J. Phys. France 51 (1990) 167-183 15 JANVIER 1990, PAGE 167

Classification
Physics Abstracts
75.10H — 05.90 — 87.10

Fast recognition of real objects by an optimized hetero-associative neural network

H. J. Schmitz, G. Pöppel, F. Wünsch and U. Krey

University of Regensburg, Faculty of Physics, D-8400 Regensburg, F.R.G.

(*Reçu le 3 août 1989, accepté le 25 septembre 1989*)

Résumé. — Un concept très bien adapté à la reconnaissance rapide de structures fortement corrélées est développé et réalisé. Nous utilisons comme mémoire hétéro-associative un code de sortie optimisé de façon minimale. Nous construisons une arborescence dont l'indiçage est déterminé par recuit simulé. De cette façon, l'algorithme de stabilisation des structures mémorisées fonctionne de façon optimale. La reconnaissance d'objets « réels », tels des lettres, est étudiée soigneusement. Dans ce cas, les bruits caractéristiques sont fortement anisotropes. Une légère modification de la stratégie de recouvrement minimal de Krauth et Mézard, par entraînement à ce bruit spécifique, permet d'améliorer les performances de notre réseau. Afin d'étudier le réseau et son comportement, nous utilisons une mesure baptisée « constructivité » qui met clairement en évidence les effets d'anisotropie. Un réseau est entraîné à reconnaître un texte et à produire le fichier correspondant. Grâce à son architecture, de nombreux processus peuvent être traités en parallèle et des « transputers » sont uilisés pour sa réalisation.

Abstract. — We have developed and realized a concept which is very well suited for a quick recognition of highly correlated patterns. For a hetero-associative memory we used a minimal optimized output code (index memory). We constructed a tree structure in which the assignment of indices has been optimized by simulated annealing. Thus the algorithm for optimal stability of the learned patterns works most effectively. Special care was taken of recognizing « real » objects, e.g. scanned letters. Here the characteristic noise is very anisotropic. We have slightly modified the minimal overlap strategy of Krauth and Mezard [1] by training with this specific noise, and could improve the performance of our network. In order to get insight into the network and its behaviour we used a measure called *constructivity* which shows clearly the anisotropic effects. We trained a network to recognize a scanned text and to produce the associated text file. Due to the architecture of the network many processes can be treated in parallel. Therefore we used transputers for the implementation.

1. Introduction.

In the last few years there was a lot of progress concerning neural networks [2] even in fields which have been believed to be relatively completed, e.g. the perceptron [3]. To be mentioned is the enhanced performance by improved learning rules for hetero- and auto-associative memories [4-7]. Especially the minimal overlap algorithm [MO] of Krauth and Mezard [1] achieves optimal *stability*, a measure for the performance of a network. Besides linear separable problems there was also a lot of progress in finding solutions for hard computational problems. Some examples are the introduction of higher order correlations [8-10] and new concepts to handle continuously valued neurons [11] and hidden units [12]. All these developments led to a greater attractivity for applications.

Nevertheless practical limitations arise very often due to the huge computational power needed for the simulation of real systems. If one wants to process e.g. images with high resolution in an associative memory, the number of couplings and by this the number of operations grows very quickly. In this paper we want to show how this problem can be treated by an index memory which has a minimal and optimized output code. Our first version only uses $N \cdot \log_2 p$ couplings (p is the number of stored patterns and N the number of neurons). So we need only $N \cdot \log_2 p$ multiplications and additions for the recognition. Therefore both the memory requirement is low and the network is very quick ; additionally the computations can be separated into parallel processes.

If minimal storage requirement is not so crucial, one can essentially increase the optimal stability in our second version by introducing an index tree, i.e. hierarchical arrangement. This memory has $(p-1)N$. couplings but nevertheless there are only $N \cdot \log_2 p$ operations (mult + add) necessary for the decision which pattern belongs to a given input object. The value of the optimal stability depends on the chosen output code ; therefore we optimized this code by *simulated annealing* with a particular cost function to achieve « best » optimal stability. For our index tree this means a skillfull rearrangement of the indices. We used a realistic test for the performance of our network : we scanned a printed text ; different type styles were tested. After classical preprocessing (letters with 1024 pixels, $N = 1\,024$ and 64 patterns, $p = 64$) the text was transposed by the hetero-associative network into a text file.

Our input patterns are highly correlated, the percentage of correlation depending on the type style. The patterns were disturbed by anisotropic noise arising from paper and printing quality and from the resolution of the scanner. We extended the MO algorithm by training with noise [5, 7] for this specific case. With this procedure we got an essential improvement of the retrieval quality. Now we could even retrieve letters, which got by scanning errors higher correlations to other patterns than to the actual letter. For strongly correlated and structured patterns the *stability* gives only a hint for the basins of attraction. We therefore introduced the *constructivity*, a measure which gives more detailed information about the net. A similar measure can be found for other types of nets [13, 14].

We want to emphasize that all kinds of processes were carried out in parallel with two coupled T800-transputers. This included segmentation of the pictures, simulated annealing in the index tree, and the learning process. Especially the assignment of indices to a given set of patterns and the learning process are highly parallelisible. For storing p patterns one might use $p-1$ transputers, i.e. 63 transputers in our case.

The retrieval errors depend on the type style and the quality of the scanned script and vary between one promille and some percent in the worst case. Our results indicate that the index memory, and especially its enhanced version, the optimized index tree memory, are in fact very well suited for real problems in practical applications.

2. Image preprocessing for letters.

Before starting the learning process and the text recognition the patterns have to be extracted from the graphic background and must be normalized. We used a 300 dpi scanner to produce a binary black/white pixel file. There was also some picture preprocessing with classical methods : we applied picture segmentation, scaling and shift in position of the extracted objects, but we did not use methods like edge detection or any kind of filtering. We only give here some ideas what we applied. Nevertheless these steps have to be done very carefully in order to avoid loosing necessary information.

SEGMENTATION OF SCANNED PICTURES. — The information contained in the picture was transformed into the so called Run-Length-Code [15]. From this we extracted by a labelling algorithm objects consisting of continuous black regions. In fact not all letters consist of only one such region, e.g. « i » or the german « ä ». Using a special algorithm to detect such structures we regarded them as *one* object. Furthermore we took care of separating kerned objects e.g. Tē, where the two letters have a vertical overlap.

SHIFT AND SCALE INVARIANCES. — The extracted objects were now adjusted according to their « center of mass ». We looked for the relation between height and width of all letters and assigned the neurons to an area having about the same proportion. We computed a common zoom factor for all letters in order to maximize the number of neurons which carry information. Because of the invariance against shifts the information about the vertical position is lost ; we handled this problem by introducing additional neurons for this position, which is useful to distinguish « p » and « P ».

3. The heteroassociative index memory.

3.1 OPTIMAL RECOGNITION TIME WITH AN INDEX MEMORY. — Handling a completely connected neural network needs a lot of number crunching. For practical applications a resolution of $N \geqslant 1\,000$ is necessary which can only be mastered by a powerful computer. We want to reduce the time for learning and recognition without loosing a high retrieval quality. Therefore we choose an associative index or address memory.

We assign to each pattern an index consisting of a set of binary values -1 and $+1$. For $p = 2^x$ patterns we need therefore at least $x = \log_2 p$ index bits. This results in an asymmetric coupling matrix J_{ik} with rows $i = 1, \ldots, \log_2 p$ and columns $k = 1, \ldots, N$ (see Fig. 1). The number of couplings is given by $N \cdot \log_2 p$.

The assignment of a certain input pattern to an index is realized by the following one step dynamics :

$$u_i = \operatorname{sgn}\left(\sum_k J_{ik} S_k\right) \tag{1}$$

with input vector $\mathbf{S}$ ($S_k = \pm 1$), index vector $\mathbf{u}$ ($u_i = \pm 1$) and real couplings J_{ik}.

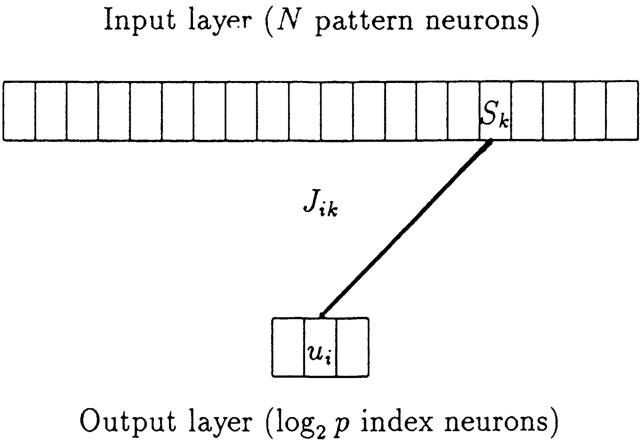
Input layer (N pattern neurons)
S_k
J_{ik}
u_i
Output layer ($\log_2 p$ index neurons)

Fig. 1. — The two index memory layers.

One sees that for this one-step relaxation only $\log_2 p \, . \, N$ multiplications and additions are necessary. This means an enormous reduction of computational effort in comparison to a completely connected net. The procedure is also very economic concerning memory requirement. For realistic dimensions ($p = 64$, $N = 1\,024$) the coupling matrix needs only 24 KBytes.

In table II we give an example for the assignment of indices to the eight letters « A » till « H ». Here they are simply alphabetically ordered, they may represent memory addresses, where their ASCII-Code is stored. Whether one can find a more efficient ordering, will be discussed below (see Tab. III).

3.2 LEARNING ALGORITHM AND CONSTRUCTIVITY. — We use the minimal overlap algorithm of Krauth and Mezard [1] to adapt the net to a given set of patterns. At first all couplings are cleared. In step 2 we present all patterns $\boldsymbol{\xi}^{\mu}$ to the network and measure the stability Δ_i^{μ} for each index bit i :

$$\Delta_i^{\mu} = u_i^{\mu} \cdot \frac{\sum_k J_{ik} \cdot \xi_k^{\mu}}{|\mathbf{J}_i|} \tag{2}$$

with the row vector $\mathbf{J}_i$ of the coupling matrix. In step 3 we learn for each index bit only one pattern, namely one of those wich have the smallest stability (« minimal overlap ») :

$$\Delta J_{ik} = \frac{1}{N} u_i^{\mu} \, . \, \xi_k^{\mu} \, . \tag{3}$$

Steps 2 and 3 are repeated until $\min_{\mu} \{\Delta_i^{\mu}\}$ reaches a maximum (« optimal stability »). At the end the row vector $\mathbf{J}_i$ is given by $\mathbf{J}_i = (1/N) \sum_{\mu} g_i^{\mu} \, u_i^{\mu} \, \boldsymbol{\xi}^{\mu}$. The parameters g_i^{μ} indicate, how often bit i of pattern μ has been learned.

For each index bit i only the row vector $\mathbf{J}_i$ determines which pattern has to be learned at a certain learning step. The learning procedure for a bit i also involves no other index bits $j \neq i$. Therefore each row $\mathbf{J}_i$ of the coupling matrix can be learned separately : the learning algorithm can be carried out in parallel using $\log_2 p$ processors (e.g. transputers).

Text recognition is a hard task for this memory model. The patterns have a mean correlation between 0.6 and 0.7, i.e. 80 % till 85 % of the pixels are equal on an average. Between some special pairs of patterns (e.g. « e » and « c ») there exists a correlation of more than 0.9, about 95 % of the pixels are equal. Furthermore we demand that the retrieval is correct even for letters which, by scanning errors, got noisy at the transitions between black and white regions or are displaced as a whole by one pixel (discretization error). For this aim very anisotropic basins of attraction are necessary.

To characterize such anisotropic regions the stability Δ_i^{μ} gives only a hint. The system may have a high stability but is affected strongly by the specific noise. A few large coupling constants may produce a high stability. But if the characteristic noise will influence just the neurons belonging to these couplings, recognition will not work. We therefore define a new measure $K_{ik}^{\mu}(S_k)$ called *constructivity* :

$$K_{ik}^{\mu}(S_k) = \frac{u_i^{\mu} \, . \, J_{ik} \, . \, S_k}{|\mathbf{J}_i|} \, . \tag{4}$$

It describes for a given input vector $\mathbf{S}$ the contribution of one input neuron k to the recognition of the index bit i. A positive value of K_{ik}^{μ} favours a correct assignment, a negative

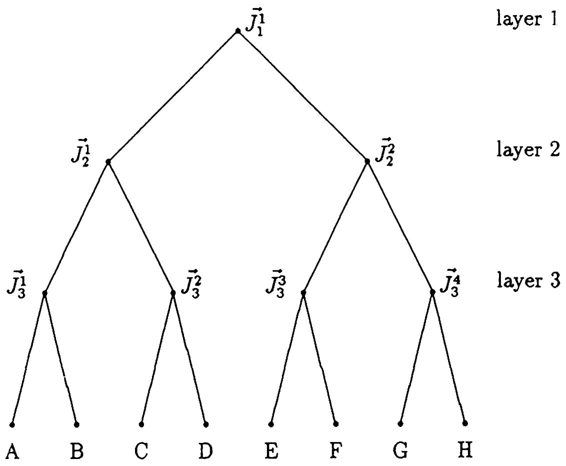

Fig. 2. — The tree structure for 8 patterns.

Table I. — *Retrieval test with* 1 022 *patterns.*

Number of the test	1	2	3
Learned with noise by shifting	No	Yes	Yes
Learned with noise at edges	0 %	0 %	3 %
Number of MO-learning steps	5 000	30 000	30 000
Retrieval errors	14	9	8
Errors marked as incertain decision	4	8	8
All incertain decisions	24	10	16
$\min_{i\mu} \{\Delta_i^\mu\}$	3.45	2.65	2.49
$\langle \min_i \{\Delta_i^\mu\} \rangle_\mu$	3.84	3.09	3.04
$\langle h(\mathrm{S}) \rangle_\mathrm{S}$	2.75	2.13	2.3

value works against it. The *stability* of a pattern's index bit, Δ_i^μ, is simply given by the sum over constructivities :

$$\Delta_i^\mu = \sum_k K_{ik}^\mu(\xi_k^\mu) .$$

It is useful to know which and how many input neurons of a pattern are allowed to be changed without loosing the correct assignment by the net. Sorting all constructivities

K_{ik}^{μ} by magnitude answers this question. E.g. the sign of the largest constructivities can be changed as long as the sum over all K_{ik}^{μ} stays positive (see also Sect. 5).

In figures 3 and 6 examples for the vector $\mathbf{K}_i^{\mu}(\mathbf{S}) = (K_{i1}^{\mu}(S_1) \dots K_{iN}^{\mu}(S_N))$ for fixed i, μ and $\mathbf{S}$ are plotted. The range of the index $k = 1, \dots, N$ is broken into lines giving a two-dimensional picture of the same kind as the original scanned pattern, however with nine different grey values for the pixels. Dark points indicate constructive neurons, white points destructive ones. If the number of learned patterns is small one recognizes in these grey scale pictures the patterns stored in the J_{ik} (e.g. « i » and « m » in the case of figures 3 and 4 where the input pattern $\mathbf{S}$ was « i »). By this method the decisive neurons with a large positive or negative constructivity can be found and reasons for wrong assignment can be studied. Neurons which are never or seldom affected by noise can also be detected.

Figure 3 and figure 4 are obtained with the same input vector $\mathbf{S}$ (« i »), but with different coupling matrices resulting from training without noise (Fig. 3) and with noise (Fig. 4). Details on the training procedure will be discussed later. At present we only concentrate on the differences of these two figures : namely in figure 3, the constructivities are either zero or

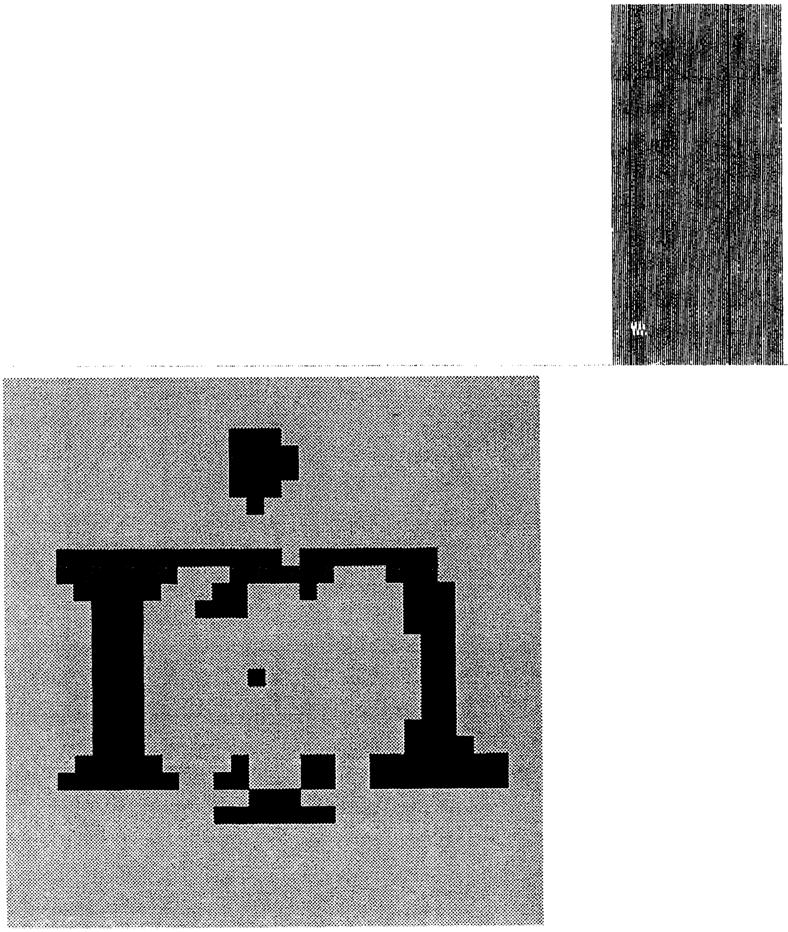

Fig. 3. — Gray scale picture (lower part) and sorted constructivities (upper part) for a matrix row within the lowest layer ; learning without noise ; stored patterns : « i », « m » ; tested pattern : « i ».

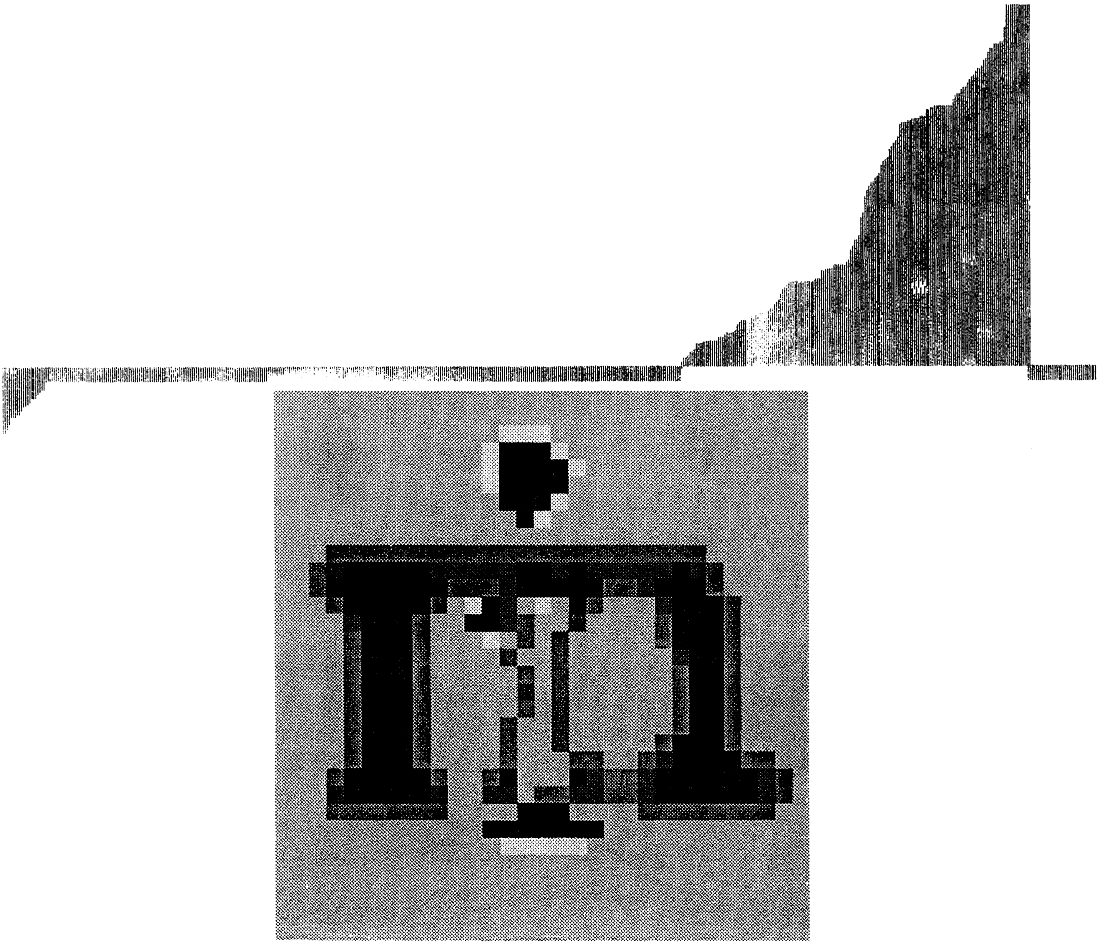

Fig. 4. — Gray scale picture (lower part) and sorted constructivities (upper part) for a matrix row within the lowest layer ; learning with noise ; stored patterns : « i », « m » ; tested pattern : « i ».

take a constant positive value (see the upper part of the figure) whereas in figure 4 the distribution of the constructivities is more smooth, particularly at the edge, which implies that scanner errors are no longer so-critical.

From these and other figures one finds that for *small number of patterns* many constructivities are zero. In that case there is still place for more information within the coupling matrix, and the basins of attraction can be modelled without destroying the stability of the patterns themselves. A good modelling can be realized by learning also displaced patterns and patterns with noisy edges (see in Sect. 4.4). In connection with the index tree (see Sect. 4), these considerations are of great importance especially for the lower part of the tree, where the condition of a small number of patterns is fulfilled.

4. The index tree.

When we use exactly $x = \log_2 p$ index bits for $p = 2^x$ patterns we see from table II that $\sum_{\mu} u_i^{\mu} = 0$. For every index bit i there exist two classes of patterns with the same number of

Table II. — *An example for an assignment of indices to* 8 *patterns.*

pattern	u_1	u_2	u_3
A	-1	-1	-1
B	-1	-1	$+1$
C	-1	$+1$	-1
D	-1	$+1$	$+1$
E	$+1$	-1	-1
F	$+1$	-1	$+1$
G	$+1$	$+1$	-1
H	$+1$	$+1$	$+1$

elements. The members of the first class have a positive u_i^μ, the members of the second one have a negative u_i^μ. One now of the coupling matrix produces one index bit and thus makes an assignment to one class. The complete index vector of a pattern results from the common part of these $\log_2 p$ classes.

A hierarchically structured index tree (see Fig. 2) is a better way to recognize a pattern as correctly as possible. The first step is the same as above. We decide in this layer 1, whether the most significant index bit ($i = 1$) is $+1$ or -1. By this we find out to which of two main classes the pattern belongs. For this aim only the 1st row $\mathbf{J}_1$ of the coupling matrix is needed. Depending on the result we use two branches for the next step, i.e. for the determination of the index bit 2 (layer 2). Both branches use the 2nd row $\mathbf{J}_2$ of the coupling matrix. However, the two versions of $\mathbf{J}_2$ are generally different ; we therefore introduce a numbering $\mathbf{J}_i^\ell$ with i as the number of the layer and ℓ as the special version of the matrix row within one layer. We continue this procedure until — within the last layer — there rests only a decision between two possible patterns.

The number of decision ($\log_2 p$) is the same as in the unstructured index model, however as we will see, with the hierarchical tree higher stabilities will be obtained. The reason is that more coupling constants are available for optimization, namely e.g. $\mathbf{J}_i^1$ and $\mathbf{J}_i^2$ instead of only $\mathbf{J}_i$. A detailed proof will be given below.

The system considered in the following consists of $p - 1$ one-row-matrices, but for the retrieval of a special pattern only $\log_2 p$ of them are used, e.g. for the letter « A » in figure 2 only $\mathbf{J}_1^1$, $\mathbf{J}_2^1$ and $\mathbf{J}_3^1$.

4.1 SAME RECOGNITION TIME, BUT HIGHER STABILITIES. — In this section we want to show that the minimal stability $\Delta^\mu = \min_i \{\Delta_i^\mu\}$ of a pattern generally becomes greater if we use the tree model. Of course we assume a fixed assignment of a set of patterns to a set of index vectors. The reason for this improvement is rather obvious. Only the top layer has to decide between all p patterns with rather small basins of attraction. Whereas the matrices of layer i ($i > 1$) have to store a reduced number of patterns, namely $p \cdot 2^{1-i}$. As already mentioned in section 3.2, a smaller number of patterns can improve the performance of the network. Generally, the stabilities become larger when we go down the index tree.

We want to illustrate these remarks by the example of table II. In the simple index model $\mathbf{J}_2$ has to distinguish between the set of patterns $\{A, B, E, F\}$ ($u_2 = -1$) and $\{C, D, G, H\}$ ($u_2 = +1$). Using the index tree $\mathbf{J}_2^1$ has only to decide between $\{A, B\}$ and $\{C, D\}$, $\mathbf{J}_2^2$ between $\{E, F\}$ and $\{G, H\}$, whereas in the simple index model $\mathbf{J}_i$ must recognize all patterns which are known to the $\mathbf{J}_i^\ell$ for all ℓ. Of course, when for each layer i all matrices J_i^ℓ are equal, the tree model and the simple index net are equivalent, however, if they are different stability may be gained. In fact, for a fixed output code, each stability of a pattern within the index tree is always greater than or equal to the corresponding stability of the simple index net, since a given matrix J_i^ℓ has to store a smaller number of patterns in the index tree, as already mentioned.

To prove this, we show that the minimal stability cannot become greater when new patterns are additionally stored into one 1-row-matrix $\mathbf{J}_i^\ell$. For these considerations we drop the subscripts i and ℓ and only write for the matrix $\mathbf{J}$ and for the index of a pattern u^μ. Vectors $\boldsymbol{\eta}^\mu$ are defined for all index bits by

$$\boldsymbol{\eta}^\mu = u^\mu \cdot \boldsymbol{\xi}^\mu . \tag{5}$$

These vectors are parallel to the patterns vectors $\boldsymbol{\xi}^\mu$, but with reversed direction for $u^\mu = -1$.

We remind to the geometrical point of view introduced by Krauth and Mezard [1]. Their learning algorithm adjusts $\mathbf{J}$ in such a way that $\mathbf{J}$ becomes the symmetry axis of the smallest cone enclosing all vectors $\boldsymbol{\eta}^\mu$. We can understand this fact by the following consideration. At each learning step the pattern $\boldsymbol{\xi}^\mu$ with the smallest stability is learned. According to the definition of the stability we have

$$\Delta^\mu = \frac{u^\mu}{|\mathbf{J}|} \cdot \sum_k J_k \, \xi_k^\mu = \frac{\boldsymbol{\eta}^\mu \cdot \mathbf{J}}{|\mathbf{J}|} .$$

The smallest stability is given by $\Delta = \min_\mu \{\boldsymbol{\eta}^\mu \cdot \mathbf{J}/|\mathbf{J}|\}$ which corresponds to the largest angle between the vectors $\boldsymbol{\eta}^\mu$ and $\mathbf{J}$. Therefore all $\boldsymbol{\eta}^\mu$ lie within a cone with symmetry axis $\mathbf{J}$. The angle of the cone is determined by the minimal stability reached at a given moment. Each learning step turns $\mathbf{J}$ a little bit towards the vector $\boldsymbol{\eta}^\mu$ to be learned at the moment. When everything is stable $\mathbf{J}$ is the symmetry axis of the cone mentioned. The assignment of a (noisy) input vector $\tilde{\boldsymbol{\xi}}$ will be correct if the angle between its $\tilde{\boldsymbol{\eta}}$ and $\mathbf{J}$ is smaller than $\pi/2$.

From these considerations we clearly see that the smallest cone including p patterns cannot get smaller if a new pattern $\boldsymbol{\eta}^{p+1}$ is added. The cone either remains the same, when $\boldsymbol{\eta}^{p+1}$ lies already within it, or will have a larger angle. Therefore adding a new pattern cannot increase minimal stability. This is valid for an arbitrary but fixed arrangement of indices, which completes our proof.

The index tree increases the requirement of memory. For 1024 pixels and 64 letters 252 Kbytes are necessary instead of 24 Kbytes. However as mentioned above, the number of computing steps for recognition is not increased. Only learning becomes a bit more complex ; $p-1$ instead of $\log_2 p$ matrix rows have to be learned, but the mean number of patterns per matrix row decreases. For learning the whole tree, using L learning steps according to Krauth and Mezard we need $L \cdot N \cdot (p \cdot \log_2 p + p - 1)$ multiplications and additions instead of $L \cdot N \cdot (p+1) \cdot \log_2 p$. For $p = 64$ that is only a factor of 1.15, and for $p \to \infty$ agrees asymptotically. Furthermore numerical tests show that the number of learning steps can be reduced in lower branches of the tree.

Each of the $p-1$ nodes within the tree can be computed in parallel. So in any case learning may be accelerated very much by using many processors.

4.2 OPTIMAL INDICES FOR HIGHER STABILITY.

— For highly correlated patterns the assignment of indices to the patterns is of great importance. By skillful arranging of the indices stabilities can be drastically increased and errors reduced. Similar patterns should get similar indices, for example. For the top matrix of the tree it then will be simple to assign these patterns to the same class. Very difficult decisions between high correlated patterns have to be made in the last layers, where the matrices only store a few patterns.

Mean stabilities for three different arrangements of the same 64 patterns are shown in table III. The mean correlation of the pattern was $\langle \boldsymbol{\xi}^\mu \cdot \boldsymbol{\xi}^\nu \rangle \Big|_{\substack{\mu,\nu \\ \mu \neq \nu}} = 0.65$.

Table III. — *Retrieval of a scanned text with* 1 022 *letters and after learning with* 20 000 *Krauth/Mezard steps without noise ; an index tree was used with three different assignments of patterns to indices.*

	$\langle \min_\mu \{\Delta^\mu\} \rangle_\ell$ in layer						retrieval errors
	1	2	3	4	5	6	
arrangement 1 well chosen	4.1	5.6	6.5	7.9	9.8	12.1	0.2 %
arrangement 2 alphabetically	2.2	3.0	4.6	7.2	10.4	15.1	1.7 %
arrangement 3 badly chosen	1.1	2.1	3.6	6.5	10.8	17.4	14 %

By the different arrangements there result different vectors $\boldsymbol{\eta}^\mu = u^\mu \cdot \boldsymbol{\xi}^\mu$. The minimal angles of Krauth/Mezard's cones are the different, and the stabilities reached are not the same. Therefore one should find an optimal arrangement where the cone formed by the $\boldsymbol{\eta}^\mu$ has a minimal angle. Then the minimal stability can reach a maximum. There may exist more such arrangements. These « best » arrangements cause a cone with the same angle and the same minimal stability. However the mean stabilities for these arrangements can vary because the $\boldsymbol{\eta}^\mu$ may have different directions within the cone.

4.3 OPTIMIZED INDICES BY SIMULATED ANNEALING. — We looked for optimal indices by maximisation of the mean stabilities. For a first trial we used couplings which were constructed according to the Hebb rule : $J_k = (1/N) \sum_\mu u^\mu \cdot \xi_k^\mu$ (the index i is here dropped again). The mean stability is then given by

$$\frac{1}{p} \sum_\mu \Delta^\mu = \frac{1}{pN \cdot |\mathbf{J}|} \sum_{\mu,\nu} u^\mu u^\nu \sum_k \xi_k^\mu \xi_k^\nu = + \frac{1}{p} \sqrt{\sum_{\mu\nu} \boldsymbol{\eta}^\mu \cdot \boldsymbol{\eta}^\nu} .$$

We performed our simulated annealing along the lines of standard technics [16] with the following cost function :

$$H = - \frac{1}{pN} \left(\sum_\mu \Delta^\mu \right)^2 . \qquad (6)$$

Taking account of to the constraint that half of the indices must be + 1, we exchange always two index bits with opposite sign in course of the annealing procedure. If the indices of pattern $\boldsymbol{\xi}^\alpha$ and $\boldsymbol{\xi}^\beta$ would be exchanged, then we get

$$\Delta H = \frac{1}{pN} \sum_{\nu \neq \alpha, \beta} (\boldsymbol{\eta}^\alpha \cdot \boldsymbol{\eta}^\nu + \boldsymbol{\eta}^\beta \cdot \boldsymbol{\eta}^\nu) . \tag{7}$$

The best of the above mentioned configurations (see Tab. III) has been found by this method. Interestingly the procedure is equivalent to the search of index configurations where the patterns are grouped into classes so that the mean correlations between these classes are very small. In this case the cost function would be :

$$C = \frac{1}{pN} \cdot \sum_{\mu \in A, \nu \in B} \boldsymbol{\xi}^\mu \cdot \boldsymbol{\xi}^\nu$$

with $A = \{\mu : u^\mu = +1\}$ and $B = \{\nu : u^\nu = -1\}$.

If there is an exchange of the indices e.g. u^α and u^β $(u^\alpha = -u^\beta)$, one gets

$$\begin{aligned}\Delta C &= \frac{1}{pN} \left[- \sum_{\substack{\nu \neq \beta \\ u^\nu = -u^\alpha}} \boldsymbol{\xi}^\alpha \cdot \boldsymbol{\xi}^\nu - \sum_{\substack{\nu \neq \alpha \\ u^\nu = -u^\beta}} \boldsymbol{\xi}^\beta \cdot \boldsymbol{\xi}^\nu + \sum_{\substack{\nu \neq \alpha \\ u^\nu = u^\alpha}} \boldsymbol{\xi}^\alpha \cdot \boldsymbol{\xi}^\nu + \sum_{\substack{\nu \neq \beta \\ u^\nu = u^\beta}} \boldsymbol{\xi}^\beta \cdot \boldsymbol{\xi}^\nu \right] \\ &= \frac{1}{pN} \sum_{\nu \neq \alpha, \beta} u^\nu u^\alpha (\boldsymbol{\xi}^\alpha \cdot \boldsymbol{\xi}^\nu - \boldsymbol{\xi}^\beta \cdot \boldsymbol{\xi}^\nu) \\ &= \frac{1}{pN} \sum_{\nu \neq \alpha, \beta} (\boldsymbol{\eta}^\alpha \cdot \boldsymbol{\eta}^\nu + \boldsymbol{\eta}^\beta \cdot \boldsymbol{\eta}^\nu)\end{aligned}$$

which means $\Delta C = \Delta H$, i.e. the above mentioned equivalence.

The rearrangement of the indices is done first in the top layer of the index tree and then step by step in the lower layers. Because the calculations for nodes of the same layer are independent they could be done in parallel. For every node of the layer i there are $\binom{p \cdot 2^{1-i}}{p \cdot 2^{-i}}$ configurations of indices possible. Note that the computing time for the rearrangement of the indices needs only a fraction of the computing time for the learning procedure. Simulated annealing with the above mentioned cost function (6) leads to good stabilities with the MO-algorithm. From geometrical considerations of the problem we know that there must exist a cost function which takes *exactly* respect of the learning with the minimal overlap algorithm (this is presently under study). In any case we want to stress the fact that the rearrangement of the indices leads also to a better performance for the normal index memory, but especially for the index tree this procedure shows its great effectivity. Only for the index tree memory it makes sense to set priority to the first index neuron where corresponding couplings have to do the most difficult decisions, whereas in the normal index memory all rows of the coupling matrix have to store the same number of patterns.

Applying the cost function (6), the patterns within one class of the tree exhibit higher correlations than the patterns of different classes. Lower in the tree there are more difficult decisions, but this effect is compensated by the lower number of patterns which have to be stored in the corresponding matrices. We observed that within the top node the algorithm of Krauth and Mezard only asked for a subset of all patterns. At the end all patterns were recognized by the first layer, even those which were not learned explicitly [17]. Obviously the algorithm looks for « typical » patterns to learn ; other patterns with a high overlap to these ones are learned implicitly. For highly correlated patterns it should be possible to enlarge the tree without problems and to increase the number of patterns to be learned using the same number of neurons.

4.4 TRAINING WITH NOISE. — During the training phase we used a learning procedure consisting of a series of two alternating steps :

At first, one learning step with an MO algorithm was performed after presenting the original patterns with noise at the edges, i.e. $\boldsymbol{\xi}^\mu \rightarrow \tilde{\boldsymbol{\xi}}^\mu$, and $\Delta J_k = \frac{1}{N} u^\mu \cdot \tilde{\xi}_k^\mu$, if $\tilde{\boldsymbol{\xi}}_k^\mu$ is the pattern with the minimal overlap. The noisy patterns were produced by flipping a fixed percentage of the neurons at the edges.

Then, at the second step, for each pattern one randomly selects one of nine shift vectors $\boldsymbol{\tau}^\mu = (n_1, n_2)$ with $n_i \in \{-1, 0, +1\}$ for each pattern, shifts the corresponding 2D-image as a whole by $\boldsymbol{\tau}$ and applies the MO procedure for this set of shifted original patterns.

This noise used during learning procedure reflects the typical noise, appearing during recognition.

Both training-steps are done alternatingly, so that the pure patterns remained favorised. This method leads to the remarkable result that even such test patterns are correctly associated by the network, for which the classical method of comparing hamming distances failed. For the future we want to introduce weighted noise at different layers of the network, because strong noise destroys the memory of the upper node but leads probably to a better formation of the basins of attraction at lower levels.

5. Results.

We tested our index tree with a scanned text and good quality copies with different type styles. In front of the text we always printed the 64 letters to be learned. We used these training patterns for simulated annealing and the learning process itself. All in all the text contained 1 022 letters. Characters, which were already known to be critical were often repeated. The copies were of good quality. For further considerations we have chosen as typical the text printed with the type style « roman ». We observed little better or little worse results with other type styles.

The following examples were learned with the same assignment of indices and with the same patterns. For the top node we used L learning steps according to Krauth and Mezard, for lower layers i only $L \,.\, (2/3)^{i-1}$ steps were applied. We occupied the nodes of the lowest layer, which have to store only two patterns, according to Hebb's learning rule.

We wanted that the system itself judges if the assignment of an input pattern **S** to an index is certain or uncertain. For this aim we use a quality measure h defined by

$$h(\mathbf{S}) = \min_i \left\{ \frac{|\mathbf{J}_i \cdot \mathbf{S}|}{|\mathbf{J}_i|} \right\} .$$

For the special case that the input vector **S** is a learned pattern $\boldsymbol{\xi}^\mu$, is the minimum of the stabilities Δ_i^μ of the index bits. Assignments with $h(\mathbf{S}) < 1$ were marked as uncertain.

5.1 RETRIEVAL QUALITY OF THE INDEX TREE.

Learning without noise. — After 5 000 steps the stabilities practically did not change any more and learning was stopped. 98.6 % of the text was recognized correctly (see Tab. I) only highly correlated pairs of letters were mixed up, e.g. u/n, l/I, !/l, i/l. Due to the ordering used these errors occurred in the nodes of layer 3 till 6. We found out, that either the pattern itself or the training patterns were shifted or were very noisy.

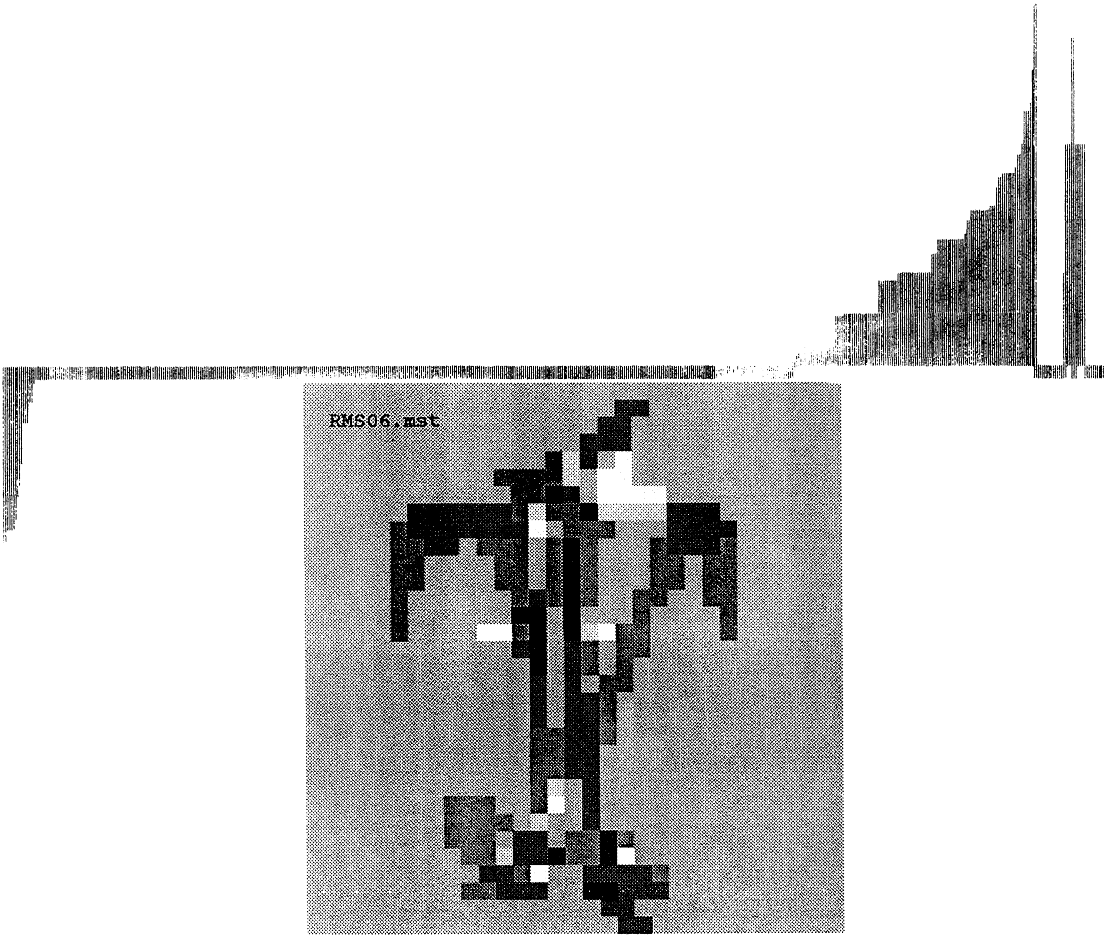

Fig. 5. — Gray scale picture (lower part) and sorted constructivities (upper part) for a matrix row of layer 3 ; learning without noise ; stored patterns : (TY(y!1If) ; tested pattern : « l ».

A problem arises from the choice of the training patterns. If they are, by chance, bad then recognition cannot work well because these objects form the only basis for learning. Tests with the mean value of 5 realisations of one pattern did not lead to satisfactory results. We suppose that we have to use many more scanned versions of one letter to form an ideal training pattern.

Nevertheless even by learning without noise we got a retrieval quality which is as good or even better as the classical method of comparing correlations (1.4 % errors comparing to 1.7 % of the classical method, see figure 8 for an example, for an error of the classical method and correct recognition with the index tree). However, our recognition is much quicker. The ratio of computations needed for the recognition scales like $p/\log_2 p$. For 64 patterns our method is one order of magnitude faster.

Learning with noise. — The two kinds of noise (shifting and noisy edges, see Sect. 4.4) were applied during the learning process in order to still improve the retrieval quality. For this aim there are more learning steps necessary, e.g. 30'000 instead of 5 000. But we see from table 1 that the number of errors was reduced almost by a factor of two. The number of uncertain

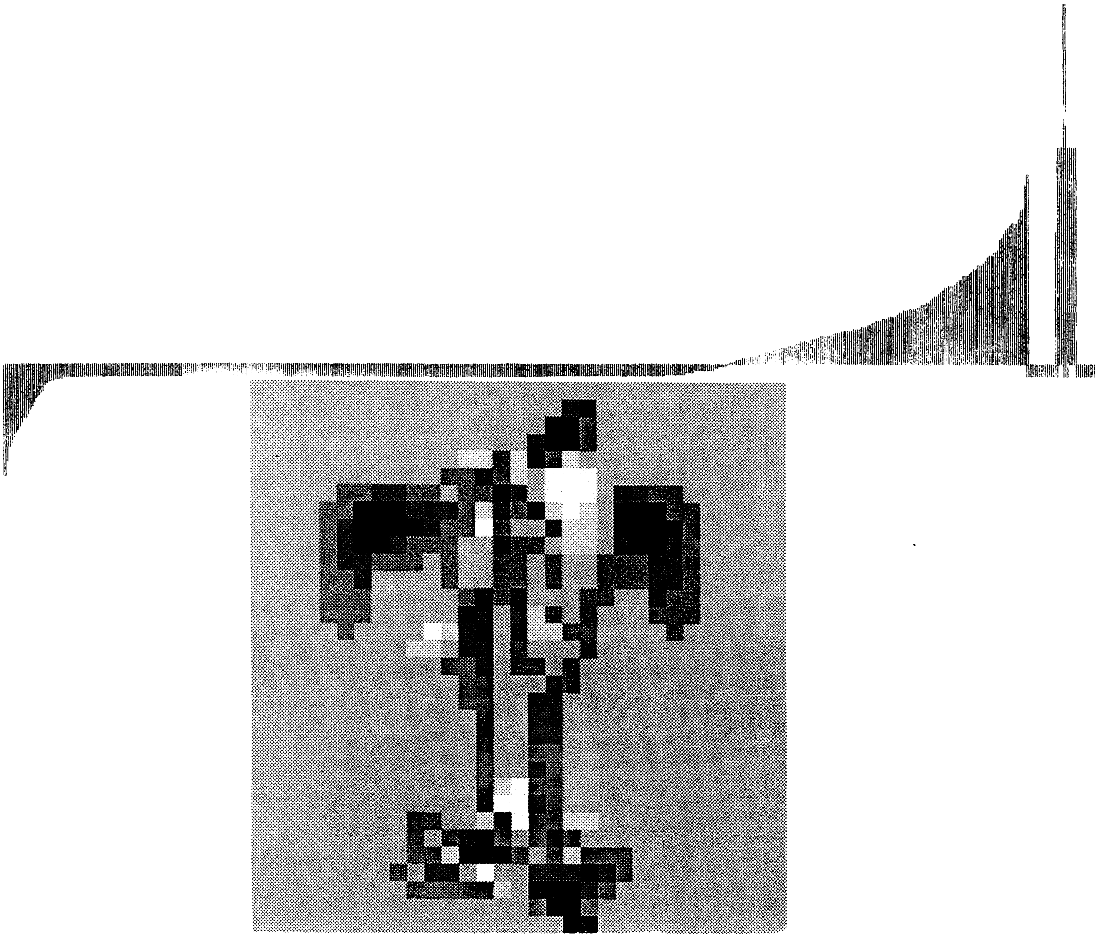

Fig. 6. — Gray scale picture (lower part) and sorted constructivities (upper part) for a matrix row of layer 3 ; learning with noise ; stored patterns : {TY(y!1If} ; tested pattern : « l ».

decisions got smaller, too. Moreover, after learning with noise the system « knows » when it has made a wrong assignment. Practically : all errors are marked to be uncertain decisions with $h < 1$. A very hard test for the system is the recognition of inputs which have a higher correlation to other trained patterns than to the correct one. With our method more than the half of such inputs were assigned correctly, but are marked as uncertain decisions (see example in Fig. 7).

The special parameters for the learning with noise depend strongly on the patterns, i.e. on the type style of the letters. The type style « roman » on which we report here consists of letters with many thin lines. Flipping more than 3 % of the pixels on edges will distroy such letters. For this style the significant improvement of retrieval quality was reached by shifting, not by noisy edges (see Tab. 1). For other styles we observed the opposite behavior. A very suitable method to optimize the noise parameters was the evaluation of the *constructivities* plotted as grey scale pictures (see next section).

Table I also contains the mean value of the « quality » parameter h over all scanned letters. For systems with enhanced performance (number 2 and 3 in Tab. I) this mean value

Original text

a b c d e f g h i j k l m n o p q r s t u v w x y z . , ? ( ) !
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ß - : ; 64

A NEUROPHYSIOLOGICAL POSTULATE

Let us assume then that the persistence or repetition of a reverberatory activity (or "trace") tends to induce lasting cellular changes that add to its stability. The assumption can be precisely stated as follows: When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

D. O. Hebb, The Organization of Behavior

Recognition by tree learned with noise and optimized indizes:

a b c d e f g h i j k l m n o p q r s t u v w x y z . , ? ( ) !
A B C D E F G H I J K L M $\underline{N}$ $\underline{O}$ P Q R S T U V W $\underline{X}$ Y $\boxed{Z}$ ß - : ; 64

A NEUROPHYSIOLOGICAL POSTULATE

Let us assume then that the persistence or repetition of a reverberatory activity (or $\boxed{,}\boxed{,}$trace$\boxed{,}$,) tends to induce lasting cellular changes that add to its stability. The assumption can be precisely stated as follows: When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A,s efficiency, as one of the cells firing B, is increased.

D. O. Hebb, The Organization of Behavior

Recognition by tree learned without noise, and with alphabetical indices:

a b c d e f $\underline{g}$ h i j k $\underline{l}$ m n o $\boxed{P}_{p}$ $\underline{q}$ r s t u v w x $\underline{y}$ z . , ? ( ) ! $\boxed{A}$ $\underline{B}$ C D E $\underline{F}$ $\underline{G}$ H $\boxed{I}$ J $\boxed{Z}_{K}$ $\boxed{L}$ M
$\underline{W}_{N}$ $\boxed{Q}_{O}$ P Q $\boxed{R}$ $\boxed{S}$ $\boxed{T}$ U V W $\underline{X}$ $\boxed{Y}$ Z ß - $\underline{:}$; 6$\underline{4}$

$\boxed{A}$ NEU$\underline{R}$$\boxed{D}_{O}PH\underline{YS}$$\boxed{I}$$\underline{OLOG}IC\underline{A}$L P$\underline{O}STU\boxed{L}$$\boxed{A}T\boxed{B}_{E}$

$\underline{L}$et us assume then that the $\boxed{P}_{p}$ersistence or repetition o$\underline{f}$ a reverberatory activity (or $\boxed{.}$.trace$\underline{.}$,) tends to induce $\boxed{l}$asting ce$\boxed{;}_{l}$$\boxed{;}_{l}u\boxed{;}_{l}$$\underline{a}$r changes that add to its stabi$\underline{l}$ity. T$\boxed{n}_{h}$e assum$\boxed{p}$t$\underline{i}$on ca$\underline{n}$ be $\underline{p}$recisel$\underline{y}$ stated as follows: When an $\underline{a}$xon of ce$\underline{l}$l $\underline{4}_{A}$ is near eno$\underline{ug}$h to excite a ce$\underline{l}$l $\boxed{B}$ and re$\boxed{p}$e$\underline{a}$tedl$\underline{y}$ or $\underline{p}$ersisten$\boxed{I}_{l}$$\underline{y}$ ta$\underline{k}$es $\boxed{p}$art in f$\underline{i}$ring it, so$\underline{m}$e gr$\underline{o}$wt$\boxed{k}_{h}$ $\underline{p}$rocess or metabolic chan$\underline{g}$e ta$\underline{k}$es P$_{p}$lace in one or both ce$\boxed{l}$$\boxed{l}$s suc$\underline{h}$ that $\boxed{A}$,$\underline{s}$ efficienc$\boxed{y}$, as one of t$\boxed{k}_{h}$e cells firin$\underline{g}$ $\boxed{B}$, is incre$\boxed{a}$sed.

D. $\underline{O}$. $\boxed{H}$ebb, $\underline{T}$he $\boxed{O}$r$\boxed{g}$an$\underline{i}$zation of $\underline{B}$eh$\boxed{a}$vior

Fig. 7. — This is an example of recognition of a scanned text of Donald O. Hebb. We used several training matrices, and uncertain decisions are marked in steps : letters are underlined if $0.5 < h \leq 1$ and have a frame for $h \leq 0.5$. In case of errors made by the index tree, the correct letter is printed as a lower index. Note that at first all 64 letters are printed, which should be recognized by the system. The quotation marks are not among these patterns, so they are not correctly recognized in the following text, where otherwise the recognition is performed without errors. In contrast the badly trained tree makes a lot of errors and uncertain decisions.

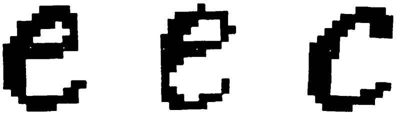

Fig. 8. — Learning an *italic type*, we had a very bad training pattern for the letter « e ». From left to right the pictures show : 1. The test pattern in the text, to be recognized (an « e » as a humain being will see at once), 2. The corresponding training pattern, and 3. The training pattern « c ». A recognition by a simple comparison of hamming distances to the training patterns fails — the testpattern is mostly correlated to the training pattern « c ». But our index tree recognizes this pattern correctly as an « e », marking its decision as uncertain.

decreases. But we observed that the standard deviation of the distribution was reduced, too. By this the number of uncertain decisions ($h < 1$) becomes smaller.

5.2 CONSTRUCTIVITIES. — In figures 3-6 some typical results for the constructivity are presented. For figure 3 and figure 5 we applied simple learning, for figure 4 and figure 6 learning with noise. Figure 3 and figure 4, which have already been introduced in section 3 display the constructivity for one node of the lowest layer, namely the one which contains the letters « i » and « m ». We looked for the constructivity of « i ». Many couplings belong to the background and contain no information. Without noise only the different pixels between the two letters are marked by some constant positive value of the constructivity. Applying noise « smoothes » the constructivities. The number of big constructivities corresponding to the number of decisive pixels decreases. This is the case especially at the edges of the letters. We observed however that some few constructivities which lay outside the regions of noise became bigger, e.g. inside the lines of « m ».

Figure 5 and figure 6 present one node of layer 3 which stores 8 letters. To this node belongs a matrix row which distinguishes between {TY (y} and {!lIf}. Here we regarded the constructivity of « l ». In the plot without noise there are some discrete values of the constructivity ; again they become continuous when noise is applied.

6. Conclusion.

We developed a perceptron like network especially optimized for practical applications. It offers a high retrieval rate concerning « real » objects, works very quickly and does not consume much computer memory. We showed that the assignment of the output vectors to a given set of patterns strongly influences the effectivity of a neural net. For out hierarchical index tree structure this arrangement is crucial, so we optimized it by simulated annealing with a cost function related to the optimal stability. We modified a minimal overlap learning algorithm according to Krauth and Mezard. By this we got an enhanced retrieval of objects disturbed by a specific noise appearing in practical applications. We are certain that there are many more possibilities to still improve the performance of our index tree, which will be done in future work.

References

[1] KRAUTH W., MEZARD M., *J. Phys. A* **20** (1987) L745.

[2] For a recent review see VAN HEMMEN J. L., DOMANY E. and SCHULTEN K., *Physics of neural networks,* to be published by Springer Verlag and references therein.

[3] MINSKY M., PAPERT S., Perceptrons (MIT-press, Cambridge Ma.) 1969.

[4] DIEDERICH S., OPPER M., *Phys. Rev. Lett.* **58** (1987) 949.

[5] PÖPPEL G., KREY U., *Europhys. Lett.* **4** (1987) 979.

[6] GARDNER E., *J. Phys. A* **21** (1988) 257.

[7] GARDNER E., STROUD N., WALLACE D. J., Neural Computers, Eds. R. Eckmiller and Chr. V. D. Malsburg *NATO ASI Series F* (Berlin, Springer) **41** (1988) 251.

[8] GARDNER E., *J. Phys. A* **19** (1986) 3453.

[9] PERSONNAZ L., GUYON I., DREYFUS G., *Europhys. Lett.* **4** (1987) 863.

[10] KREY U., PÖPPEL G., Proc. of the conference Measures of Complexity held from 29 Sept. till 2nd Oct. 1987 in Rome, Eds. L. Peliti and A. Vulpiani, *Lect. Notes Phys.* (Berlin, Springer) **314** (1988) 35.

[11] FUCHS A., HAKEN H., *Biol. Cybern.* **60** (1988) 17, 107 ; *Erratum* **60** (1988) 476.

[12] MEZARD M., NADAL J. P., *J. Phys. A* **22** (1989) 2191.

[13] BUHMANN J., SCHULTEN K., *Biological Cybernetics* **54** (1986) 319.

[14] LINDEN A., KINDERMANN J., DANIP workshop held from 24 to 25 April 1989, GMD Schloß Birlinghoven, D-5205 Sankt Augustin, to be published.

[15] YOUNG I. T., PEVERINI R. C., VERBEEK P. W., VAN OTTERLOO P. J., A new Implementation for the Binary and Minkowski Operators, Computer Graphics and Image Processing 17 (1981) 189.

[16] KIRKPATRICK S., GELATT C. D., Jr, VECCHI M. P., Optimization by Simulated Annealing, *Science* **220** (1983) 671.

[17] OPPER M., *Phys. Rev. A* **38** (1989) 3824.