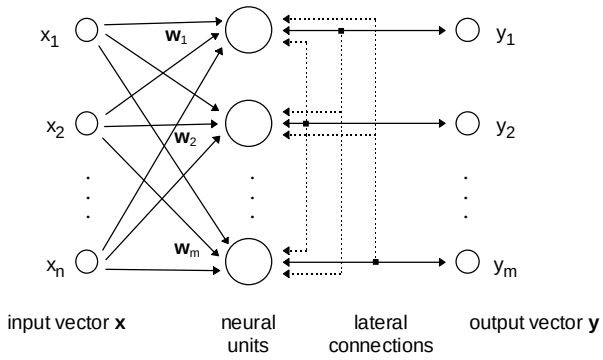


Fig. 3 The symmetric, laterally interconnected network model

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

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{s} \qquad \mathbf{s} = \mathbf{U}\mathbf{y}, \;\; u_{kk}=0 \tag{4.1}$$

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

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

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 [15].

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 $R_1$

$$R(\mathbf{a}_1,..,\mathbf{a}_m) = 1/4\,\beta \sum_i \sum_{j\neq i} \left\langle y_i y_j \right\rangle^2 - \frac{1}{2}\sum_i \left\langle y_i^2 \right\rangle = R_1 + R_2 \tag{4.2}$$

and is reached when the weight vectors become the eigenvectors of the correlation matrix C for $|\mathbf{a}_i|=1$, see [14]; the lateral inhibition weights become zero and the output

variance of a neuron becomes the corresponding eigenvalues $\lambda_i$. 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

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \gamma(t)\,\mathbf{x}\,(y_i + ß \sum_{i \neq j} u_{ij} y_j) = \mathbf{w}_i(t) + \gamma\,\mathbf{x}\,\tilde{y} \qquad (4.3)$$

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

Please note that (4.3) is an associative learning rule of the Hebbian type. It should be emphasized that the whole associative process converges only because the restriction $|\mathbf{a}|$=const is maintained; otherwise the weights would increase infinitely without directional preference. This is indeed an important constraint which manages a kind of resource distribution by increasing the weights for active lines and weakens them for passive ones. The constraint corresponds to the "least resource" demand of preposition 2 and can be explained by a limited molecule flow for the synaptic development process. For VLSI systems, it can be easily implemented by the Kirchhoff law, see [16], which describes a resource restriction law for electric current.

### *4.2 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 eigenimages) 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 [17] and fits also well to the needs of VLSI design which favors 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 neighborhood dimensions. A typical input layout is shown in figure 4. Here, only the sensor elements (disks) and the neurons (rectangles), but no output lines are shown.

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) [18]. 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.[19]) the CNN paradigm has been adopted by an international group of scientists as a paradigm of a supercomputer for image processing, having a performance of $10^{12}$=1000 GIPS (Giga instructions per second) in current available technology [20]. Here, the weights (template) $\mathbf{W}(ij)$ and $\mathbf{U}(ij)$ of a neuronal cell at location (i,j) are set arbitrarily by the user and can be seen as a form of programming.

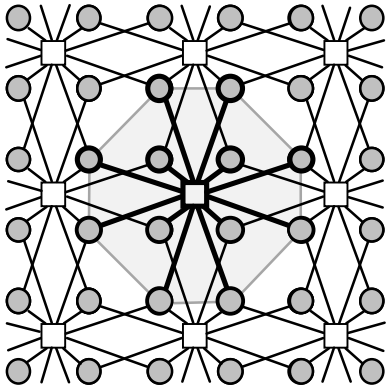

Fig. 4 The modularized, 2-dim neural net design

Now, let us 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 a local window. In the case of square tiled windows we know that the eigenfunctions are two-dimensional sine and cosine waves [21]. For Gaussian type of windows simulations show that this results in the same, but Gaussian modulated kind of waves [22]. 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 [23].

Now, we want to show that the only locally defined interactions between the neurons of our model 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 Fig. 5), are randomly initialized with a fixed vector length $|\mathbf{w}_i|=1$; the lateral weights are initialized with zero. The parameters $\beta$ and $\gamma_0$ 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 unit by the index of the eigenvector (denoted by the descending order of their associated eigenvalues $\lambda_i$, with $\lambda_1=\lambda_{max}$) the following index configurations can be observed in three runs, see Fig. 5.

The inhibition forces all other neurons within the inhibition radius of each unit to converge to eigenvectors with other eigenvalues enabling a self-organized two-dimensional formation of eigenvectors. This is also the case in 1-dim. inhibition structures, see [24].

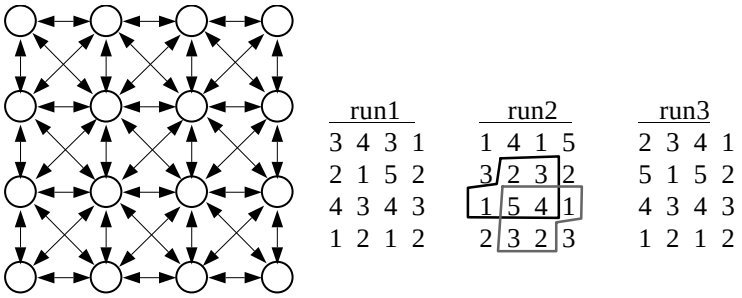


Fig. 5 The lateral inhibition interactions of m=16 CNN-neurons and the formation of local eigenvector sets

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 neighbored data points are more correlated than ones with a longer distance, independent of the absolute 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 figure 5 two sets of run2 are encircled as examples.

The image representation can be compared to the 3-dot color matrix encoding used in color TV tubes to encode an arbitrary color by a linear superposition of three components. The resolution of such a device is determined by the distance in the field between two eigenvector sets, i.e. two eigenvectors of the same index. If we choose the inhibition radius equal for all neurons, a regular pattern like the one in figure 5 will occur.

It should be noted that the local eigenvector decomposition developed above depends on the linear proportion of the neurons used. If we use non-linear neurons instead, we might get a signal decomposition based on specific patterns, not on the average pattern, see [25], which leads to a segmentation, not a linear decomposition.

### *4.3 Layered information processing and multiresolution encoding*

We have shown in the previous section that a self-organization process can be driven by a lateral inhibition and restricted Hebbian learning. This corresponds to a local Karhunen-Loéve transform (KLT). It is well known that for natural image statistics the analytical solution of the KLT are the sine and cosine basis functions with distinctive frequencies and phases. Thus, the main difference between transcendental transforms as the Fourier transform or the cosine transform and the KLT is the determination of the optimal frequencies and phases in the latter. This supports the view of Okajima [4] for the vision system, see statement 2) of section 1.

Now, how can we describe mathematically the *layered* information processing with the tool of Fourier transforms? In a classical paper, Marko [26] developed a formalism for describing layered filters. Nevertheless, let us concentrate on the fact that there is a kind of convergence of the signals by the multilayer approach. In Fig. 6 this is shown in one dimension for non-overlapping receptive fields.

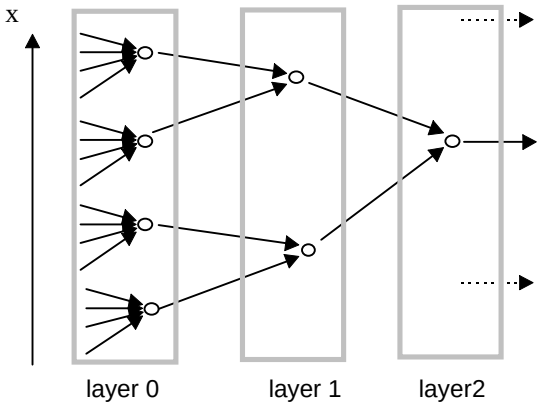


Fig. 6 The convergence of input signals through the layers

In the last few years possible mechanisms for the information processing in layers became clearer. One of the most favorite candidates is the model of multirate sampling or multiresolution encoding which is well described in the book of Vaidyanathan [27]. Here, the convergence of the signal wiring can be described by a filter (the neuronal processing in one layer) and a subsampling of the resulting signal by the wiring from one layer to the next layer. This is shown schematically in Fig. 7.

The filtering process consists of two symmetrically arranged, overlapping filter banks (a high-pass filter and a low-pass filter: Quadrature Mirror Filter QMF-Filter [27]) The low pass filter forwards all signal frequency components which are lower than a certain bandwidth limit without processing to the next layer, whereas the high pass filter measures the amount of high frequency components. Since the low frequencies (due to Shannon´s sampling theorem [28]) need only lower sampling frequencies and therefore less sample points, each low pass filter is followed by a subsampling stage (see fig.7) which is implemented in Fig. 6 by the fact that we have a smaller number of output lines than input lines. For the wavelet decomposition [29], the basis function of the low pass filter is called a *scaling function* and for the high pass filter a *wavelet.*

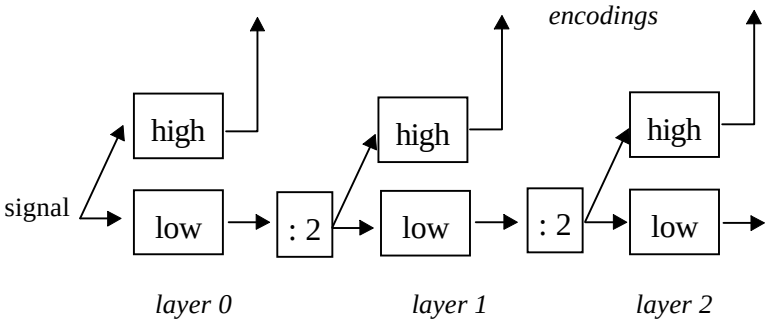


Fig. 7 The sequence of filters and subsampling

Now, since the low pass filter let all low frequency signal components pass without processing, by the repetitive filtering and frequency rescaling the frequency band is

successively cut off at the high end. At each cut, a part (subband) of the spectrum is encoded, the rest is passed to the next layer. We can implement this multirate system also by a parallel approach, showed in Fig. 8. The power spectrum |y(f)| of the signal x(t) is devided into several overlapping intervals or *subbands* by the linear decomposition of basis functions with different frequency characteristics (*filter banks*). In Fig. 8a a filter bank system and in Fig. 8b the frequency responses of the different encoded signals $y_i$ are shown.

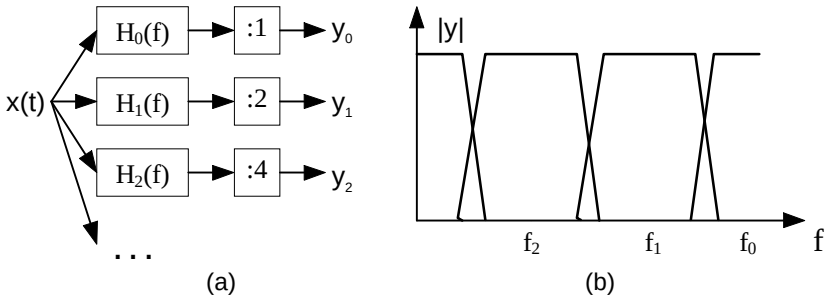


Fig. 8 The parallel multiresolution scheme: Filter banks and subbands for multirate sampling

Since each filtered signal is subsampled, the corresponding part of the original signal is scaled (compressed) on the time scale. Thus, for the parallel system the corresponding basis function have to be rescaled (expanded) to represent the real basis function. The corresponding sampling interval is therefore also expanded, resulting in a different interval, i.e. in image encoding in a different area surface for each basis function. Thus, the parallel signal decomposition is made by different basis functions (of different layers) with different sampling interval sizes (different receptive field sizes). The wide basis functions of the low frequencies cover the rough details of the signal whereas the small basis functions of higher frequencies (first layers in the sequential scheme) encode the finer details. For this, the name *multiresolution signal encoding* even for the sequential scheme has evolved.

In conclusion: what can we deduce by this subsection? We have shown that the multilayer modeling, implied by the observations of 1) in section 1.1, can be explained by a multiresolution scheme. Also the enlargement of the receptive fields of observation 2) which has been observed in the cortex has an interpretation in this context. Nevertheless, the sequential multiresolution scheme assumes two different kind of base functions which have not been observed yet explicitly. It is not clear up to now whether this is not principally the case or due to the similarity of the waveforms of the two kinds of base functions.

## 5 Learning Movement Patterns

According to preposition 1 and Fig. 2, the learning is done in layers. This also includes the motor skills. One of the common sense models for such a layering is the software layer model for robot control, see e.g. [30] and Fig. 9.

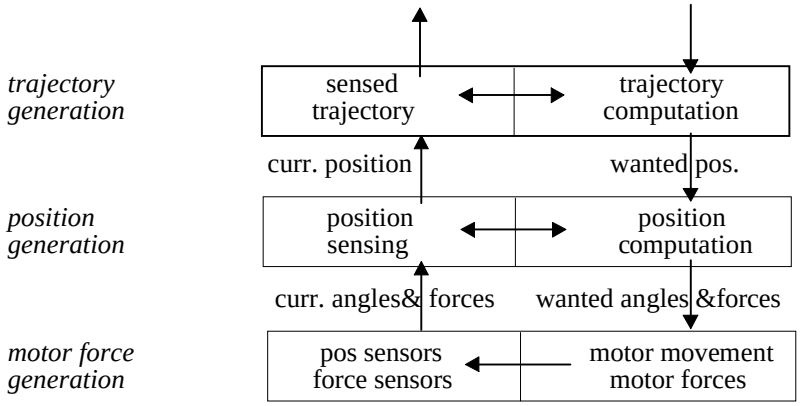


Fig. 9 The robot control layers

Here, all low level signals are processed in one signal layer and the more abstract signals are passed to the upper layer. From the upper motor layer abstract commands are passed to the lower motor layer and transformed in concrete associated timed pattern sequences. There are possible interactions between the two parts of one layer. In the lowest layer, they are called *motor reflexes*.

Each layer processes the low level sensor signals and higher level commands as inputs and has higher level sensor signals and low level commands as outputs. This is shown in Fig. 10

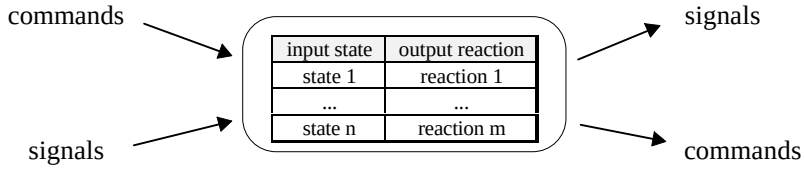


Fig. 10 Associative information processing in one layer

Within the layer, the input states (input patterns) are associated to the output reactions (output patterns) in a list, i.e. a kind of associative memory.

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 [31]. 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 [32] has been successfully used, based on the gradient search

$$\mathbf{w}(t) = \mathbf{w}(t\text{-}1) - \gamma(t)\, \nabla_w R(\mathbf{w}) \qquad (5.1)$$

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 L

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

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

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \gamma(t)\,(L(\mathbf{x})-\mathbf{w}^T\mathbf{x})\,\mathbf{x} \tag{5.3}$$

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

Unfortunately, for the learning of complex movement patterns, no human being knows the complex derivatives of his internal movement generation mechanism which is used in equation (5.2). Instead, a much simpler mechanism of associative learning can be used which is described in the next section.

### *5.2 Evolutionary associative learning*

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

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \gamma(t)\,L(\mathbf{x})\,\mathbf{x} \tag{5.4}$$

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 successful. 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 Fig. 11. Here, the squared error if shown during an iteration of 160 samples. Obviously, there is no convergence.

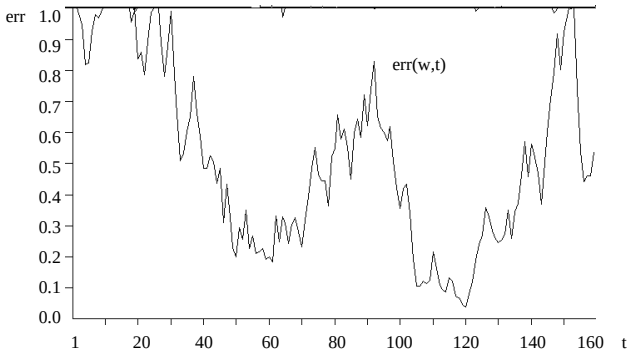


Fig. 11 The error of pure associative learning

This brings us to the conclusion that, in order to learn something, we have to include not only the actual pattern performance $R(\mathbf{x}(t)) = R_t$ 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 by $R_t$-$R_{t-1}$>0, otherwise not

$$\mathbf{w}(t) = \mathbf{w}(t\text{-}1) + \gamma(t)\, L(\mathbf{x})\, \mathbf{x} \tag{5.5}$$

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

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

which is a kind of evolutionary learning [33]. In Fig. 12 the error development of such a learning system is shown.

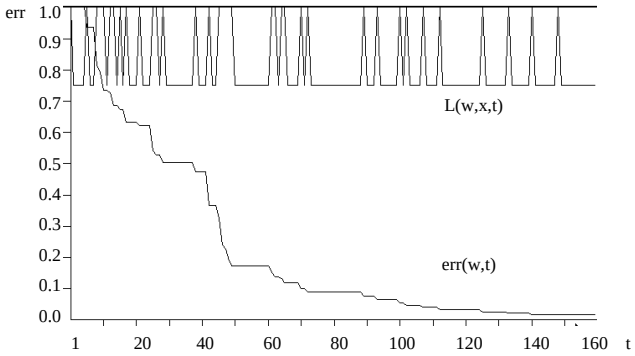


Fig. 12 The error of binary evolutionary associative learning

Each improvement x is a random deviation of the pattern *w* according to a Gaussian distribution with equal width σ=0.3, i.e. $C^{-1}$= $\mathbf{I}\sigma^{-1}$ and γ=0.2. The figure shows additionally as a change indicator the scaled and shifted version of L(t), the function L'(t)=0.75+0.25L(t) which indicates each 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

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

In Fig. 13 the error development of (5.5) using (5.7) instead of (5.6) is shown. 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 neighborhood 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.

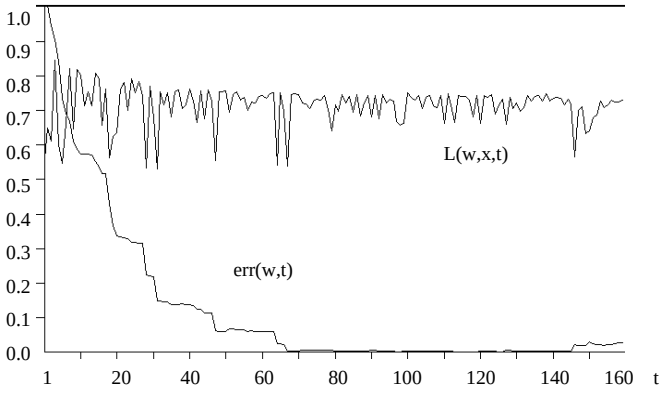


Fig. 13 The error of linear evolutionary associative learning

In Fig. 14 the three algorithms are compared by the random walks they produce.

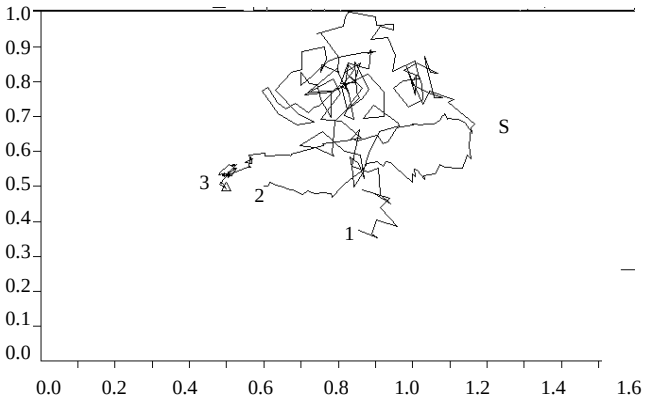


Fig. 14 The random walks of evolutionary associative learning

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] M. Levine, *Vision in man and machine*; McGraw Hill 1985

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

[3] J. Daugman, *Complete Descrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression*; IEEE Transactions on Acoustics, Speech and Signal Processing Vol 36, No 7, pp.1169-1179 (1988)

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

[5] B. Gordon, E. Allen, P. Trombley, *The Role of Norepipene in Plasticity in the Visual Cortex*; Progress in Neurobiology, Vol 30, No.2/3, pp. 171-191 (1988)

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

[7] R. Linsker, *Self-Organization in a Perceptual Network*; IEEE Computer, pp. 05-117, (March 1988)

[8] H. Haken, *Information and Self-Organization*; Springer Verlag Berlin Heidelberg 1988

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

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

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

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

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

[14] 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

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

[16] 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

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

[18] 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

[19] D. H. Ballard, Ch. Brown, *Computer Vision*, Prentice Hall 1982

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

[21] R. Brause, *Neuronale Netze*, Teubner Verlag 2nd ed., Stuttgart 1995

[22] T. Sanger, *Analysis of the Two-Dimensional Receptive Fields Learned by the Generalized Hebbian Algorithm in Response to Random Input*; Biol. Cybernetics, Vol 63, pp.221-228 (1990)

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

[24] 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.
[25] J. Shapiro, A. Prügel-Bennet, *Unsupervised Hebbian Learning and the shape of the Neuron Activation Function*; in, I. Aleksander, J. Taylor (eds.), Art. Neural Netw. 2, Elsevier Sc. Publ. 1992, pp. 179-182
[26] H Marko, *Die Systemtheorie der homogenen Schichten*; Kybernetik, Vol.5, pp. 221-240, (1969)
[27] P.P Vaidyanathan, *Multirate Systems and Filter Banks*; Prentice Hall PTR, Englewood Cliffs, New Jersey 1993
[28] C.E. Shannon, *Communication in the presence of noise*; Proc. of the IRE, Vol. 37, pp.10-21, Jan. 1949
[29] S. Mallat, *A Theory for Multiresolution Signal Decomposition, The Wavelet Representation;* IEEE Trans. Pattern Anal. Mach. Intell., 11, 1989, pp. 674-693
[30] M. Groover, M. Weiss, R. Nagel, N. Odrey, *Industrial robotic*s, McGraw Hill, 1986, pp.504
[31] T.Kohonen, *Correlation Matrix Memories;* IEEE Transactions on Computers, Vol C21, pp.353-359, (1972)
[32] D.E.Rumelhart, J.L.McClelland, *Parallel Distributed Processing;* Vol I, MIT press, Cambridge, Massachusetts 1986
[33] Ingo Rechenberg, *Evolutionsstrategie;* problemata frommann-holzboog, 1973