Sparsely Connected Hopfield Networks for the Recognition of Correlated Pattern Sets[‡]

Thomas Stiefvater
M.P.W. Lasec
Anklamerstr. 32
D-1040 Berlin

Klaus-Robert Müller
GMD FIRST
Rudower Chaussee 5
D-1199 Berlin
klaus@first.gmd.de

Herbert Janßen
Institut für Neuroinformatik
Ruhr-Universität Bochum
44780 Bochum
Universitätsstr. 150
heja@neuroinformatik.ruhr–uni–bochum.de

Abstract: A sparsely connected Hopfield network for the recognition of natural, highly correlated video images is proposed. A general design mechanism for the construction of a local neighbourhood structure using a statistical analysis of an arbitrary given pattern set is suggested. The duality between learning and dilution is employed and different learning respectively dilution schemes are discussed. The practical use and the efficiency of the model are shown in simulations of a large network (N=12288). We use a set of natural patterns with high inter pattern correlations and a high site correlation within each pattern, in which the correlations are given and not constructed by special rules as for highly correlated random pattern sets. The results obtained are analysed for different coding types of the binary pattern set.

1 Introduction

Up to now Hopfield networks have mostly been investigated in the context of statistical mechanics for random patterns [1, 2, 5]. In this work we would like to show how to construct a sparsely connected associative memory model of the Hopfield type for storage and retrieval of a large set of preprocessed video images. We have seen in our study that due to the unfavourable correlations in our natural data set - which we think are typical for real world data - standard models proposed for the processing of correlated random patterns fail to work. They cannot handle high correlations which are not constructed.

Our pattern set has high inter pattern correlations and high site correlations within each pattern. The inherent local correlations of the patterns are on one side caused by the special coding and filtering operations used for preprocessing [13] and on the other side they are a quite general property of real world patterns, due to the spatial and temporal continuity of nature.

In our opinion, it is very important to deal with models, which are not only accessible to statistical methods, but which are also able to serve a practical purpose. This means that a model has to consider the non-ideal properties of typical real world data sets. Practical usability also means, that the relaxation times should be of the order of a few seconds on a usual workstation and that the learning times should be reasonable. This imposes a number of constraints on the type of learning

[‡]Published in: Network 4 (1993), pp. 313 – 336

one can use and as well on the model itself. Also iterative learning algorithms are to be preferred, since they are applicable to a broader spectrum of problems. The most important constraint is efficiency, which is usually neglected since the system size used to see a generic behaviour of a model is $N \sim \mathcal{O}(10^2)$. Therefore we propose a model which is on one side efficient enough to process $N \sim \mathcal{O}(10^4)$ and on the other side able to handle the disadvantageous correlations found in a real world problem.

Having in mind that diluted models [3] are rather efficient for processing random patterns, and that learning and dilution are dual methods to "metastabilize" a given pattern set [4], we conclude that these techniques should be even more appropriate for patterns with a naturally given correlation structure, since the dilution can be adapted to this structure.

For random patterns it was demonstrated that annealed dilution strategies, in which cutting is done in correlation with a given pattern set, produce basins of attraction which are larger than those found in networks with a uniform geometric neighbourhood structure [11, 12]. This is clear because in this case important long-range interactions - typical for random patterns - are cut and only synapses in a local neighbourhood are left over. So an interresting question is: are the high valued, important synapses - i.e. the synapses actually picked by annealed dilution - in the case of a natural pattern set consistent or even identical with the choice of a certain local neighbourhood geometry.

If so, we would have a powerful learning rule at hand, a learning by specifying a certain dendritic field or network geometry equivalent to the annealed cutting of the meaningless synapses.

This concept would also correspond more to biology, since "geometrical computation" - i.e. computation by consistently structured connections - seems to be one of the most important principles of neural information processing [17].

Moreover, designing a certain local dendritic field is a simple and efficient way of learning, which does not require more evolved iterative learning algorithms like pseudoinverse or perceptron learning. Finally, a network with local geometry is much easier to implement in parallel hardware.

We developed a number of heuristics to design this local network architecture based on a statistical analysis of the natural pattern set and it is shown that the appropriate choice of a local neighbourhood indeed selects the high valued synapses. Three new learning approaches are considered in the course of our work, a geometric one, a site statistics dependend one and a combination strategy.

As a performance measure of the Hopfield network, we use a scalar product classifier, which gives an information about the best possible recognition capability in the sense of ideal discrimination of the original stored patterns. Of course due to its linearity such a classifier lacks the ability of autoassociation, self-organization of classes and nontrivial class boundaries. Since it is in general not clear how to measure the classification or discrimination and the retrieval capabilities of an autoassociative memory, we propose three different measures in order to solve this task (appendix) and it is shown in our investigations, that a Hopfield network with the appropriate neighbourhood structure is performing comparably well to the ideal scalar product classifier.

We would like to mention again at this point that our study is neither limited to the special natural pattern set we used nor to the kind of preprocessing, as long as the patterns are sparse and binary. It is a rather general ansatz which provides techniques helping to construct an efficient associative memory even if the correlations are as high as in our present case.

In the following, we will proceed to describe the pattern set and the statistical analysis in section

2. In section 3 we will briefly review the sparsely coded Hopfield model. The dual procedures of learning and dilution as well as the recognition results are discussed in part 4 and a conclusion is given in the last section.

2 The Pattern Set

2.1 Data Set Properties

As a test set, we use a typical "natural" pattern set as it was generated by an image recognition application for a robot mounted camera system implementing a model for saccadic object recognition [13, 14, 15]. Each pattern was created by binary coding of a set of 2 dimensional gabor filter responses to a video image. The images contain various scenes from a laboratory (see also fig.5, fig.9, fig.11 and fig.12).

A pattern is a binary string (12288 bit long) which contains a number of 1024 substrings (12 bit long), each used to code an image feature (orientation) at a specified position in the image. For that, any correlation of image features will result in a correlation of substrings. Each substring uses a k ($k \in 1 \ldots 4$) out of 12 code to specify the orientation of the dominating edge (brightness discontinuity) in an image region. The coding always results in $k-1$ bits being neighboured on the substrings in the sense of a periodic geometry. This effects a good recognition (high overlap) of such images, which only differ by a slight shift of orientations.

The resulting inherent structure of every pattern is 3–dimensional: The set of 12 bit substrings contains the 2–dimensional spatial structure of an image, while each substring contains the 1–dimensional periodic structure of the chosen feature (edge orientation).

For the evaluation of the recognition reliability, the original image set is disturbed by different levels of white noise. By this, we gain a disturbance, which is not statistical in the sense of the binary patterns used, but is structured in the same way as the patterns themselves.

The average mutual correlation of the pattern set is $\langle m^{\mu\nu} \rangle_{\mu \neq \nu} > 0.89$, where $m^{\mu\nu}$ is given by

$$m^{\mu\nu} = \frac{1}{Nb} \sum_i \xi_i^\mu \xi_i^\nu \tag{1}$$

and $\xi_i^\mu \in \{0, 1\}$ ($i = 1 \ldots N, \mu = 1 \ldots p$) are sparse binary patterns with fixed global sparseness, i.e. $b = 1/N \sum_i \xi_i^\mu$. This data set with $p = 461$ patterns ($k = 3$) still contains some pattern subsets with correlations $m^{\mu\nu} > 0.99$. Note that the correlation measure (1) considers only the number of coinciding active bits in pattern μ and ν, the zeros are not taken into account. Therefore $\langle m^{\mu\nu} \rangle_{\mu \neq \nu} > 0.89$ means that on average only 340 bits are different in two patterns for $b = 1/8$.

By removal of all patterns with a correlation bigger or equal a given threshold m_{max}, we generate a number of lower correlated pattern sets with a smaller number of patterns and smaller average mutual correlation. This allows us to get at least some reasonable systematic results for models like the fully connected Hopfield model which are not especially well performing for highly correlated natural pattern sets. m_{max} gives us also a measure for the maximal admissible correlation.

2.2 Statistical analysis

For a natural pattern set in which the correlations are not known in terms of probability distributions it is important to get at least some knowledge about the clustering of the patterns in phase space and their correlation structure. Moreover to show that the appropriate choice of a local neighbourhood $\mathcal{M}(i)$ indeed contains the necessary high valued synapses for our natural pattern set, we have to demonstrate that on average most of the important synapses can be found in a neighbourhood $\mathcal{M}(i)$ of neuron i.

First we neglect that we actually deal with 2–dimensionally structured data, i.e. we just handle the problem of storing a set of 1–dimensional vectors ξ_i^μ with a topology given by the distance between two neuron indices $d(i,j) = |i-j|$. Now we calculate the correlation J_{ij} as in eq.(6) but averaging over all equally distant weights J_d

$$J_d = \frac{1}{n_d} \sum_{\substack{i,j \\ d(i,j)=d}} J_{ij}, \tag{2}$$

where $n_d = card\{J_{ij} \mid |i-j| = d\}$. J_d is a quantity which tells us whether there is a correlation between two pattern sites with distance d averaged over all patterns and all possible sites i and j with $d(i,j) = d$. A graph for J_d is given in fig.1 for the natural pattern set and in fig.2 for random patterns, for $d(i,j) = 1, \ldots, 384$.

In this context the random patterns are to be constructed in the same structure as the preprocessed natural patterns. This means for a k out of 12 code with activity b we choose $b*N/12*k$ substrings for every vector and place randomly a k-block into the respective substring (cf. fig.5c).

For the natural pattern set we can see the regular 12–fold periodic structure resulting from the 12 possible substrings (orientations) and the synchronously active neuron blocks caused by the k-block coding. Both figures show that the $k-1$ nearest neighbour connections are the most important correlations, due to the block structure of the pattern sets. Further significant structures can be seen in fig.1, where the 3–block code is shown ($k = 3$) for natural patterns.

In the overall average, the natural pattern set shows a preference for vertical lines or interactions between those orientations. We see on average all $J_{ij} > 0$ with $|i-j| < 3$ in both sets, while for random patterns $J_{ij} \sim 0$ for $|i-j| > 12$. This is what we expect to find since there is no strong short range correlation beyond the 3–block structure for random patterns, while for natural images some smoothness conditions for neighbouring orientations have to be fulfilled.

The *second* and "correct" way to analyse the geometric correlation structure inherent to our pattern set is to apply the norm $\widehat{d}$, induced by the 3–dimensional periodic data structure. The "correct" structure of the analysis is then determined by the data structure of the pattern contents, in the sense of the underlying feature preprocessing.

As mentioned above, the data structure is a 32×32 grid, with an additional periodic dimension representing the 12 orientations, attatched to every point of the grid. So we have to do the same averaging $J_{\widehat{d}}$ as in eq.(2) but now with the 3–dimensional "manhattan" metric $\widehat{d}$ induced by the data structure:

$$\widehat{d}(i,j) = \Big(|x(i) - x(j)|\,,\ |y(i) - y(j)|\,,\ \min\{|z(i) - z(j)|,\ 12 - |z(i) - z(j)|\} \Big). \tag{3}$$

$x(i)$ and $y(i)$ are the transformed indices corresponding to the 2–dimensional image geometry, while

$z(i)$ is the orientation index (cf. fig.13). The normalisation $\widehat{n_d}$ is chosen accordingly. An implicit asumption in $\widehat{d}$ is of course that x, y and the orientation coordinate z are weighted equally. Since the orientations are 12–periodic we have a hypercylinder geometry. Fig.3 shows that this is a better approximation to the pattern structure.

We also see that for distances $|x(i) - x(j)| > 4$ or $|y(i) - y(j)| > 4$ (at the same orientation) the correlation value is vanishing. This means that the "effective" subspace where correlations actually occur can be taken from fig.3. In order to get a better knowledge about the regular structure in fig.1–3 - which is probably both an effect coming from the common laboratory background of the video images and from the smoothness conditions in natural scenes - we calculate the average site activity (cf. fig.4b).

$$b_i = \frac{1}{p} \sum_{\mu} \xi_i^{\mu}. \tag{4}$$

We find both a strong periodic structure and a high activity in the left and right center area of the image. At this point we see that we actually have to deal with a superposition of two special, but typical problems:

- The first point is that our pattern set contains sequences of very highly correlated patterns (e.g. images for which the camera was shifted only for a small angle or portraits with different facial expressions). The scalar product may be even larger than 0.99 in some of these subsets (cf. fig.5b, fig.11c,d and fig.12).
- The patterns are locally correlated due to the underlying spatial correlation of the images and the structure of the coding. This gives rise to a very strong redundancy (cf. $\bar{\xi}_i$ fig.4b, fig.5a and fig.11a).

Both properties are typical not only for the pattern set we used, but for most patterns gained by coding of natural features. The correlations in the pattern sets appear as a result of the non-randomness of choosing what is interesting for recognition in the real world as well as from the preprocessing of the acquired feature data.

A very straight forward method to deal with sequences of similar patterns or common subpatterns is to simply refuse the learning of new patterns with an overlap of $m > m_{max}$. This is of course decorrelation by brute force but it will depend on the recognition task whether it makes sense to try to really distinguish between patterns with a scalarproduct close to 1.0.

It is clear that a learning algorithm for a neural network should have the ability to handle both problems, i.e. it should supply a decorrelation of common subpatterns (background) and a decorrelation of subsets of similar patterns.

If we deal with highly correlated patterns, we should furthermore be able to solve two tasks. The first would be to distinguish between all patterns no matter how similar they are and store them as individual attractors in phase space. The second task is to provide a model which is capable of condensing sequences of resembling images to one class (cf. fig.12 (a)-(c)).

3 The Model

3.1 A Low Activity Network For Random Patterns

The patterns $\xi_i^\mu \in \{0,1\}$ of our application are highly correlated, sparsely coded binary patterns with average activity b,

$$b = \frac{1}{N}\sum_i \xi_i^\mu. \tag{5}$$

For random patterns there exists a class of low activity associative models [8, 9, 10, 14]. These models are fully connected and they show a relatively high storage capacity, which is actually not exploited in our application. The critical storage capacity α_c of low activity networks is inverse proportional to the pattern activity $\alpha_c \sim 1/b\ln b$, this holds for random patterns. The synaptic couplings are chosen to be

$$J_{ij} = \frac{1}{Nb(1-b)}\sum_{\mu=1}^{p}(\xi_i^\mu - b)(\xi_j^\mu - b). \tag{6}$$

In priciple eq. (6) is already a first approximation to the pseudoinverse [10]. The straight forward use of the original Hebb rule was not suitable here, since low activity patterns should be represented correctly {active: $1-b$, inactive: $-b$} as in (6). They are correlated because their sparseness and their distribution p which is given by the analytical expression

$$p(\xi_i^\mu) = (1-b)\delta(\xi_i^\mu) + b\delta(\xi_i^\mu - 1), \tag{7}$$

where δ is the usual δ function. In our study we use a parallel deterministic dynamics

$$s_i = \Theta(h_i - \vartheta), \tag{8}$$

where $\vartheta = 1/2 - b$ is the first order approximation to the optimal threshold [8], $s_i \in \{0,1\}$ and the local field h_i is given by the usual weighted sum over all neighbouring neurons

$$h_i = \sum_{j\neq i}^{N} J_{ij}s_j = \frac{1}{Nb(1-b)}\sum_{j\neq i,\mu}(\xi_i^\mu - b)(\xi_j^\mu - b)s_j. \tag{9}$$

The overlap m^μ which shows to what extent pattern μ is retrieved, is given by comparing the number of identical active bits in pattern μ and the state vector s_j, i.e. by computing the ordinary scalar product

$$m^\mu = \frac{1}{Nb}\sum_j \xi_j^\mu s_j. \tag{10}$$

The measure (10) means that one does not consider the zeros in the pattern vectors as "information carrying elements", i.e. only the active bits contribute to similarity. Of course (10) is not a unique measure for the retrieval of a pattern μ, so we also used the number of wrong bits

$$err^\mu = \frac{1}{Nb}\sum_j |\xi_j^\mu - s_j| \tag{11}$$

as a check for correct association in our investigations.

3.2 The Model For Natural Patterns

For natural patterns we are obviously not able to give such a closed expression (7) for the pattern statistics, furthermore no martingale property as for hierarchically correlated patterns [10] can be found, because the correlations of the natural pattern set are not *constructed* but *given*. Therefore there also exists no efficient learning rule as for hierarchically correlated patterns and a calculation of ϑ by means of a signal to noise analysis or the replica trick is no longer possible. We now make use of eq.(6) although we know that it is not the appropriate way of learning for our video image patterns, but a signal to noise analysis shows a feasible way to refine the model. For this we calculate the local field $h_i(s_i = \xi_i^1) = h_i^1$

$$\begin{aligned} h_i^1 &= \frac{1}{bN(1-b)} \sum_{\mu=1}^{p} (\xi_i^\mu - b) \left(bN m^{\mu 1} - b \sum_{j=1}^{N} \xi_j^1 - (\xi_i^\mu - b)\, \xi_i^1 \right) \\ &= \frac{1}{bN(1-b)} \left(\xi_i^1 - b \right) \sigma^1 + \frac{1}{bN(1-b)} \sum_{\mu=2}^{p} (\xi_i^\mu - b)\, \sigma^\mu \end{aligned} \tag{12}$$

where

$$\sigma^\mu = \left(bN m^{\mu 1} - b \sum_{j=1}^{N} \xi_j^1 - (\xi_i^\mu - b)\, \xi_i^1 \right) \tag{13}$$

and

$$\sigma^1 = \begin{cases} bN(1-b) & \text{for} \quad \xi_i^1 = 0 \\ (bN-1)(1-b) & \text{for} \quad \xi_i^1 = 1 \end{cases},$$

where $m^{\mu 1} = m^\mu(s_i = \xi_i^1)$. The first term in eq.(12) is the signal term S_i trying to stabilize ξ_i^1

$$S_i = \frac{1}{bN(1-b)} \left(\xi_i^1 - b \right) \sigma^1$$

while the noise term R_i corrupts the signal

$$R_i = \frac{1}{bN(1-b)} \sum_{\mu=2}^{p} (\xi_i^\mu - b)\, \sigma^\mu \tag{14}$$

For equally distributed random patterns the two approximations

$$\sigma^\mu \sim \sqrt{bN} = c \tag{15}$$

$$\sum_{\mu=1}^{p} \xi_i^\mu \sim pb \tag{16}$$

hold below the critical storage capacity, therefore the noise term becomes rather small. As soon as we deal with natural patterns this is no longer true because the average site activities b_i vary strongly, as we see in fig.. 4b and fig. 8a. So we have to represent the pattern activity correctly, i.e. $\xi_i^\mu \in \{1 - b_i, -b_i\}$ for active rsp. inactive pixels. Assuming σ^μ for $\mu \neq 1$ to be constant for the moment, would yield a vanishing noise term.

$$R_i = \frac{1}{bN(1-b)} \sum_{\mu=2}^{p} (\xi_i^\mu - b_i)\, \sigma^\mu \tag{17}$$

σ^μ is of course far from being constant or small for natural patterns, since we have sequences of similar patterns and spatial correlations (fig.5a,b). So how do we get σ^μ to be constant? In principle we have to find a neighbourhood set $\mathcal{M}(i)$ for every neuron i on which all σ^μ are decorrelated or constant.

$$\sigma^\mu = \left(\tilde{m}^{\mu 1} - \sum_j b_j \xi_j^1 - (\xi_i^\mu - b_i)\, \xi_i^1 \right) \qquad \text{for } j \in \mathcal{M}(i) \tag{18}$$

where

$$\tilde{m}^{\mu 1} = \frac{1}{zb} \sum_j \xi_j^\mu \xi_j^1 \qquad \text{for } j \in \mathcal{M}(i).$$

We use $z = card\{\mathcal{M}(i)\}$ for the number of connections leading to neuron i. This is equivalent to finding a subspace $\mathcal{M}(i)$ for every neuron where the σ^μ are distributed homogenously, i.e. they behave similar to random patterns.

The subspace $\mathcal{M}(i)$ can be found by heuristic. A direct iterative decorrelation algorithm is both too time consuming and takes an unacceptable amount of main storage for our application. In the following we develop a number of more heuristical learning procedures with a local representation of the pattern activities based on hebbian learning on a neighbourhood $\mathcal{M}(i)$

$$J_{ij} = \frac{1}{zb(1-b)} \sum_{\mu=1}^{p} (\xi_i^\mu - b_i)(\xi_j^\mu - b_j) \qquad \text{for } j \in \mathcal{M}(i). \tag{19}$$

The local threshold is choosen to be $\vartheta_i = 1/2 - b_i$.

4 Learning and Retrieval Results

We consider the fully connected architecture according to eq. (6) as the standard case. Clearly eq. (6) will work correctly as long as the pattern set has low average correlations, in this case natural patterns behave similar to random patterns. Above a certain memory loading respectively a certain average correlation of the pattern set, the standard learning will fail to work for non-random patterns in a fully connected network. This also holds for a fully connected network using hebbian learning with a local patten representation as in eq. (19).

Comparing the correlation histograms of random patterns for the complete set of natural patterns ($p = 461$) in the standard model, one can see in fig.6 that the random patterns are associated correctly, while the recognition of the natural pattern set is a disaster: The fully connected network converges to a mixture state $\bar{\xi}$ (cf.fig.4a), which is similar to a superposition of all patterns (cf.fig.4b) and will therefore be called average pattern in the following. In table 1 we see that the highest correlation for which the fully connected network can still stabilize the natural pattern set is $m_{max} < 0.2$ ($p = 35$).

The failure of the fully connected network for natural patterns is of course to be expected from our results in section 2, so one has to find an efficient way to decorrelate the pattern set. The straight forward application of standard iterative learning algorithms as the pseudoinverse or Gardner learning rule [7, 6] is not reasonable for our problem, since it would take an unacceptable amount of computation time and main storage.

From Bouten [4] we have seen that learning and dilution of weights are dual strategies, and we also know that diluted networks can be more efficient than their fully connected counterparts [11, 12]. The reduction of weights – usually meaning a loss of imformation – yields a linear decrease in dynamical complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(zN)$ where z is the number of connections per neuron.

So the main question to ask is, how can we find a heuristic to construct such an efficient network that still performs well, although it is sparsely connected.

In the hebbian case annealed dilution tells us to take only the "meaningful" high valued synapses J^*_{ij} with $\mathcal{M}_\lambda(i) = \{j \neq i || J^*_{ij}| > \lambda\}$so the local field reads

$$h_i = \sum_{j \in \mathcal{M}_\lambda(i)} J^*_{ij} s_j. \tag{20}$$

It is clear that annealed dilution cuts in correlation with the pattern set and it was recently shown for random patterns that such a network performs fairly well even at high levels of dilution [4, 11, 12]. Note that $\mathcal{M}_\lambda(i)$ is not uniform, i.e. for every neuron a special neighbourhood is chosen according to the information in J^*_{ij}. Another quite different way of cutting is to choose a uniform local neighbourhood structure, i.e. the dendritic field of a neuron i contains only the finite set of neighbouring neurons in $\mathcal{M}(i)$.

$$h_i = \sum_{j \in \mathcal{M}(i)} J_{ij} s_j \tag{21}$$

For random patterns the annealed strategy eq. (20) yields better results than a uniform "local geometry" eq. (21), because strong random long-range correlations between neurons occuring in random pattern sets are considered with $\mathcal{M}_\lambda(i)$ but are cut with a local neighborhood $\mathcal{M}(i)$.

Considering efficiency, the model (21) is easy to parallelize while the geometry of (20) is not known in advance in the general case.

However, in case we know the geometry of the correlation structure in the data set, a network with local neighbourhoods $\mathcal{M}(i)$ may come very close to the annealed strategy $\mathcal{M}_\lambda(i)$.

From section 2 we know already that for our pattern set, it is comparable to either define the appropriate neighbourhood set (21) or to pick the meaningful weights (20) (cf. fig.1 and 3). This fact is not too astonishing since natural patterns are known to have strong short-range and weak long-range correlations. It is on the other hand a priori not clear how a diluted network will handle the two correlation types mentioned in section 2.

In the following the idea of learning by dilution is refined and discussed under various aspects until we finally get a good associative memory. We proceed in three steps: In the first step the geometric properties of the correlation analysis (1) of the pattern set are used to define a uniform neighbourhood[1]. Additional iterative learning on the local neighbourhood improves the recognition quality. The second approach takes the site dependency b_i in the pattern set into account: here all synapses leading to sites with low "information content" are cut. In the third approach both "information content" and 2 point correlations are considered for dilution.

[1] Note that the ansatz of a uniform neighbourhood, i.e. for every neuron an identical geometry is used, is in itself an approximation to the choice of the possibly more complicated and non-uniform subspace $\mathcal{M}(i)$ from eq. (19).

4.1 Geometric Learning

The *first* approximation for the network's architecture is taken from the diagramms (1-3) as $\widehat{d} < (4,\ 4,\ *)$ and one can visualize the architecture with the help of fig.13. We simplify this structure even more and choose the neighbourhood of the i-th neuron uniformly as

$$\mathcal{M}(i) = \{j \neq i |\ j = i - \frac{z}{2}, \ldots, i + \frac{z}{2}\} \tag{22}$$

Thus we omitted the information found in the statistical analysis in favor of efficiency. Our simulations showed, that the neighbourhood defined by $\widehat{d} < (4,\ 4,\ *)$ did not yield any results justifying the higher algorithmic complexity. Moreover, $\mathcal{M}(i)$ is certainly not a very perfect approximation to the subspace where the σ^μ from eq. (19) are small or constant. Nevertheless we will follow the ansatz in eq. (22) for simplicity and efficiency reasons and change the neighbourhood in order to obtain an average recognition rate depending on the network geometry and threshold. In order to get systematic results, in principle one would have to run the simulations for all reasonable pairs $\{z,\ \vartheta\}$ (cf. fig.7a). Some exemplifying cases are listed below.

For $z = 8$ the network only recognizes the block structure of the code. The pure patterns cannot be stabilized but the network converges into a spurious state with large overlap to the input pattern. From the point of view of the autoassociator, our network fails completely since it does not stabilize the pure patterns on the neighbourhood. Nevertheless considering classification quality $C_Q^{\pm}$ (for definitions of the classification measures see appendix) and recognition rate C_{rel} the locally connected network is able to recognize the pattern set correctly, i.e. $C_{rel} = 1$ but with only medium quality C_Q. C_{rel} is the relative number of correctly classified patterns and C_Q measures how good the network can distinguish between patterns. We consider a picture ν to be recognized, respectively classified if $C_Q^{\nu +} >> 0$.

As mentioned above the fully connected network converges to the mixture state $\bar{\xi}$ (with large overlap to every pattern) for pattern sets with average correlation $m_{max} \geq 0.2$.

For $m_{max} < 0.2$ the recognition rate of the $z = 8$ network is comparable to the fully connected network, but the fully connected net is able to stabilize the given patterns correctly. For a neighbourhood spreading beyond the block structure as for $z = 24$, the local network fails and recognizes only $\bar{\xi}$, where $C_{rel} = 0$ for every m_{max} (cf. table 1).

So for small z the recognition abilities are bad because of lack of information in general (small number of $J_{ij} \neq 0$), while for a large value of z, the recognition fails due to similarity to the fully connected model. With z in between these boundaries (which are defined by the correlation structure of the data set), recognition is possible, but the final states deviate relatively strong from the learned patterns.

At this point we also see, that the strong simplification of the original neighbourhood $\widehat{d} < (4,\ 4,\ *)$ together with the hebbian learning rule yields a network with high recognition rate C_{rel}, but a final state found in the relaxation process cannot be used by itself.

Since our goal was to have a network with good recognition qualities as well, we choose in the following step a $z = 48$ neighbourhood[2] in order to give a better approximation to $J_{\widehat{d}}$ (cf. fig.3) and try to stabilize respectively decorrelate the pattern set on this larger neighbourhood.

[2] This larger neighbourhood includes more of the original neighbourhood $\widehat{d} < (4,\ 4,\ *)$, i.e. $|x(i) - x(j)| \leq 2$ and $|z(i) - z(j)| \leq 6$, but lines of the picture are still not connected for efficiency reasons (cf. fig. 13).

For this we define a local learning algorithm for low activity patterns. We adapted it to our problem from the nonlinear Abbott-Kepler learning algorithm known for fully connected Hopfield networks [5, 6]. The Abbott-Kepler learning is an adaptive parallel learning rule considered faster than Gardner learning, unlearning or minover [16].

We define a normalized local stability h_i^μ and κ as a parameter to "shape" the basins of attraction

$$h_i^\mu = \frac{\sum_{j\in\mathcal{M}(i)} J_{ij}\xi_j^\mu}{\sqrt{\sum_{j\in\mathcal{M}(i)} J_{ij}^2}} \tag{23}$$

For low activity learning the synaptic weights J_{ij} are changed if a pattern $\xi_i^\mu \in \{0,\ 1\}$ is not stabilized apropriately, i.e.

$$\begin{aligned} (1) \quad & \vartheta_i - h_i^\mu < \kappa \ \text{ if } \quad \xi_i^\mu = 0 \\ (2) \quad & h_i^\mu - \vartheta_i < \kappa \ \text{ if } \quad \xi_i^\mu = 1. \end{aligned}$$

The change of the couplings ΔJ_{ij} is given by

$$\Delta J_{ij} = \frac{1}{zb(1-b)} \sum_\mu F_{nl}(h_i^\mu) \parallel J_i \parallel \ (\xi_i^\mu - b)(\xi_j^\mu - b) \qquad \text{for } j \in \mathcal{M}(i), \tag{24}$$

where the adaptive nonlinear function F_{nl} is defined as

$$\begin{aligned} (1) \quad & F_{nl}(h_i^\mu) = (\kappa + \vartheta_i + \delta - h_i^\mu) + \sqrt{(\kappa + \vartheta_i + \delta - h_i^\mu)^2 + \delta^2} \\ (2) \quad & F_{nl}(h_i^\mu) = (\kappa - \vartheta_i + \delta + h_i^\mu) + \sqrt{(\kappa - \vartheta_i + \delta + h_i^\mu)^2 + \delta^2} \end{aligned} \tag{25}$$

and where δ is a constant choosen to be $\delta = 0.01$ [5]. Thus, we have introduced two new features in the Abbott-Kepler learning algorithm: learning is now performed on a local neighbourhood $\mathcal{M}(i)$ and a case descision has to be made.

In fig.7b we show the dependency of the number of errors

$$err = \sum_{\mu,i} |\Theta(h_i^\mu - \vartheta_i) - \xi_i^\mu| \tag{26}$$

on the number of iterations of (24) and κ. Note that err is not normalized here. The pattern set with $m_{max} < 0.3$ is stabilized on $z = 48$ with $\kappa = 0.1$ in 200 iterations (marked with superscript $^{(conv)}$ in table 1). This takes 7 hours on a Sparc II workstation.

For $m_{max} < 1.0$, $z = 48$ and $\kappa = 0$ we see, that the learning algorithm is not fully converged after 200 iterations, here the average final overlap is found to be $\langle m_f \rangle = 0.95$. The additional learning (24) improves C_Q^+ from 0.23 ($z = 8$) to $0.355^{(200)}$ ($z = 48 +$ additional learning). The ammount of improvement depends of course on the stability parameter κ; above a critical κ_c, both C_Q^+ and C_{rel} break down, because the learning algorithm is not capable of stabilizing the correlated patterns. So sumarizing the results for geometric learning:

- The fully connected network only "recognizes" the average pattern $\bar{\xi}$ for $m_{max} \geq 0.2$.
  The sparsely connected network with $z = 8$ is driven into mixture states with large overlap to the original input patterns. The recognition rate is observed to be $C_{rel} = 1$ for every m_{max}, however the recognition quality C_Q^+ is rather low (cf. table 1).

- We can prevent the network to converge to $\bar{\xi}$ by selecting small local neighbourhoods ($z = 8$) but the variables σ^μ from eq. (19) are still far from being decorrelated or constant on $\mathcal{M}(i)$.
- The iterative learning on an uniform neighbourhood ($z = 48$) provides a suitable ansatz to reduce the remaining correlations to $\bar{\xi}$ and the subsets of similar patterns. This allows a stabilisation of the original patterns.

The above investigations were done for pure input patterns ξ^μ, more interesting and closer to application is the setting in which the network has to associate noisy patterns (cf. table 2).

- Here the fully and geometrically connected models fail completely.
- Additional learning yields an improvement of the network's robustness but although our algorithm converges we can only associate noisy input patterns with $n = 0.062$ (16% noise) to their originals with fairly high values of $\langle m_{fin} \rangle$. Some patterns are associated with $m_{fin} > 0.9$ while on the overall average we observe $\langle m_{fin} \rangle \sim 0.78$, which is higher than the respective value of the $z = 8$ geometry.

Table 1 and 2 give an overview over the different models, i.e. the fully connected Hopfield model, the locally connected Hopfield model and the locally connected Hopfield model with iterative learning regarding retrieval quality $C_Q^{\pm}$ and mean final overlap $\langle m_{fin} \rangle$. As expected we see that the retrieval quality decreases if higher correlations in the pattern sets are considered.

We conclude that the set $\mathcal{M}(i)$ including the highest weights does not provide the suitable subspace on which the σ^μ are decorrelated or constant respectively. Since $J_{\widehat{d}}$ shows that only sites with $\widehat{d} < (4, 4, *)$ are on average correlated, we see that geometric learning is strongly disturbed by the common subpatterns (background) in our pattern set.

m_{max}	# of patterns	fully connected	geometric dilution		dilution + local learning
			z=8	z=24	z=48
$m_{max} < 0.2$	35	0.831	0.570	$\bar{\xi}$	–
$m_{max} < 0.3$	142	$\bar{\xi}$	0.468	$\bar{\xi}$	$0.729^{(conv)}$
$m_{max} < 0.4$	229	$\bar{\xi}$	0.353	$\bar{\xi}$	–
$m_{max} < 1.0$	461	$\bar{\xi}$	0.230	$\bar{\xi}$	$0.355^{(200)}$

Table 1: *Recognition rates C_Q^+ (columns 3-6) for the different geometric learning rules and pattern sets with different maximal correlation m_{max}. The entry $\bar{\xi}$ indicates that the network converges to the average pattern $\bar{\xi}$. $C_{rel} = 1$ holds for all diluted models.*

We also see, that the idea of taking only the highest weights (i.e. the hebbian limit of annealed dilution) without further iterative learning cannot be transfered in a straight forward manner to a natural pattern set.

model	$n = 0.000$			$n = 0.062$			$n = 0.125$			$n = 0.250$		
	C_Q^+	C_Q^-	$\langle m_{fin} \rangle$	C_Q^+	C_Q^-	$\langle m_{fin} \rangle$	C_Q^+	C_Q^-	$\langle m_{fin} \rangle$	C_Q^+	C_Q^-	$\langle m_{fin} \rangle$
$z = 8$ (g)	0.468	–	0.60	0.394	–	0.57	0.334	–	0.53	0.244	0.005	0.48
$z = 48$ (l)	0.729	–	1.00	0.439	–	0.78	0.333	0.02	0.69	0.213	0.007	0.65

Table 2: *Comparison of the different models proposed considering robustness for $m_{max} < 0.3$. The noise strength n in the upper row gives the relative σ of the gaussian noise in the video image (see appendix). "-" for C_Q^- indicates that there were no false classifications and $\langle m_{fin} \rangle$ refers to the size of the final overlap. Again the symbol $z = 8$ (g) stands for geometric dilution, while $z = 48$ (l) implies geometric dilution plus local learning.*

4.2 Site Statistics Dependend Learning

From our results we see that some of the high weights are "highly" redundant. They are strong because similar patterns have strong correlations at certain sites, so in order to get the positions where the redundancy occurs, we use the site statistics b_i obtained in section 2 (cf. fig.4, eq. (4)). In fig.8 we show the distribution of b_i for natural patterns (8a) which fluctuates much more than the distribution for random patterns (8b) and which is strongly inhomogenous.

The site statistics b_i provides a good way to cut redundant synapses. A low b_i means that on average there are few active bits sitting at site i and a high b_i means that most of the patterns have an active neuron at this site (cf. fig.3). So the synapses to neurons with b_i close to the average activity b are the more "important ones". With this empirical argument we define a neighbourhood

$$\mathcal{M}_\sigma^b(i) = \{j \neq i \mid |b_j - b| < \sigma\} \tag{27}$$

where only the synapses leading to "high information neurons" are left, σ is the standard deviation of the distribution of the b_i (fig.8). Note that through the choice of $\mathcal{M}_\sigma^b(i)$ the couplings will not be symmetric anymore. The neurons with high b_i values will have only afferent synapses while all efferent synapses are diluted, i.e. they stop contibuting to the local field. The neighbourhood given by (27) suppresses the common subpatterns (background), which give rise to high b_i at certain sites. The network will therefore refuse to store highly redundant information as e.g. patterns strongly correlated to $\bar{\xi}$. So we have found a subspace on which the patterns are better decorrelated as before but on which some patterns, i.e. the ones strongly correlated to $\bar{\xi}$ cannot be stored. Considering efficiency (27) can be implemented by the use of overlaps and bit coding, but its dynamics is slower than for the $z = 48$ network.

Site statistics dependend learning (SSDL) can be used for both tasks mentioned above. Pattern classes can be constructed and the network will associate sequences of similar patterns to these classes. SSDL can also be employed to distinguish between patterns. In this case all patterns are stored as individual attractors in phase space, which is the usual setting for an attractor neural network.

The classes are constructed in the following manner. Starting from a pattern set with low maximal correlation ($m_{max} < 0.2,\ p = 35$), we add a new pattern if the network converges to the zero-state after being presented this new pattern, i.e. if the new pattern is not recognized. If the new pattern

is recognized, the network dynamics maps it onto a pattern which was already learned before, therefore the new pattern would be rejected as a new candidate for learning. This is performed recursively for all patterns of the full pattern set ($m_{max} < 1.0,\ p = 461$) and it gives us a new pattern set from which afterwards all tested patterns, not fully associated, are removed. The zero-state ($s_i = 0$) can be interpreted as an "I can't recognize this pattern" statement. In this sense the network gives no incorrect classification.

model	$n = 0.000$		$n = 0.062$		$n = 0.125$		$n = 0.250$	
	C_Q^+	C_{rel}	C_Q^+	C_{rel}	C_Q^+	C_{rel}	C_Q^+	C_{rel}
SSDL	0.829	0.971	0.834	0.750	0.835	0.540	0.791	0.100
scalar product class.	0.906	1.000	0.698	1.000	0.588	1.000	0.360	0.993

Table 3: *Comparison of SSDL versus scalarproduct for different levels of distortion for* $m_{max} < 0.3$. C_Q^- *is of the order* 10^{-4}.

It is expected that this strategy reduces the redundancy in the pattern set and it turns out that the sequences as e.g. fig.12a-c are mapped onto a class represented by the pattern shown in fig.12a. The learning rule (19) together with our class building algorithm and the neighbourhood $\mathcal{M}_\sigma^b(i)$ allows to stabilize 250 patterns from a set of 461 patterns. This is, as already mentioned, due to the fact that the network is constructing classes, and therefore associates sequences of similar patterns to these classes. A pattern is either retrieved with $m_{fin} > 0.99$ or the network converges to the zero-state. The associative memory can retrieve the original image from a distorted input pattern with $n = 0.125$ gaussian noise in the original pattern or a 30% cut (fig.9 and fig.11e-g). Here we see that a sparsely connected network can be more reliable than a linear classifier, because it is able to map distorted input data to the correct pattern class.

The remarkable robustness against distortion of our site statistics dependend learning (SSDL) is shown in table 3. In this table we use the SSDL strategy without class construction algorithm for $m_{max} < 0.3$ and we see that the network refuses to stabilize 4 patterns (which corresponds to $C_{rel} = 0.971$). Note that these patterns have high overlap to the background. Only a certain percentage of distorted patterns is mapped to the correct original pattern class but with $m_{fin} \sim 1$. This can also be seen by the large values of the retrieval quality. The patterns who are not retrieved are converging to the null state. Comparing Table 3 and 2 we see that the C_Q^+ values for SSDL are larger than for $z = 48$ (local learning), but some patterns cannot be recognized as the lower values of C_{rel} demonstrate. So it will depend on the application whether it is more convenient to recognize all patterns with rather small robustness ($z = 48$ + local learning) or whether it is more important to construct pattern classes and retrieve all patterns perfectly.

4.3 A Combination Strategy

From the statistical analysis $J_{\widehat{d}}$ we know that the correlation structure is short-range, i.e. a decorrelation should take this fact into account and a coarse approximation of the subspace on which we want to decorrelate is therefore given by $z = 48$. As we have seen from the character of our pattern set, not only smoothness conditions of the patterns but also common subpatterns (background) have an influence on $J_{\widehat{d}}$, so it seems reasonable to decorrelate the common subpatterns by SSDL

on a given geometric neighbourhood.

This should in principle be successful since the statistical analysis of the correlation length $\tilde{J}_{\widehat{d}}$ (fig.10) where (27) is taken as a constraint shows the same behaviour as (fig.3). The only difference is the scale of fig.10 which is by a factor of 2 smaller than in fig.3. Such a combination strategy on a small neighbourhood could increase the network efficiency, since the site statistics dependend learning ansatz still uses about $z = N/2$ connections.

Our results however show that the very specialized subspace given by $z = 48$ is too small to yield satisfactory results, a larger subspace will nevertheless slow down the network dynamics too much. Iterative learning could in principle improve the network performance, but an implementation would be both too complex and unefficient.

So we have to drop this ansatz for efficiency reasons.

5 Conclusion

The standard models available can not handle "natural" correlations. Therefore we have proposed a heuristic to construct a locally connected Hopfield network, which is able to recognize a highly correlated natural pattern set perfectly. Its classification performance is similar to the ideal scalar product. It is clear that iterative learning rules operating on a fully connected network to decorrelate the given pattern set or constructive dilution algorithms would be highly inefficient if not impossible for practical purposes. So our ansatz provides not only a simple but an efficient heuristic way of dealing with highly correlated sparse binary pattern sets from applications. A major ingredient to our ansatz is that some knowledge about the nature of the pattern set is taken into account (statistical analysis).

We will again repeat the three construction possibilities.

The *first* geometrical ansatz uses a statistical analysis of the pattern correlations. This provides a good heuristics how to choose the local neighbourhood since it tells us the "distance" of neuron sites which are on average strongly correlated. A local learning on this diluted network structure helps to stabilize the patterns and to increase the retrieval quality. In this case distorted input patterns with noise $n > 0.062$ cannot be stabilized.

The *second* ansatz uses the knowledge about the site statistics, i.e. the probability of finding an active edge orientation at a certain site. Especially high or low probabilities are considered as redundant features of the patterns and are therefore neglected by the network dynamics. This method yields an associative memory of remarkable robustness.

Our *third* approach uses the geometry given by $J_{\widehat{d}}$ ignoring the redundant sites at the same time. For the neighbourhoods studied we did not find convincing results.

We conclude, either we spend a large amount of computing time for learning (ansatz 1 and 3) and we have a low complexity in the dynamics sector if the neighbourhood is small enough or we use a simple learning rule (27) and have a comparably large complexity for the dynamics.

There are in principle two possible applications for our associative memory models. They can serve to increase the reliability of pattern recognition systems, since they are able to reproduce original patterns correctly from highly distorted input data. On the other side they could also be used as modules to reduce the redundancy in a pattern set, since they can associate highly correlated

pattern sequences to the same class. We consider the remarkable robustness and efficiency as the most important features of our network ansatz.

Our future interest will be to gain further understanding of the neural models appropriate for the processing of natural data.

Appendix: Classification Measures

Actually there is no need to define different classification measures, but by introducing them, we are able to answer the following questions.

1. How good is our autoassociator?
2. What is the success rate of the autoassociator as a classifier?
3. How can we build a reliable pattern recognition system?

The three measures for the evaluation of the associative memory properties are the average final overlap $\langle m_{fin} \rangle$, the classification rate C_{rel} and the classification quality $C_Q^{\pm}$. The average final overlap

$$\langle m_{fin} \rangle = \frac{1}{p} \sum_{\mu} m_{fin}^{\mu} = \frac{1}{pbN} \sum_{\mu} \sum_{j} \xi_j^{\mu} s_j \tag{28}$$

indicates how good the autoassociation for a given pure or distorted input state was done. In our case the distortion "noise" can either be gaussian noise in the original image or a cut of a certain percentage of the image.

The relative number of correctly classified patterns C_{rel} is given by

$$C_{rel} = \frac{\text{number of correctly classified patterns}}{\text{total number of patterns}} \tag{29}$$

and the average classification quality $C_Q^{\pm}$ for right and faulty classification

$$C_Q^{\pm} = \frac{1}{p} \sum_{\mu} m^{\mu max1} (m^{\mu max1} - m^{\mu max2}) \tag{30}$$

measuring how good a certain pattern ξ_i^{μ} is distinguished from the other patterns during retrieval. The overlap $m^{\mu max1}$ denotes the best match and $m^{\mu max2}$ is the second best match. We consider a picture ν to be recognized, respectively classified if $C_Q^{\nu +} >> 0$.

From the view point of autoassociators, it is enough to monitor the average final overlap, but since we also want to compare our autoassociator working as a classifier with the results of the scalar product classification, (29) and (30) have to be taken into account. Eq.(29) is the straight forward success rate of the classifier or the autoassociator, but this does not give us a hint of how good the right classification answers really are. If the network has problems to distinguish between patterns, this is answered by (30). A high C_Q^{+} for recognized patterns, and a low C_Q^{-} for wrong

classified patterns is the best setting that can happen to a classifier. In this case it distinguishes good between the correctly classified pattern and all other patterns, while wrong classified patterns are very close to their second best match.

In order to get a "conservatively" working classifier, we take the largest value of C_Q^-, i.e. the best classified faulty output, as a threshold. This defines rate of conservatively correct classificated patterns as the relative number of patterns that are classified with

$$C_Q^+ > \max_\mu C_Q^- \tag{31}$$

Therefore eq. (31) gives us a good criterion for maximizing the accuracy and reliability of a pattern recognition system.

Noisy Patterns

To gain a kind of disturbance which resembles the structure of the patterns itself, we did not apply noise to the binary patterns, but to the video images. We choose gaussian noise with varying σ to change the pixel intensities. In table 2 we give the noise level as the relative size of σ to the maximum amplitude A of the pixels.

$$n = \frac{\sigma}{A} \quad \text{with} \quad A = 255 \tag{32}$$

Therefore the maximum noise value of 1.0 means that the disturbed image has no similarity with the original.

Acknowledgement

K.- R. M. gratefully acknowledges partial financial support by Landesgraduiertenförderung Baden-Württemberg and A. Glenz. This work is part of the Ph.D. thesis of K.- R. M. done at the department of Logics, University Karlsruhe. H.J. is partially supported by the German Federal Department of Research and Technology (BMFT) under Grant No. ITR8800K4. It is a pleasure to thank H. Horner and R. Kühn for valuable discussions.

A Figure Captions

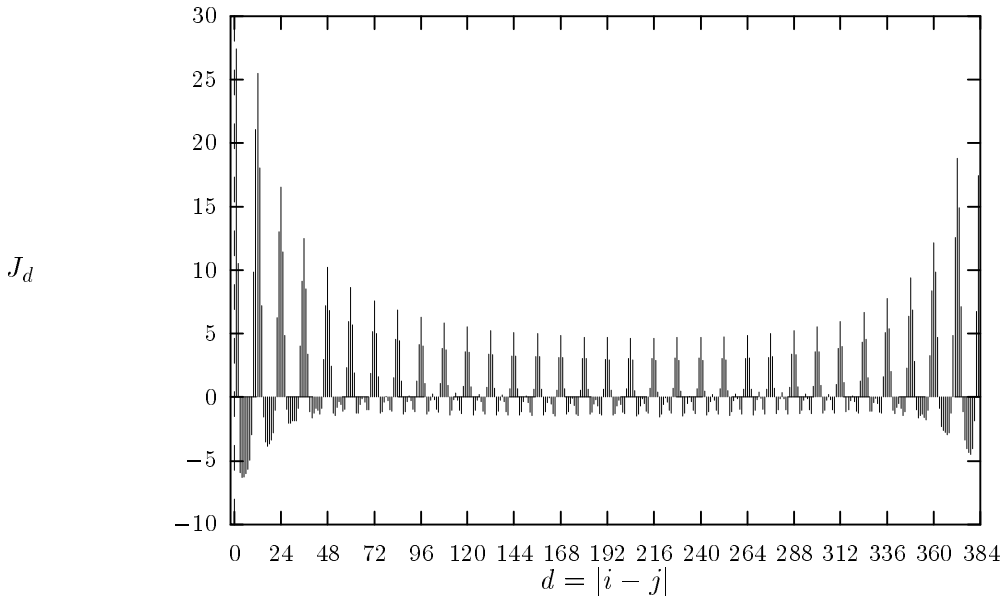


Figure 1: *2 point correlation function for natural patterns, with* $(0,\ 0,\ 0) = 0$ *and* $(0,\ 1,\ 0) = 384$ *on the datacube (cf. fig.13).*

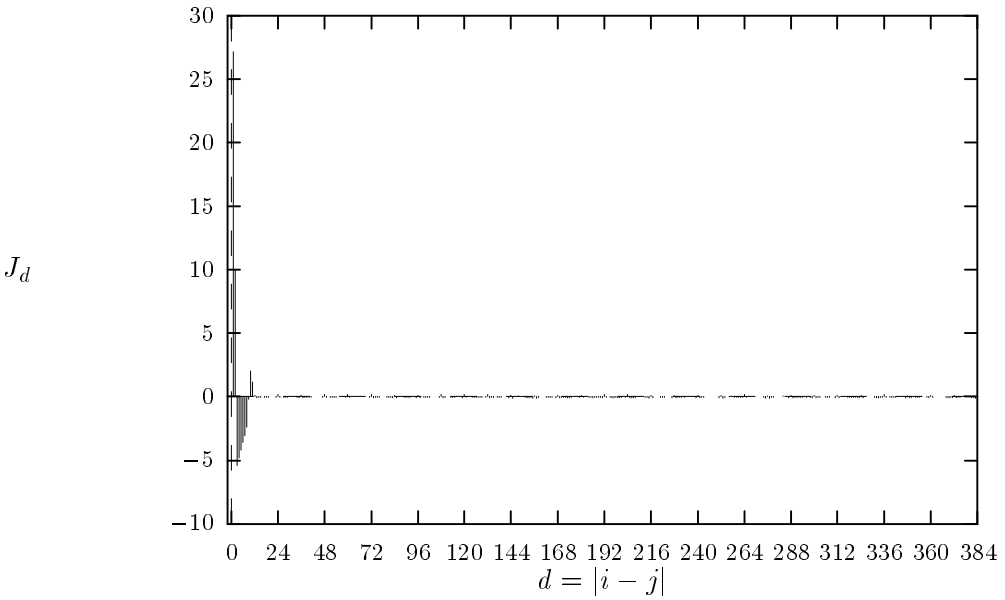
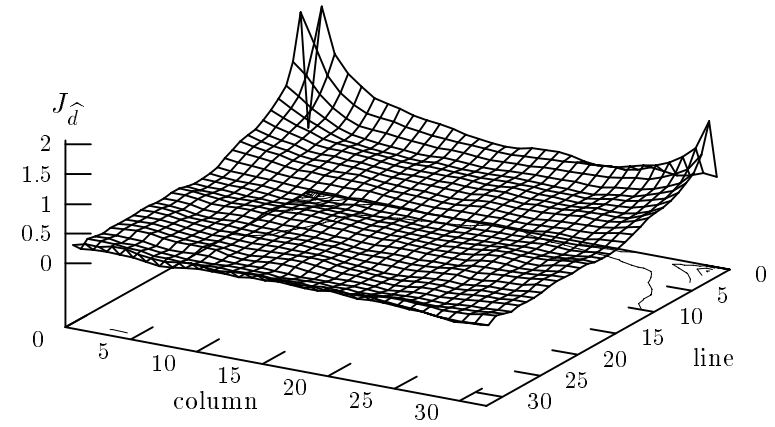
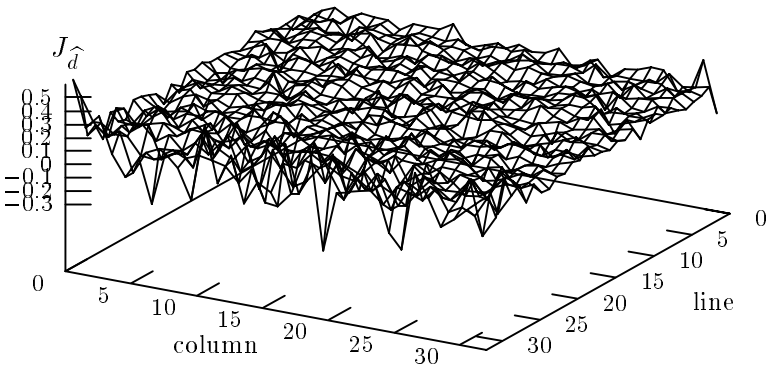


Figure 2: *2 point correlation function for random patterns, with* $(0,\ 0,\ 0) = 0$ *and* $(0,\ 1,\ 0) = 384$ *on the datacube (cf. fig.13).*



(a)



(b)

Figure 3: *2 point correlation function for natural patterns considering the 3 dimensional datastructure where* $\Delta z = |z(i) - z(j)| = 0$*, i.e.* $\widehat{d} = (\Delta x, \Delta y, 0)$ *for (a) natural, (b) random patterns. Note that for all 6 possible constant values of* Δz *we can compute a similar correlation surface as in this figure.*

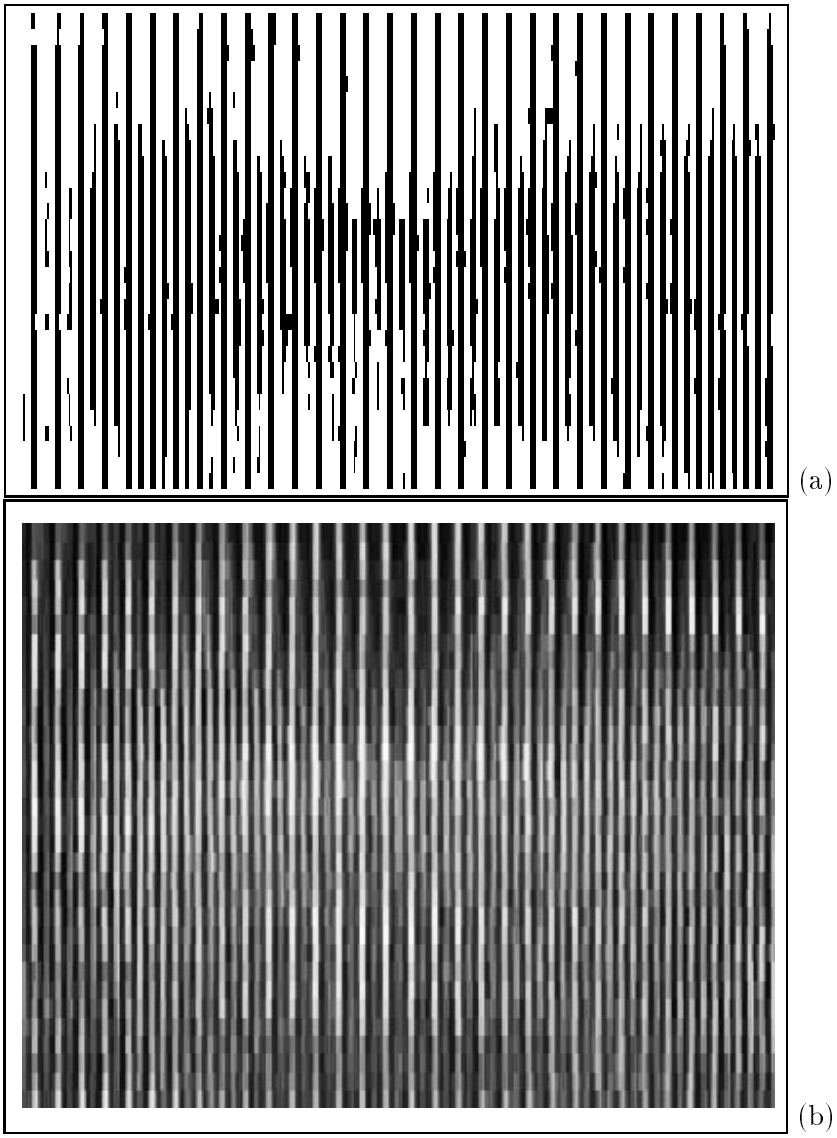


Figure 4: *(a) Final state $\bar{\xi}$ found in a fully connected network, black indicates a one, (b) average site activity b_i, white indicates high values of b_i. In x-direction we plot $0 \ldots 383$, in y-direction $0 \ldots 31$. In the datacube representation this indicates $(0,0,0)$ – $(31,0,11)$ in the first line $(0,1,0)$ – $(31,1,11)$ in the second line and so on. Both images are magnified in the y-direction. Note that both images are very similar, especially they posess a high 12 periodicity. The high values of b_i (white) are situated at comparable sites as the active bits, i.e. the active neurons in $\bar{\xi}$.*

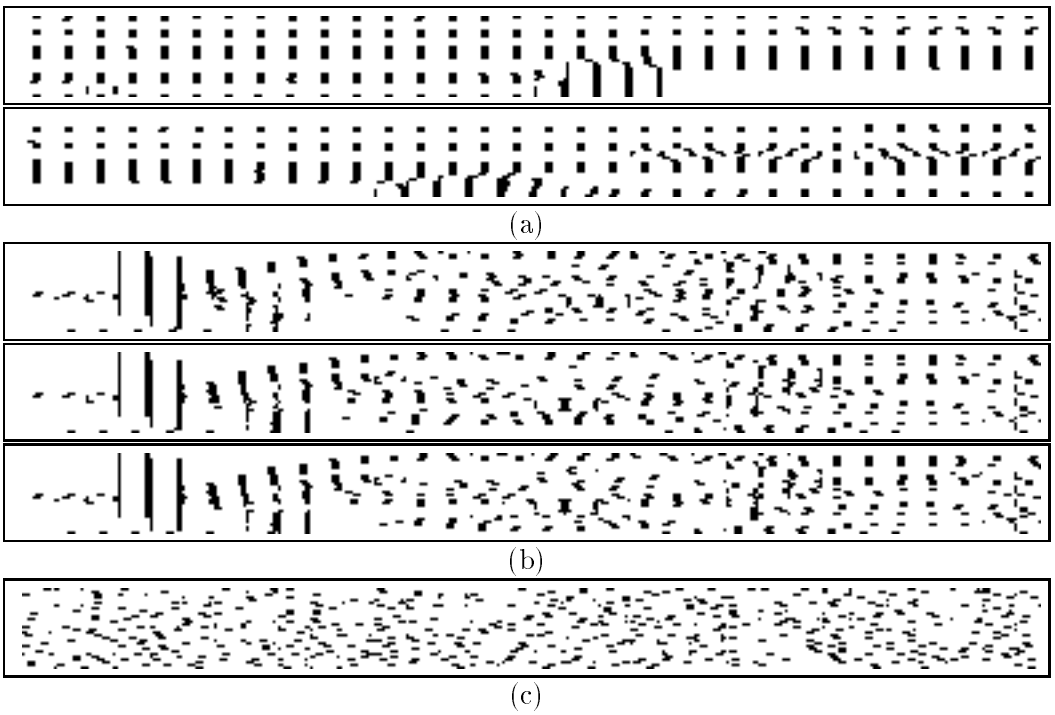


Figure 5: *(a) Example for high spatial correlations in the natural pattern set with dataformat (x: 32*12, y: 32). In the datacube representation this indicates* $(0,0,0) - (31,0,11)$ *in the first line* $(0,1,0) - (31,1,11)$ *in the second line and so on. (b) Example for a subsequence of highly correlated natural patterns, (c) Example for a random pattern generated as described in the text with a 3 out of 12 coding.*

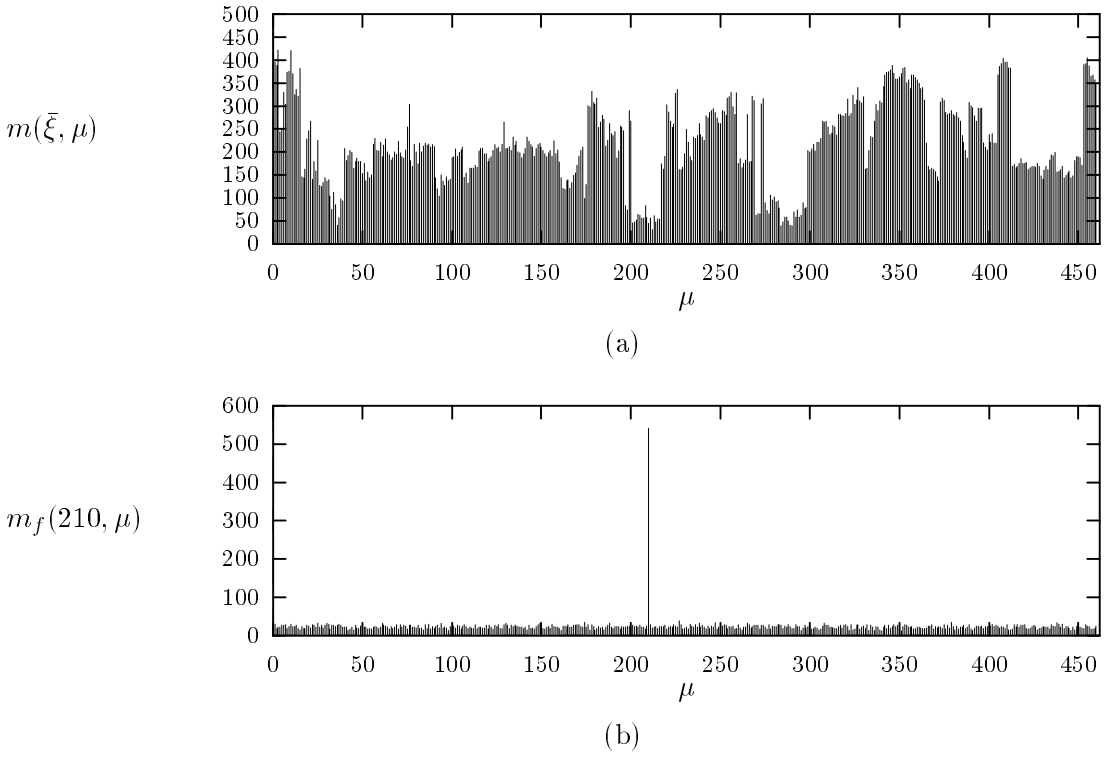


Figure 6: *(a) Correlation histogram of $\bar{\xi}$, the final state found in a fully connected network (1 out of 12 coding) for natural patterns, (b) Correlation histogram of the final state found for random pattern number 210 to all other patterns in a fully connected network (1 out of 12 coding).*

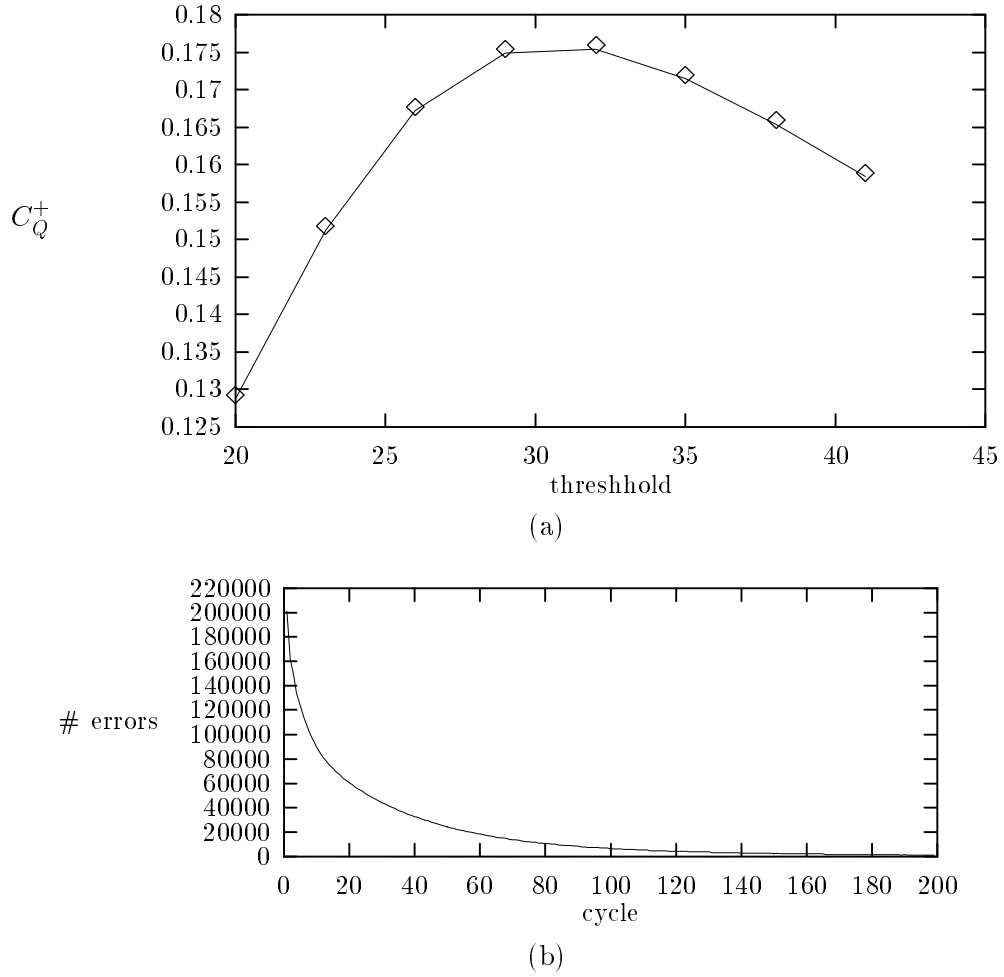
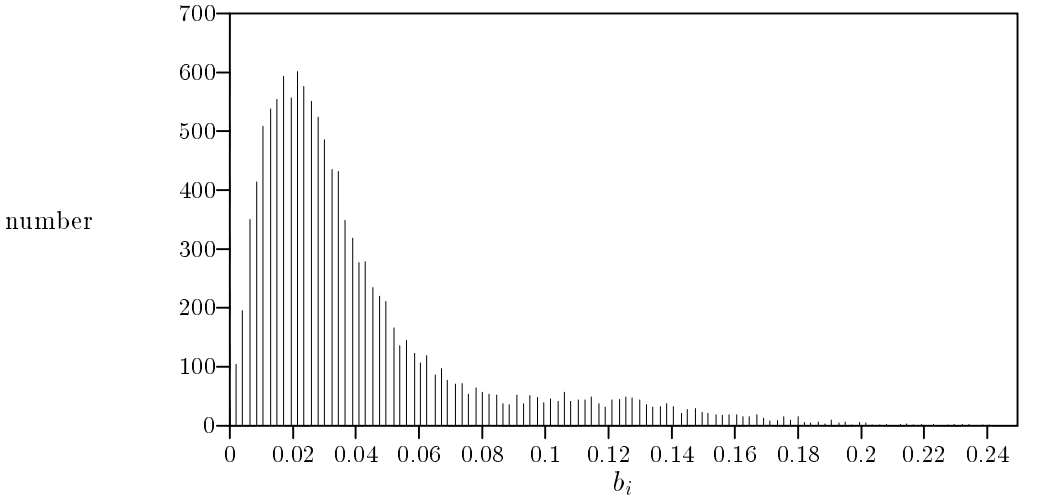
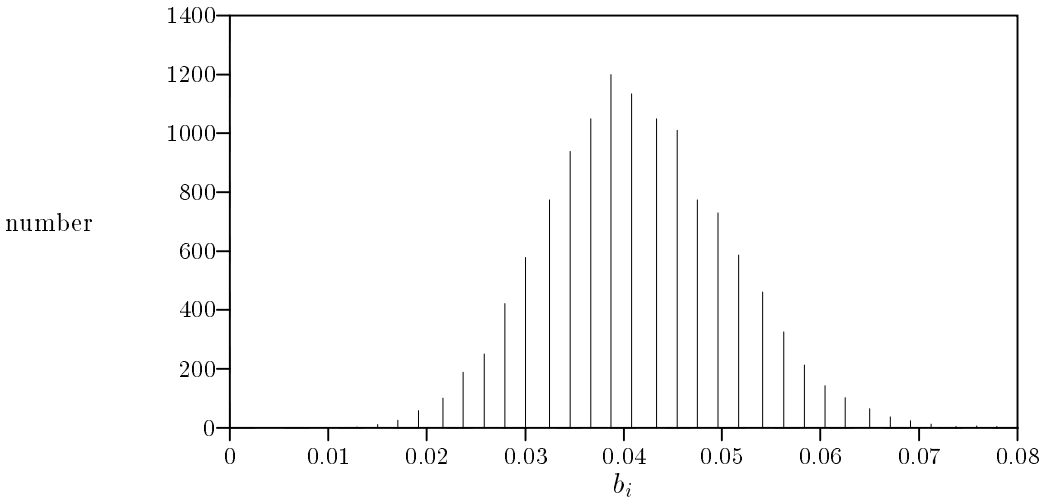


Figure 7: *(a) Retrieval quality C_Q^+ versus unnormalized threshold ϑ for the z=8 network, (b) error rate of the learning algorithm ($z = 48$ network with additional learning) versus number of learning cycles for the parameters $\kappa = 0.1$ and $m_{max} < 0.3$.*



(a)



(b)

Figure 8: *(a) Average site activities b_i for natural patterns, (b) Average site activities b_i for random patterns (1 out of 12 coding).*

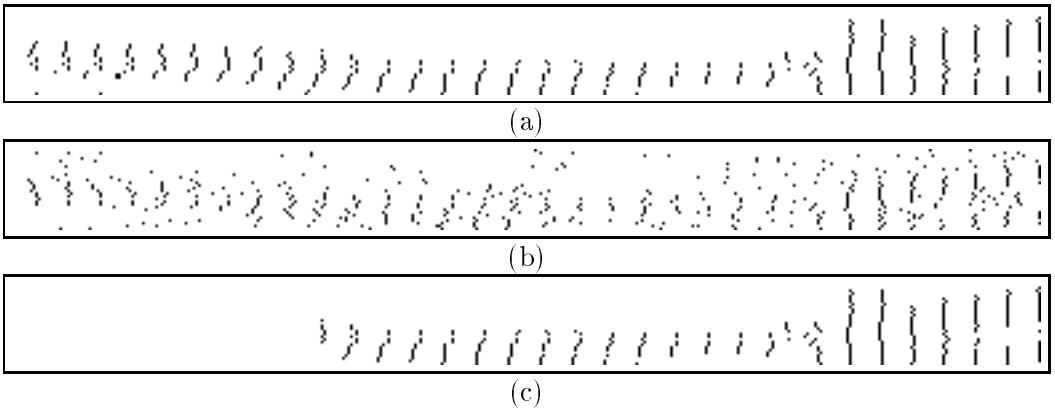


Figure 9: *Comparison of original pattern (a), distorted pattern with gaussian noise in the original pattern (b) and cut pattern (c), both distorted patterns (b,c) are fully associated to (a). Dataformat (x: 32*12, y: 32). In the datacube representation this indicates* $(0,0,0) - (31,0,11)$ *in the first line* $(0,1,0) - (31,1,11)$ *in the second line and so on.*

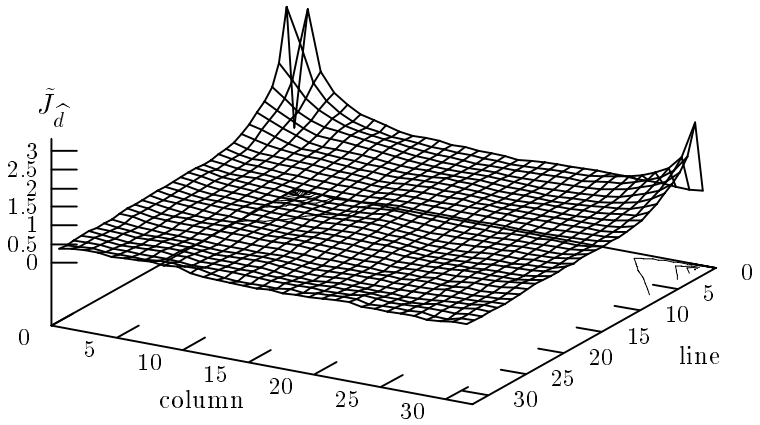


Figure 10: *2 point correlation function for natural patterns considering the constraint* $\mathcal{M}^b_\sigma(i)$ *and the 3 dimensional datastructure (cf. text,* $\Delta z = |z(i) - z(j)| = 0$*).*

Figure 11: *Examples of video images (cf.fig.5) (a) no. 0 (b) no. 3 (c) no. 300 (d) no. 304 (e) no. 17 (original) (f) no. 17 (noisy* $n = 0.125$*) (g) no. 17 (cut to* 50%*).*

(a) (b) (c)

Figure 12: *Example of a typical sequence of video images (cf.fig.5) (a) no. 292 (b) no. 293 (c) no. 294.*

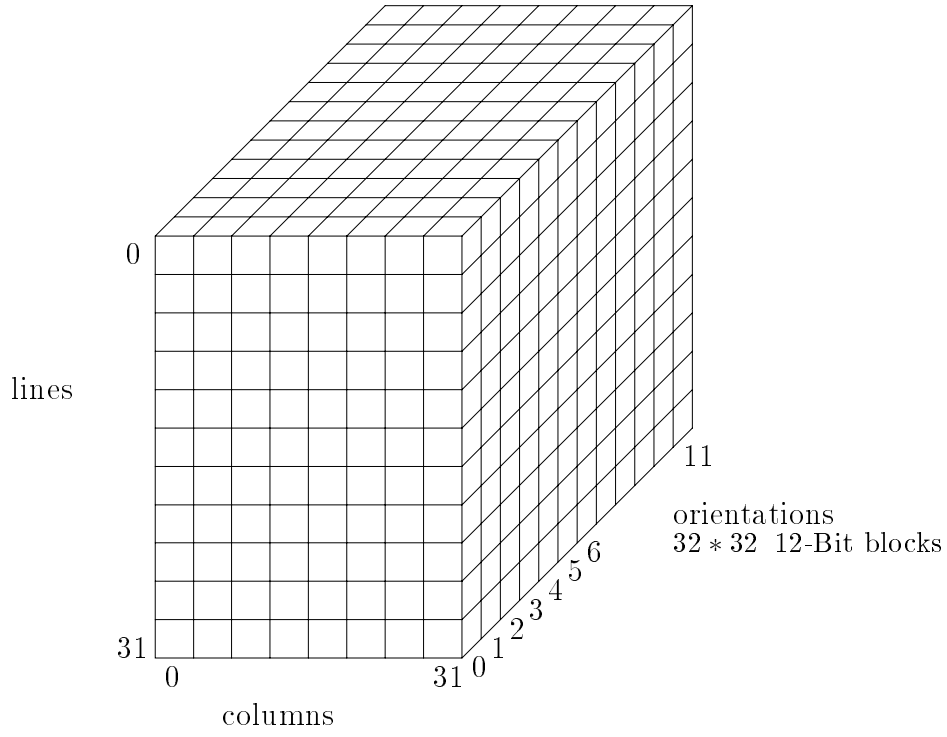


Figure 13: *The 3 dimensional datastructure of our pattern set. The coordinates are given as $x(i)$ (column), $y(i)$ (line) and $z(i)$ cyclic orientation. The numbering of the neuron sites is as follows:* $(0,0,0)$ *is numbered as* 0*, furthermore we abreviate* $(0,0,0) = 0$*,* $(0,0,1) = 1$*, ...,* $(0,0,11) = 11$*,* $(1,0,0) = 12$*, ...,* $(1,0,11) = 23$*,...,* $(0,1,0) = 384$ *and so on.*

References

[1] Amit, D.J., Gutfreund, H., Sompolinsky, H., Phys. Rev. Lett. 55, 1530 (1985), Phys. Rev. A32,1007, (1985), Ann. Phys., NY 173, 30 (1987)

[2] E. Domany, J.L. van Hemmen, K. Schulten (eds.), Physics of Neural Networks, Springer Berlin (1991)
Hertz, J., Krogh, A., Palmer, R.G., Introduction to the Theory of Neural Computation, Addison–Wesley Redwood City (1991)

[3] Canning, A. , Gardner, E., J. Phys. A:Math. Gen. 21, 3275 (1988)

[4] Bouten, M., Engel, A., Komoda, A., Serneels, R., J. Phys. A:Math. Gen. 23, 4643 (1990)

[5] Forrest, B.M., J. Phys. A:Math. Gen. 21, 245 (1988),
Kepler, T.B., Abbott, L.F., J. Physique 49, 1657 (1988)

[6] Gardner, E., J. Phys. A:Math. Gen. 21, 257 (1988), J. Phys. A:Math. Gen. 22, 1969 (1989)

[7] Kinzel, W., Opper, M., Dynamics of Learning, in Physics of Neural Networks, ed. E. Domany, J.L. van Hemmen, K. Schulten, 149, Springer Berlin (1991)

[8] Horner, H., Z. Phys. B-Condensed Matter 75, 133 (1989)

[9] Buhmann, J., Divko, R., Schulten, K., Phys. Rev. A39, 2689 (1989)

[10] Feigel'mann, M.V., Ioffe, L.B., in Physics of Neural Networks, ed. E. Domany, J.L. van Hemmen, K. Schulten, Springer Berlin, 173 (1991)

[11] Müller, K.-R., Proc. of ISCIS VI conf., ed. M. Barray, B. Özgüç, Elsevier Pub., 845 and int. rep 6/91 University Karlsruhe (1991)

[12] Müller, K.-R., in Proc. of the int. conf. on Artificial Neural Networks 2, ed. I.Aleksander, R.Taylor, Elsevier Pub., 95 (1992)

[13] Janßen, H., Kopecz, J., in close-range Photogrammetry meets Machine Vision, 1050, ISPRS, SPIE (1990)

[14] Janßen, H., Kopecz, J., in Proc. of the int. conf. on Artificial Neural Networks, ed. T. Kohonen, K. Mäkisara, O. Simula, J. Kangas, Elsevier Pub., 1203 (1991)

[15] Giefing, G.-J., Janßen, H., Mallot, H., in Proc. of the int. conf. on Artificial Neural Networks, ed. T. Kohonen, K. Mäkisara, O. Simula, J. Kangas, Elsevier Pub., 63 (1991)

[16] Müller, K.-R., Waldenspuhl, A., in Proc. of the int. conf. on Artificial Neural Networks 2, ed. I.Aleksander, R.Taylor, Elsevier Pub., 961 (1992)

[17] Mallot, H.A., von Seelen, W., in Parallel Processing in Neural Systems and Computers, ed. R. Eckmiller, G. Hartmann, G. Hauske, 129-132 North-Holland (1990)