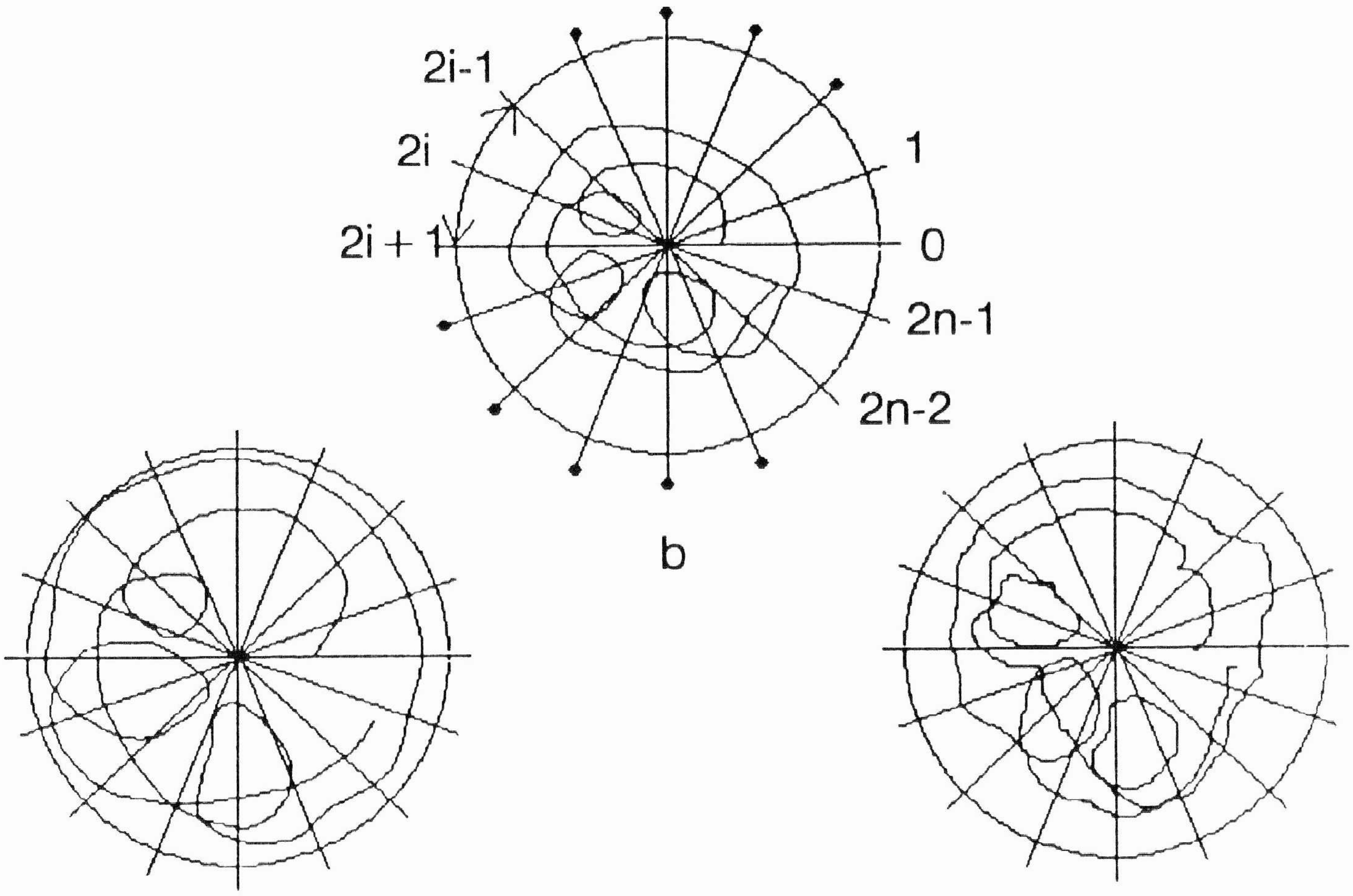
IDG

NEURAL NETWORK WORLD

International Journal on Neural and Mass-Parallel Computing and Information Systems

VOLUME 1 1991 NUMBER 4



NEURAL NETWORK WORLD is published in 6 issues per annum by the Computer World Company, Czechoslovakia, 120 00 Prague, Blanická 16, Czechoslovakia, the member of the IDG Communications, USA.

Price Information:
Subscription rate 399 US $ per annum.
One issue price: 66.50 US $.
Subscription adress: ComputerWorld Co., Czechoslovakia, 120 00 Prague 2, Blanická 16, Czechoslovakia.

Scanning the Issue

ISSN 1210-0552

Editorial

The world of neural networks extends rapidly almost in all the areas of human knowledge. Though still the interest of many people dealing in this area is concerned to the fundamental problems related to the need to improve significantly our up to now very unsatisfactory level of understanding the principles how our brains are really operating (and also how the living cells are processing and storing the information coming from the external world), also large interest is given to the questions related to the simulation of neural networks by means of mathematical and technical tools and to the possibilities of their applications for practical purposes.

For this issue we have selected three mainly theoretical oriented papers dealing with some serious problems concerning one of the most important phase of all the artificial neural network activity, which is the the learning and training procedure (Gorse and Taylor; Personnaz, Nerrand, Roussel-Ragout and Dreyfus; Bitzan) and three papers oriented more to theoretical problems related to the artificial neural network applications (Bulsari and Saxen; Chudý, Chudý and Hapák; Rybak, Golovan, Gusakova, Shevtsova and Podladchikova).

We continue also in the Hořejš's tutorial on neural networks paradigms and include some book and conference, symposiums and workshops information.

We hope that the readers will enjoy this selection.

Mirko Novák
Editor - in - Chief

UNIVERSAL ASSOCIATIVE STOCHASTIC LEARNING AUTOMATA

D. Gorse)*

*J. G. Taylor**)*

Abstract.

A generalisation of the concept of binary-input stochastic learning automata is given which incorporates non-linearity and stochasticity to a maximal degree. This universal automaton is identified with the 'probabilistic random access memory' (pRAM), a hardware-realisable neural model previously proposed by the authors. A reinforcement training rule is presented for such automata, and convergence theorems proved. The nature of the invariant measure is explored for a 1-input automaton with a two dimensional state space. The reinforcement rule is then simulated in the context of a particular classification task, and the results compared favourably with those obtained by Barto and Anandan using a less general training rule.

*) **D Gorse**
Department ov Computer Science,
University College,
Gower Street,
London WC1E 6BT

) **J G. Taylor
Department of Mathematics,
King's College,
Strand
London WC2R 2LS

1. Introduction

Automata may be considered as simplified models of living systems. The automaton interacts with an environment about which it has limited knowledge, and modifies its behaviour so as to maximise the probability of receiving 'reward' from the environment. More sophisticated automata also act so as to avoid a 'punishment' signal. Stochastic automata are those which use some intrinsic 'noise' to generate exploratory actions which are then assessed by the environment. Associative automata are aware not just of the environmental reinforcement response, but of other environmental conditions which may predispose them to select certain actions.

It is clear from the above that the notion of an automaton is one which may be applied at a variety of scales, from the behaviour of an entire organism (such an approach may be termed 'mathematical psychology') to that of its individual cells. At the lower end of this spectrum, the work of Klopf [1] has been particularly influential. Klopf proposed that individual neurons exhibit a form of goal-seeking behaviour; his ideas have been adapted by a number of workers in the neural networks field, who have proposed a variety of reinforcement training schemes which in some sense embody Klopf's original "hedonistic neuron" [1] concept.

In this paper we propose to further develop these ideas by using a more general model of the neuron than has hitherto been adpoted. In fact the model we will present has claim to being the most general stochastic neural automaton operating in the binary domain (in which we take a '1' to represent a firing event at some time, and a '0' to represent inactivity). Thus the most general associative automaton (formal adaptive neuron) A has inputs $\underline{u}$ (with the vector $\underline{u}$ having N binary components $u_1, \ldots u_N$) and a binary output y which is a stochastic variable with Bernoulli distribution defined by the variable $\alpha_{\underline{u}}$, where

$$\mathrm{Prob}(\ y=1 \mid \underline{u}\ ) = \alpha_{\underline{u}}$$

The set of all 2^N values of the $\alpha_{\underline{u}}$ will be denoted $\underline{\alpha}$. The state of A is thus given by a 2^N-component vector $\underline{\alpha}$, with $\underline{\alpha} \in [0,1]^{2^N}$. It is clear that A exhibits a response which is of maximal non-linearity in the components of the input vector $\underline{u}$. In particular, in the deterministic case (when $\underline{\alpha} \in \{0,1\}^{2^N}$) it can be seen that A can compute any of the 2^{2^N} possible Boolean functions of its inputs. In the language of PDP [2], A is the ultimate $\Sigma-\Pi$ unit.

The idea of a stochastic associative learning automaton was analysed in detail in [3] (and earlier papers by Barto and colleagues) but only with respect to the restricted form of input-output relationship

$$\mathrm{Prob}(\ y=1 \mid \underline{u}\ ) = \Theta(\ \underline{u} \,.\, \underline{w} - \eta\ ) \qquad (2)$$

where $\underline{w}$ is an N-vector of real-valued weights and η is a random variable. Our approach extends this model in two ways.

The first, as indicated above, is in the highly nonlinear dependence of the output probability on the input vector components (which in the language of [3] represents the context in which A is responding). Our model is essentially identical with the full $\Sigma-\Pi$ extension of (2)

$$\mathrm{Prob}(\ y=1 \mid \underline{u}\ ) = \Theta(\ \underline{u} \,.\, \underline{w} + \sum_{ij} u_i u_j w_{ijk} + + \sum_{i,j,k} u_i u_j u_k w_{ijk} + \cdots - \eta) \qquad (3)$$

although we have preferred to work in terms of the 2^N parameters $\underline{\alpha}$ rather than the sets of weights w_i, $w_{ij \ldots}$; we feel that this allows the analysis to be presented more cleanly.

The second extension of (2) is in the assumption of stochasticity of all the parameters $\alpha_{\underline{u}}$, rather than having a single random variable, the threshold η. This extension was motivated by a model of synaptically noisy neurons developed earlier by one of us [4], which has led to the pRAM [5] and to a theoretical analysis of noisy neurons in terms of random iterative networks [6], extending the spin glass approach to neural networks [7] to a more realistic level. The pRAM represents a hardware realisation of this noisy neural model, in which the variables $\alpha_{\underline{u}}$ are output probabilities stored in the 2^N memory locations of a random access memory; this can be regarded as a neurobiologically motivated extension of the PLN model of Aleksander [8].

As in [3] we consider variable structure stochastic automata in which there are specific rules for updating the parameters $\alpha_{\underline{u}}$ of (1) at each time step k. These rules will be assumed to be of the reinforcement kind, in which the environment E emits a random variable r, on receipt of A's action y in context u. The signal r = 1 is to be regarded as a reward, and the aim of the reinforcement rule is to update the $\alpha_{\underline{u}}$ so as to increase the probability of obtaining the reward r = 1, rather than the penalty r = 0. In general, if r = 1 is obtained in context $\underline{u}$, the quantity $\alpha_{\underline{u}}$, is changed by some function $f(\alpha_{\underline{u}})$ so as to keep the new value $\alpha_{\underline{u}} + \acute{u}(\alpha_{\underline{u}}) \in [0,1]$, and improve the probability of choosing the same action y in that context on future occasions (note that the probability of the alternative action ($y \in \{0,1\}$) is automatically decreased):

$$\alpha_{\underline{u}} \rightarrow \alpha_{\underline{u}} + f(\alpha_{\underline{u}}), \quad r = 1 \qquad (4a)$$

If r = 0 is obtained, $\alpha_{\underline{u}}$ is changed by some function $g(\alpha_{\underline{u}}$ to make the chosen action less likely. again ensuring that the new value of $\alpha_{\underline{u}} \in [0,1]$, and this time increasing the probability of the alternative action:

$$\alpha_{\underline{u}} \rightarrow \alpha_{\underline{u}} + g(\alpha_{\underline{u}}), \quad r = 0 \qquad (4b)$$

If

$$\mathrm{Prob}(\ r=1 \mid y, \underline{u}\ ) = \beta_{y.\underline{u}} \qquad (5)$$

then the enviroment itself may be regarded as acting as an (N+1)-input pRAM, with 2^{N+1}-component memory content vector $\underline{\beta}$. We may thus visualise the situation as in Figure 1.

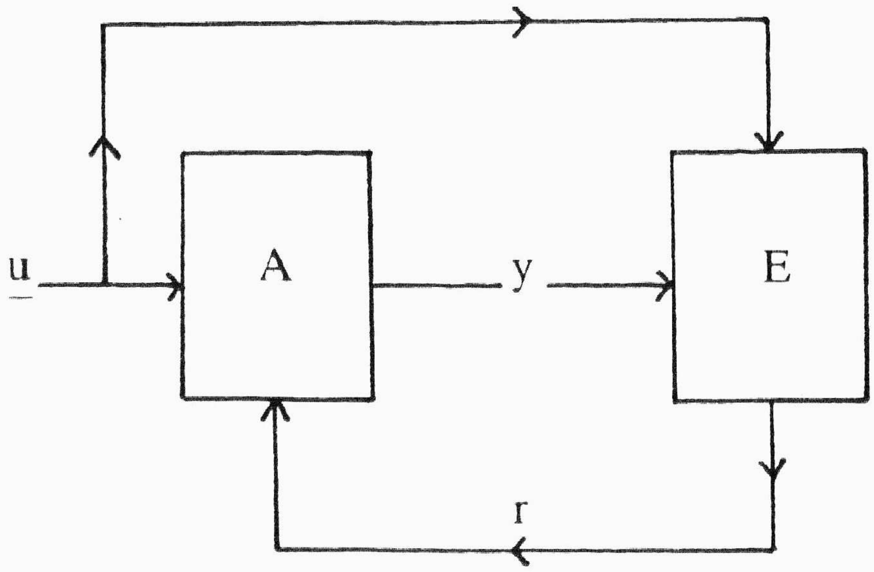


Figure 1

The universal associative stochastic automaton A has binary N-vector context input $\underline{u}$ and binary stochastic output y. The output y and the context $\underline{u}$ are evaluated by the environment A which responds stochastically with a binary reward signal r.

As in the case of [3] it is assumed that the values of the $\beta_{y,\underline{u}}$ are not known by the automaton A, and that the purpose of the training rules (4) is to produce that behaviour by A which maximises the probability of receiving a reward $r = 1$. Let $\underline{P}$ be a probability distribution over the input addresses $\underline{u}$ with $\sum_{\underline{u}} P_{\underline{u}} = 1$. If this distribution is independent of A's actions (as in the classification task described below) the enviroment is *stationary*, otherwise (as in the bug example) it is termed *non-stationary* [9]; we will assume for simplicity that the former is the case. A performance criterion for the nth of a series of trials is given by the expected probability of success at that trial:

$$E(r_n) = Q_n = \sum_{y,\underline{u}} P_{\underline{u}} \beta_{y,\underline{u}} [\alpha_{\underline{u}}^{(n)} y + \bar{\alpha}_{\underline{u}}^{(n)} \bar{y}] \tag{6}$$

(where the $\alpha_{\underline{u}}^{(n)}$ are the values of the parameters (1) at trial n, and $\bar{x} \equiv 1\text{-}x$ for any x). In terms of the average expected success rate

$$Q_0 = \sum_{\underline{u}} \frac{1}{2} P_{\underline{u}} (b_{1,\underline{u}} + \beta_{0,\underline{u}}) \tag{7}$$

the natural extensions of the concepts *expedient* and *optimal* are

$$\lim_{n \to \infty} E(Q_n) > Q_0 \tag{8a}$$

and

$$\lim_{n \to \infty} E(Q_n) = \sum_{\underline{u}} P_{\underline{u}} \max \{\beta_{1,\underline{u}}, \beta_{0,\underline{u}}\} \tag{8b}$$

(with similar definitions for the ε-versions of the above). A general discussion of these ideas is contained in [9], and the references contained therein.

The contents of the paper are as follows. In the next section the mathematical framework for the analysis to follow is established; this is based on the work of Norman [10]. In Section 3 the convergence properties of a learning scheme of particular interest, the associative reward-penalty (A_{R-P}) scheme, are analysed (this algorithm is one which is wholly implementable in pRAM technology [11]). The dynamical behaviour of the simplest (N=1) pRAM system is simulated for various values of learning rate parameters, and these results are compared with the theoretical predictions. In the following section the performance of the pRAM form of the A_{R-P} algorithm is compared with that more conventionally based on equation (2) and is found to represent a significant improvement on the Barto algorithm [3]. A brief discussion of the results presented in the paper, future extensions of the model, and possible applications is contained in the final section.

2. General Definitions

In this section we propose to make precise the structure of the automaton we are considering. The description will be based on the introductory discussion of the previous section. The stochastic automaton A operates, according to equation (1), to produce an output y with probability $\alpha_{\underline{u}}$ of being 1 for context $\underline{u}$. Its state space S is the set of all 2^N values of the $\alpha_{\underline{u}}$'s, indexed by the binary N-vector $\underline{u}$:

$$S = [0,1]^{2^N} \tag{9}$$

Its event space E, the set of triplets ($\underline{u}$, y, r), is the space

$$E = \{0,1\}^{2N+2} \tag{10}$$

The environment responds to $\underline{u}$ and y with a reinforcement signal r, according to equation (5), with probability $\beta_{y,\underline{u}}$ of r being 1 (reward). At each time step n the vector $\underline{\alpha}^{(n)}$ is updated according to the rules (4). In other words, for every event $\underline{e}_n = (\underline{u}_n, y_n, r_n)$ there must be a rule for changing $\underline{\alpha}^{(n)}$ to $\underline{\alpha}^{(n+1)}$ which depends on the event $\underline{e}_n$, expressible as

$$\underline{\alpha}^{(n+1)} = F_{\underline{e}n} (\underline{\alpha}^{(n)}) \tag{11}$$

where $F_{\underline{e}}$ is given in terms of f and g in an evident manner. The purpose of the learning rule (11) is to make A respond either optimally or expediently (or within ε of either). It can be seen that the events $\underline{e}$ occur with probability $\Phi_{\underline{e}}(\underline{\alpha})$, where

$$\Phi_{\underline{e}}(\underline{\alpha}) = [\alpha_{\underline{u}} \delta_{y,1} + \bar{\alpha}_{\underline{u}} \delta_{y,0}] [\beta_{y,\underline{u}} r + \bar{\beta}_{y,u} \bar{r}] P_{\underline{u}} \tag{12}$$

The rule above derives straightforwardly from (1) and (5), since, for example, the probability that the addressing of location $\underline{u}$ yields output y=1, and reward reinforcement r=1 is $\alpha_{\underline{u}} \beta_{1,\underline{u}}$, which must be multiplied by $P_{\underline{u}}$, the probability that location $\underline{u}$ was chosen.

The transition probability or kernel K for the automation is defined [10], for any subset B of the state space S, as

$$K(s, B) = \sum_{\underline{e}} \varphi_{\underline{e}}(s) \tag{13}$$

where the summation in (13) is over those events $\underline{e}$ for which $F_{\underline{e}}(s) \in B$. If the characteristic function of the set B in the state space S is denoted $\delta(B, s)$, so that

$$\begin{aligned} \delta(B, s) &= 1, \quad s \in B \\ &= 0, \quad s \notin B \end{aligned} \tag{14}$$

then (13) may be written as

$$K(s, B) = \sum_{\underline{e}} \Phi_{\underline{e}}(s)\, \delta(B, F_{\underline{e}}(s)) \tag{15}$$

where there is a Markov operator M induced by K on the Borel measure μ on S by

$$M(\mu)(B) = \int K(s, B)\, d\mu(s)$$

The asymptotic behaviour of the automaton may be described by means of the nth iterate of K, denoted $K^{(n)}$, with

$$K^{(n)}(s, B) = \sum_{\underline{e}_1, \ldots, \underline{e}_1} \Phi_{\underline{e}1}(s)\Phi_{\underline{e}2}(s))\ldots \Phi_{\underline{e}n}(F_{\underline{e}1\ldots\underline{e}n}(s)\ldots) \\ \delta(B, F_{\underline{e}1\ldots\underline{e}n}(s)) \qquad (16)$$

and $F_{\underline{e}1\ldots\underline{e}n} \equiv F_{\underline{e}1}\cdots . F_{\underline{e}n}$. The convergence of $K^{(n)}$ to a limiting kernel K_∞, due to the finiteness of S, will reduce to the uniform convergence of $K^{(n)}(s, B)$ to $K_\infty(s, B)$ in s and B: for any $\varepsilon > 0$, there is an integer N such that

$$| K^{(n)}(s, B) - K_\infty(s, B) | < \infty$$

for all $n \geq N$, $s \in S$ and $B \subset S$.

Norman [10] (see also [15]) derived powerful theorems for stochastic associative learning automata for which the state space S is a compact metric space, and for which all event operators $F_{\underline{e}}$ are strictly diminishing in the sense that, $\forall\ \underline{e} \in E$,

$$\sup_{s \neq s'} [d(F_{\underline{e}}(s), F_{\underline{e}}(s')) / d(s, s')] < 1 \qquad (17)$$

Since in our case S is $[0,1]^{2^N}$, in has the natural Euclidean distance function derived from $\mathbf{R}^{2^N}$; we take d to be that function. Then (S, d) is compact. Let us suppose that $F_{\underline{e}}$ is distance diminishing. Then it may be seen that one of two possibilities must occur [10]:

(a) the asymptotic distribution K_∞ is independent of the initial state; the associated (invariant) measure μ is then ergodic.
(b) there are a finite number of absorbing states s_i, and the asymptotic kernel K_∞ (or the measure μ_∞) is concetrated on these states.

The two possibilities (a), (b) are given by properties of the set of points $T_n(s)$ that a given point s iterates to with positive probability after n time steps:

$$T_n(s) = \{s': K^{(n)}(s, \{s'\}) > 0\} \qquad (18)$$

Then (a) will occur if

$$\lim_{n \to \infty} d(T_n(s), T_n(s')) = 0, \quad s, s' \in S \qquad (19)$$

whilst (b) will be the case if there are a finite set of absorbing states $a_1 \ldots, a_N$ with the property that $\forall s \in S$, $\exists$ some j(s) such that

$$\lim_{n \to \infty} d(T_n(s), a_{j(s)}) = 0 \qquad (20)$$

In this case there is a set of non-zero probabilities of arriving at each of the absorbing states from a given initial state, of the form

$$\mathrm{Prob}(s_\infty = a_i \mid s_0 = s)$$

The automaton is then termed *non-ergodic* or *absorbing*.

We note that in the ergodic case the measure μ_∞ may be either continuous or fractal [12]; we will consider these features in more detail for the simplest non-trivial case ($S = [0,1]^2$) in the next section.

3. The A_{R-P} Algorithm

We will now concentrate our attention on the extension of the associative reward-penalty algorithm of [3] to the case of the universal automaton considered in the previous two sections. This algorithm has proven to be of considerable utility in various neural network applications, and we would hope that the extension of the algorithm in the directions indicated above would result in still more powerful learning systems. The original A_{R-P} algorithm was based upon weight updates within the restricted input-output relation (2); as indicated above, our approach extends this by working directly with the 2^N random variables (1), utilising a maximal degree of non-linearity and synaptic rather than threshold noise.

The natural extension of the A_{R-P} training rule of [3] appears to be

$$F_{\underline{e}}(\alpha_{\underline{u}}) = \alpha_{\underline{u}} + \rho(y - \alpha_{\underline{u}})\, r + \rho\lambda(\bar{y} - \alpha_{\underline{u}})\, \bar{r} \qquad (21)$$

It may be seen that (21), like the original A_{R-P} algorithm, achieves the general aim of reinforcement training rules, to encourage behaviour which has led to a reward and to discourage behaviour which has led to punishment: if there is a reward ($r = 1$), the probability $\alpha_{\underline{u}}$ stored at address $\underline{u}$ is changed to be closer to the output y which produced the favourable environmental response; if there is punishment ($r = 0$), that address is made more likely to output the opposite value $\bar{y}$ when accessed. The rates of these two processes of change are governed by the independent parameters ρ and $\rho\lambda$.

It is instructive to consider the learning rule (21) in the special case which corresponds to the simple binary threshold model (2) with a single noisy synapse with weight w, and the possibility of releasing only one vesicle of neurotransmitter at any given time step (single vesicle model). In this case the distribution function f for w is given by

$$f(w) = p\, \delta(w - 1) + \bar{p}\, \delta(w) \qquad (22)$$

where p is the probability of firing of this single neuron model. Then, for a positive threshold η and binary input $u \in \{0,1\}$,

$$\mathrm{Prob}(y=1 \mid u) = \alpha_u = \int f(w)\, dw\, \Theta(wu - \eta) \\ = p\, \Theta(u - \eta)$$

The learning rule (21) becomes, for $u = 1 > \eta$,

$$\Delta p = \rho(y - p)r + \rho\lambda(\underline{y} - p)\bar{r} \tag{23}$$

There are two possible actions, $y = 1$ and $y = 0$, which occur with probabilities p, $\bar{p}$ respectively. When $r = 1$, (23) becomes

$$p_{n+1} = p_n + \rho(1 - p_n) \qquad \bar{p}_{n+1} = \bar{\rho}\,\bar{p}_n \tag{24a}$$

whilst when $r = 0$, this equation takes the form

$$p_{n+1} = \overline{\rho\lambda}\,p_n + \rho\lambda \qquad \bar{p}_{n+1} = \overline{\rho\lambda}\,\bar{p}_n \tag{24b}$$

(24a,b) are exactly the rules of the Barto and Anandan A_{R-P} algorithm (see [3], equations (1) and (2), with $\alpha = \rho$, $\beta = \rho\lambda$) for an automaton with a binary output set. This justifies our claim that the learning rule (21) is to be regarded as a natural extension of the Barto A_{R-P} algorithm to the case where synaptic as well as threshold noise is present, and where the firing rule is the most general Boolean function of the inputs, as described in the previous two sections.

We now return to the general form of the A_{R-P} rule (21), and analyse its asymptotic behaviour. As was discussed in the previous section, this behaviour is determined by the conditions (17), and (19) or (20). Let us first consider (17). Using rule (21), (17) takes the particular form

$$\sup_{\underline{\alpha} \neq \underline{\alpha}} \{ d(\alpha_{\underline{u}} + \rho(y - \alpha_{\underline{u}})r + \rho\lambda(\bar{y} - \alpha_{\underline{u}})\bar{r},\ \alpha_{\underline{u}} + \rho(y - \alpha_{\underline{u}})r + \rho\lambda(\bar{y} - \alpha_{\underline{u}})\bar{r} \,/\, d(\underline{\alpha}, \underline{\alpha}) \} < 1 \tag{25}$$

The numerator on the left hand side of (25) reduces, using the Euclidean distance function $d(\ ,\ )$ on $\mathbf{R}^{2N}$, to

$$(1 - \rho r - \rho\lambda\bar{r})\, d(\alpha_{\underline{u}}, \alpha_{\underline{u}})$$

Since $d(\alpha_{\underline{u}}, \alpha_{\underline{u}}) \leq d(\underline{\alpha}, \underline{\alpha})$ for any α, α, then since $(1 - \rho r - \rho\lambda\bar{r}) < 1$, condition (17) is satisfied; the A_{R-P} algorithm (21) corresponds to a strictly distance diminishing learning automaton.

In order to distinguish between the ergodic and absorbing cases of the last section it is necessary to determine which of conditions (19), (20) are true. In order to do this the set of first interates $T_n \equiv T_1$ is investigated for the case (21). The absorbing points of the automaton are determined first, from the following lemma (which we clarify here, although this result is well-known in the literature ([15], Chapter 5)):

Lemma

The absorbing points of the automaton (21) (with event space E given by (10)) are, for $\rho \neq 0$, $\beta_{i,\underline{u}} \neq 0, 1$
(i) absent if $\lambda \neq 0$
(ii) equal to the boundary points ($\underline{\alpha} \in \{0, 1\}^{2N}$) if $\lambda = 0$

The lemma is proved by noting that absorbing points are those which satisfy, for any $\underline{e} \in E$,

$$F_{\underline{e}}(\underline{\alpha}) = \underline{\alpha} \tag{26a}$$

or, if this is not the case

$$\Phi_{\underline{e}}(\underline{\alpha}) = 0 \tag{26b}$$

We may catalogue the values of $F_{\underline{e}}$ and $\Phi_{\underline{e}}$ for different values of r and y:

r	y	$F_{\underline{e}}$	$\Phi_{\underline{e}}$
0	0	$\alpha_{\underline{u}} + \rho\lambda\bar{\alpha}_{\underline{u}}$	$\bar{\alpha}_{\underline{u}}\bar{\beta}_{1,\underline{u}}P_{\underline{u}}$
0	1	$\alpha_{\underline{u}} - \rho\lambda\alpha_{\underline{u}}$	$\alpha_{\underline{u}}\bar{\beta}_{0,\underline{u}}P_{\underline{u}}$
1	0	$\alpha_{\underline{u}} - \rho\alpha_{\underline{u}}$	$\bar{\alpha}_{\underline{u}}\beta_{0,\underline{u}}P_{\underline{u}}$
1	1	$\alpha_{\underline{u}} + \rho\alpha_{\underline{u}}$	$\bar{\alpha}_{\underline{u}}\beta_{0,\underline{u}}P_{\underline{u}}$

Table 1

If $\lambda \neq 0$, then by inspection of the above table it is not possible to satisfy either of (26a), (26b) for arbitrary β's, so there are no absorbing points. On the other hand, if $\lambda = 0$, and again the β's are non-zero, then any boundary point $\underline{\alpha} \in \{0, 1\}^{2N}$ automatically satisfies (26). The lemma is therefore proven. □

We note that in the case that $\lambda = 0$, and some of the β's are zero, there will be further absorbing points away from the boundary, as is clear from the table; this situation will not be discussed further here. The question of most interest is the nature of the automaton in the case that there are no absorbing points. This question will be addressed in the following theorem (see [15], p. 162):

Theorem 1

If $\lambda \neq 0$ and none of the $\beta_{i,\underline{u}} \in \{0, 1\}$ (so that there are no absorbing points), the automaton (21) is ergodic.

The theorem is proved by the formula

$$d(T_1(\underline{\alpha}), T_1(\underline{\alpha})) \leq c\, d(\underline{\alpha}, \underline{\alpha}) \tag{27}$$

for $c < 1$. The left hand side of (27) is equal to

$$\min \| \underline{\beta} - \underline{\beta}' \| \tag{28}$$

where $\| \ \|$ denotes the Euclidean distance, and the minimum is taken over $\underline{\beta} \in T_1(\underline{\alpha})$, $\underline{\beta}' \in T_1(\underline{\alpha})$. From the definition (18) we may take $\underline{\beta} = F_{\underline{e}}(\underline{\alpha})$, $\underline{\beta}' = F_{\underline{e}'}(\underline{\alpha})$ for some $\underline{e}$, $\underline{e}' \in E$, with $\Phi_{\underline{e}}(\underline{\alpha}) > 0$, $\Phi_{\underline{e}'}(\underline{\alpha}) > 0$. Then (28) may be seen to be equal to

$$\min_{\underline{e}, \underline{e}' \in E} \| F_{\underline{e}}(\underline{\alpha}) - F_{\underline{e}'}(\underline{\alpha}) \| \tag{29}$$

A bound to (29) is obtained by taking $\underline{e} = \underline{e}'$, with

$$F_{\underline{e}}(\underline{\alpha}) - F_{\underline{e}}(\underline{\alpha}) = (1 - \rho r - \rho\lambda\bar{r})(\alpha_{\underline{u}} - \alpha_{\underline{u}}) \quad (30)$$

Thus the formula (27) is valid, with

$$c = 1 - \rho r - \rho\lambda\bar{r} < 1$$

Therefore the condition (19) is valid, and by the results of Norman [10] the theorem is proven. □

We note that the inequality (31) need not be valid for $\lambda = 0$ if the only values of r involved in the minimisation in (30) are zero. This is to be expected, since in the case $\lambda = 0$ all the boundary points in the state space $S = [0, 1]^{2^N}$ are absorbing points and the condition (20) must be true in that case; condition (19) is no longer valid.

The N = 1 Case

It is of interest to examine the structure of the invariant measure μ_∞ in the case of a 1-input pRAM whose memory contents $\underline{\alpha} = (\alpha_0, \alpha_1)$ are updated according to (21), since in this case the state space $S = [0, 1]^2$, allowing visualisation. At each time step one of eight update rules is chosen; these are indexed by the event $\underline{e} = (u, y, r)$:

Map	Probability
$F_{(0,0,0)}(\underline{\alpha}) = (\bar{\lambda}\rho\alpha_0 + \lambda\rho, \alpha_1)$	$\bar{\alpha}_0\bar{\beta}_{00}P_0$
$F_{(0,0,1)}(\underline{\alpha}) = (\bar{\rho}\alpha_0\ \alpha_1)$	$\bar{\alpha}_0\beta_{00}P_0$
$F_{(0,1,0)}(\underline{\alpha}) = (\bar{\lambda}\rho\alpha_0, \alpha_1)$	$\alpha_0\bar{\beta}_{01}P_0$
$F_{(0,1,1)}(\underline{\alpha}) = (\bar{\rho}\alpha_0 + \rho, \alpha_1)$	$\alpha_0\beta_{01}P_0$
$F_{(1,0,0)}(\underline{\alpha}) = (\alpha_0\ \overline{\lambda\rho}\alpha_1 + \lambda\rho)$	$\bar{\alpha}_1\bar{\beta}_{10}P_1$
$F_{(1,0,1)}(\underline{\alpha}) = (\alpha_0, \bar{\rho}\alpha_1)$	$\bar{\alpha}_1\beta_{10}P_1$
$F_{(1,1,0)}(\underline{\alpha}) = (\alpha_0, \overline{\lambda\rho}\alpha_1)$	$\alpha_1\bar{\beta}_{11}P_1$
$F_{(1,1,1)}(\underline{\alpha}) = (\alpha_0, \bar{\rho}\alpha_0 + \rho)$	$\alpha_1\beta_{11}P_1$

Table 2

Successive iterations thus generate a random trajectory in the unit square. The set of mappings above forms a random Iterated Function System (IFS); analysis of the behaviour of learning automata in terms of IFS's goes back to the work of Karlin [12]. Random IFS's have been recently used to describe the behaviour of associative stochastic learning automata by Bressloff and Stark [13], who based their discussion on a simple linear reward-penalty scheme. In particular it was shown in [13] that in a 2-action automaton with state space $S = [0,1]$, governed by a single learning rate parameter, the support of μ_∞ was fractal when the learning rate was greater than 0.5 and continuous otherwise. In our case we have two parameters to vary (ρ, λ) and a 2-dimensional state space. For $\lambda = 0$ it could be seen that the boundary points of S, (0,0), (0,1), (1,0), (1,1), were indeed absorbing (as indicated by the Lemma above) whilst for $\lambda \neq 0$ the invariant measure was continuous for $\rho \leq 0.5$, fractal for $\rho > 0.5$. For fixed ρ it was found that the measure became progressively smoother as $\lambda \rightarrow 0$. Figures 2(a)-(c) show the results of 10^7 iterations of the above IFS, using $\beta_{\underline{u}} = \frac{1}{4}$, $P_{\underline{u}} = \frac{1}{2}$, $\lambda = 0.75$ and $\rho = 0.25$, 0.50, (continuous measure) and 0.75 (fractal).

It is important to understand the implications of the above results for practical reinforcement training problems, in particular to appreciate the difference between the concepts of 'strong convergence' and 'convergence in the mean'. Convergence in the mean only ensures that the values of the $\alpha_{\underline{u}}$ converge when averaged over many runs; an individual run is typically characterised by the ergodic behaviour displayed in Figure 2. It is strong convergence which is required for practical purposes, since this will guarantee that $\underline{\alpha}$ converges to an appropriate limiting value during a particular run. Convergence in the mean can be demonstrated when the learning rate ρ is held constant ([10]), whilst strong convergence requires that ρ be appropriately decrement at each time step ([3]).

We will start our investigation of the A_{R-P} training rule (21) with the case in which ρ is constant. Strong convergence cannot be expected here, but the theorem 2.2 of [10] applies directly. We may quote this result in a form relevant to our situation:

Theorem 2

If $\lambda \neq 0$, then in the IFS (10), (21) the random variable $\alpha_{\underline{u}}$ converges in the mean to a unique value $\alpha_{\underline{u}}^{(\infty)}$ uniformly with respect to its initial value, with

$$| E(\alpha_{\underline{u}}^{(n)}) - E(\alpha_{\underline{u}}^{(\infty)}) | \leq C\alpha^{(n)}$$

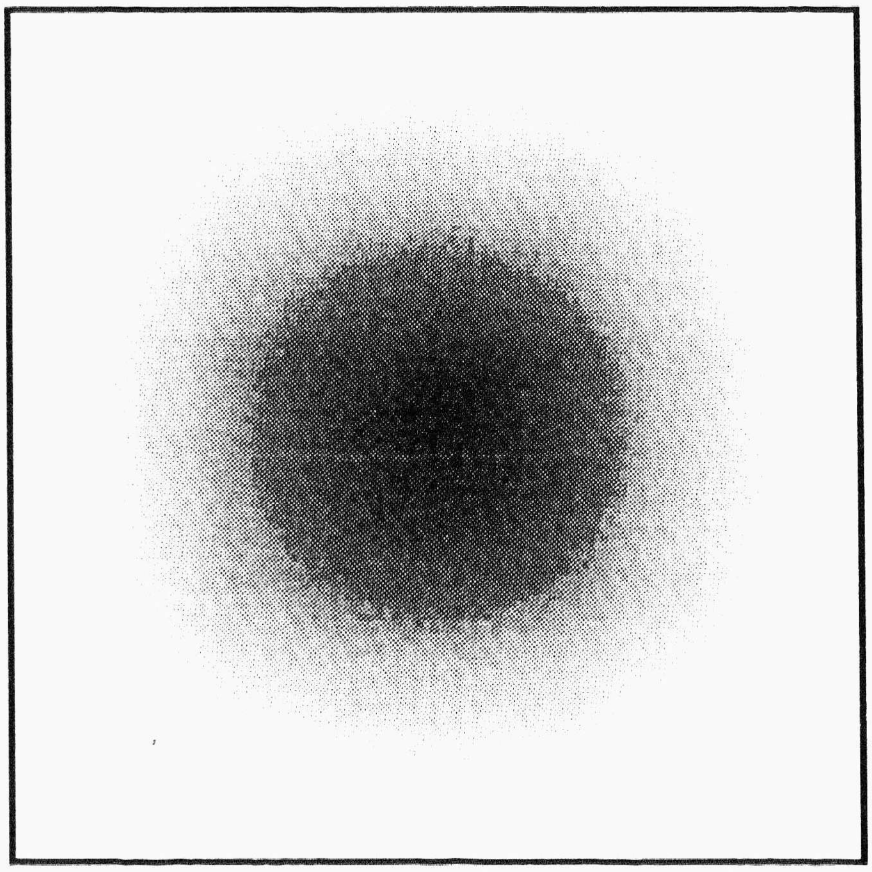


Fig. 2a

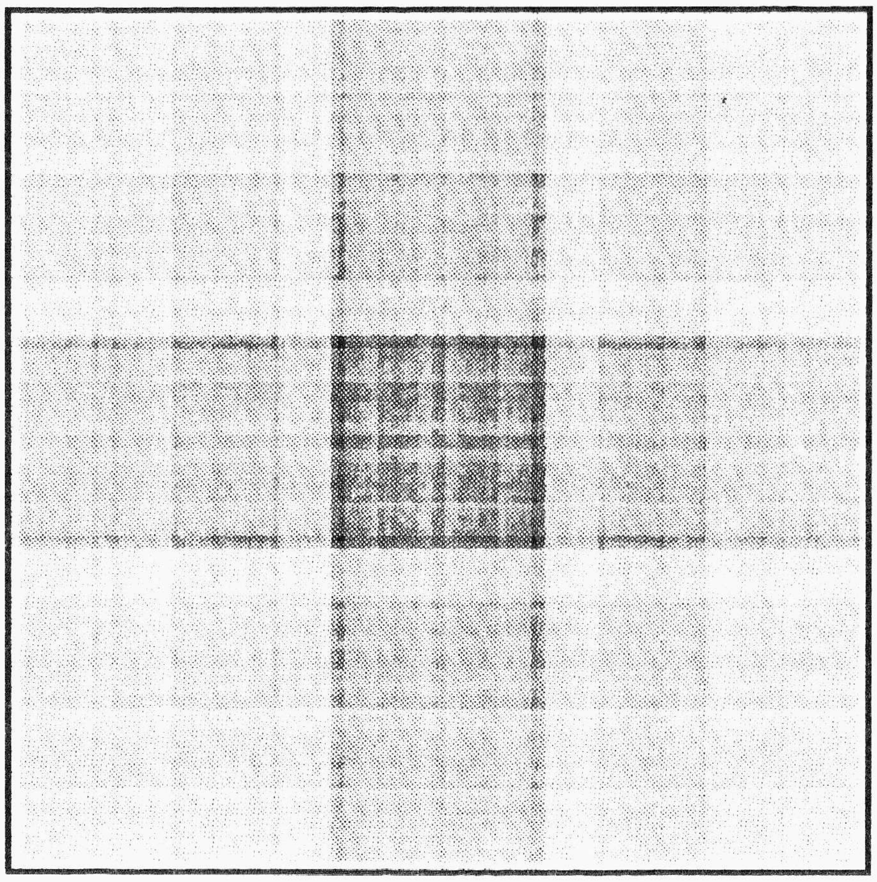


Fig. 2b

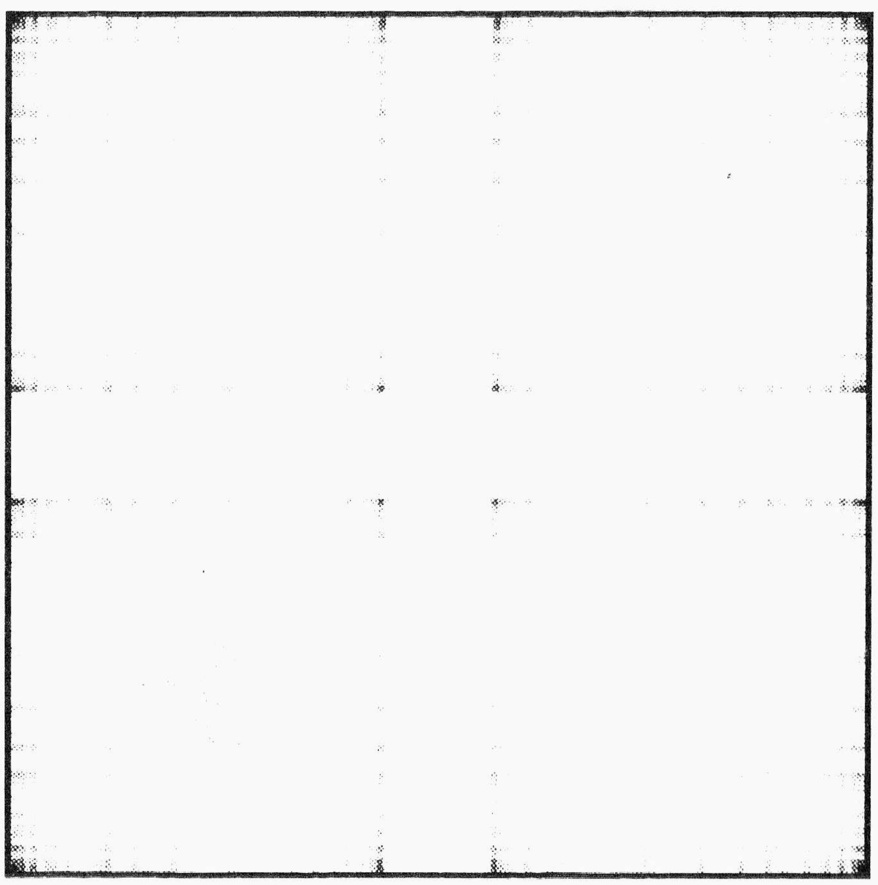


Fig. 2c

Figure 2

Invariant measure of the random IFS defined by Table 2 for $\lambda = 0.75$ and

(a) $\rho = 0.25$
(b) $\rho = 0.50$
(c) $\rho = 0.75$

where Ψ and α are constants with $\alpha < 1$ and $E(\alpha_{\underline{u}}^{(v)}) = \int_S d\alpha_{\underline{u}} K^{(n)}(\beta, \underline{\alpha})$, with $K^{(n)}$ defined in (16).

This theorem follows from Theorem 1 and [10], since when $\lambda \neq 0$ our IFS is ergodic and theorem 2.2 of [10] is then applicable. We will consider in detail the derivation of $\alpha^{(\infty)}$ for a particular example in the following section; in general the limiting $\alpha_{\underline{u}}^{(\infty)}$ can be obtained from the stationarity condition $E(\alpha_{\underline{u}}^{(n)}) - E(\alpha_{\underline{u}}^{(\infty)}) = 0$.

The case in which $\rho \to 0$ may be investigated by extending the strong convergence theorem of [3]. In fact this extension is straightforward on using the fact that we are able to work directly with the probabilities $\alpha_{\underline{u}}$ rather than with a set of weights (see the discussion in Section 1). The lemmata 1, 2 and 4 can be taken directly from Appendix II of [3] to give a result which may be seen as a traightforward application of stochastic approximation methods:

Theorem 3

Provided each $P_{\underline{u}}$ is strictly positive and the sequence $\rho_n \geq 0$ is such that $\sum_n \rho_n = \infty, \sum_n \rho_n^2 < \infty$, there exists a unique $\alpha^{(0)}(\lambda) \in S = [0,1]^{2^N}$ such that the random sequence $\{\underline{\alpha}^{(n)}\}$ generated in S by the A_{R-P} algorithm converges to $\underline{\alpha}^{(0)}(\lambda)$ with probability 1, with

$$\underline{\alpha}^{(0)}(\lambda) > 0.5 \quad \text{if } \beta_{1,\underline{u}} > \beta_{0,\underline{u}}$$
$$\underline{\alpha}^{(0)}(\lambda) < 0.5 \quad \text{if } \beta_{1,\underline{u}} < \beta_{0,\underline{u}}$$

This proves the expediency of the algorithm $\forall \lambda, \rho$. In addition, for all $\underline{u}$,

$$\lim_{\lambda \to 0} \underline{\alpha}_{\underline{u}}^{(0)}(\lambda) = 1 \quad \text{if } \beta_{1,\underline{u}} > \beta_{0,\underline{u}}$$
$$= 0 \quad \text{if } \beta_{1,\underline{u}} < \beta_{0,\underline{u}}$$

The A_{R-P} algorithm is therefore ε-optimal.

In short, if each $P_{\underline{u}}$ is assumed to be strictly positive, each context is experienced infinitely often. Hence each automation converges in the sense appropriate to the algorithm used.

4. Comparison with the Barto A_{R-P} Algorithm

The performance of the pRAM form (21) of A_{R-P} training may be compared with that based on equation (2) and developed by Barto, Sutton and others. As remarked above, we would expect the pRAM version of A_{R-P} training to be more effective due both to the maximally non-linear input-output relationship implied by (1) and due to the inclusion of synaptic noise.

The algorithms were compared in the context of a simple classification task presented by Barto and Anandan [3]. The task required a discrimination to be made between the two input vectors $\underline{x}^{(1)} = (1, 0)$ and $\underline{x}^{(2)} = (1, 1)$, which were equally likely to occur at each time step ($P_{10} = P_{11} = 0.5$). The desired output is a 1 in response to pattern $\underline{x}^{(1)}$, a 0 in response to $\underline{x}^{(2)}$. Context-dependent reward probabilities (5) are given by the parameters $\beta_{0,10}$, $B_{1,11}$, $\beta_{0,11}$, $B_{1,11}$. In order to make the task non-trivial, $\beta_{0,10}, \beta_{1,11} \neq 0$ and $\beta_{1,10}, B_{0,11} \neq 1$. Thus the automaton has to learn to cope with an environment which is responding with somewhat contradictory reinforcement signals.

Following Barto and Anandan we use Q_n, the expected probability of success at the nth trial, as a measure of the performance of the system. In this example our general expression (6) becomes

$$Q_n = \frac{1}{2}\{\bar{\alpha}^{(n)}_{10}\beta_{0,10} + \alpha^{(n)}_{10}\beta_{1,10} + \bar{\alpha}^{(v)}_{11}\beta_{0,11} + \alpha^{(n)}_{10}\beta_{1,11}\} \tag{32}$$

Initially $\alpha_{10} = \alpha_{11} = 0.5$, so that $Q_0 = 0.5$. Q is maximised when the optimal action for each input pattern is chosen with probability 1, in which case it is given by

$$Q_{max} = \frac{1}{2}\{\max(\bar{\alpha}^{(n)}_{10}\beta_{0,10}, \alpha^{(n)}_{10}\beta_{1,10}) + \max(\bar{\alpha}^{(n)}_{11}\beta_{0,11}, \alpha^{(n)}_{10}\beta_{1,11})\}$$

The average increment to α_{1u} expected on the nth trial is given by

$$< \Delta\alpha_{1u} > = \rho\{(\overline{\alpha^{(n)}_{1u}\beta_{1,1u} - \alpha^{(n)}_{0u}\beta_{0,1u}})\alpha_{1u} + \lambda(\alpha^{(n)}_{1u}\bar{\beta}_{0,1u} - \alpha^{(n)}_{0u}\bar{\beta}_{1,1u})\bar{\alpha}_{1u}\} \tag{33}$$

Asymptotic (mean) values for the α_{1u} may be obtained by setting (33) to zero; these values are given for $\beta_{0,1u} \neq \beta_{1,1u}$ by

$$\alpha^{\infty}_{1u} = \frac{(\beta_{0,1u} - \beta_{1,1u}) + 2\lambda\bar{\beta}_{0,1u} - [(\beta_{0,1u} - \bar{\beta}_{1M1z})^2 + 4\lambda^2\bar{\beta}_{0,1u}\bar{\beta}_{1,1u}]^{1/2}}{2(\beta_{0,1u} - \beta_{1,1u})\lambda} \quad \lambda \neq 1 \tag{34a}$$

$$= \frac{\bar{\beta}_{1,1u}}{2 - \beta_{0,1u} - \beta_{1,1u}} \quad \lambda = 1 \tag{34b}$$

If $\beta_{0,1u} = \beta_{1,1u}$, so that the automaton is equally likely to be rewarded for either action, $\alpha^{\infty}_{1u} = 0.5$. The asymptotic (mean) value of Q, Q_∞, may be obtained using (32) and (33):

$$Q_\infty = \frac{1}{2\lambda}\sum_u [\beta_{0,1u} + \beta_{1,1u} - 2\lambda + [(\beta_{0,1u} - \beta_{1,1u})^2 + 4\lambda^2\bar{\beta}_{0,1u}\bar{\beta}_{1,1u}]^{\frac{1}{2}}] \quad \lambda \neq 1 \tag{35a}$$

$$= \frac{\beta_{0,1u}\bar{\beta}_{0,1u} + \beta_{1,1u}\bar{\beta}_{1,1u}}{2 - \beta_{0,1u} - \beta_{1,1u}} \quad \lambda = 1 \tag{35b}$$

Task 1

In this case

$\beta_{0,10} = 0.2$, $B_{1,10} = 0{,}9$, $\beta_{0,11} = 0.8$, $\beta_{1,11} = 0.1$

Optimal performance would be obtained by setting $\alpha_{10} = 1$, $\alpha_{11} = 0$, resulting in a value for Q_{max} of 0.85. Figure 3(a) shows the result of simulating the A_{R-P} algorithm (21) for p = 0.5 and three different values of λ. The horizontal lines show the theoretical values for Q_∞, obtained from (35a), for (top to bottom) $\lambda = 0.01$ ($Q_\infty = 0.8485$), $\lambda = 0.25$ ($Q_\infty = 0.8152$), $\lambda = 0.5$ ($Q_\infty = 0.7868$). These asymptotic values are numerically identical to those obtained by Barto and Anandan [3], although the expressions involved in the derivation are necessarily quite different — our derivation is in fact simpler because we are able to work directly with the probabilities α_{1u}. The data show the average value of Q_n over 100 runs for $0 \leq n \leq 200$. The results of this simulation are qualitatively similar to those obtained by Barto and Anandan (compare with Figure 1(a) of [3]), for example in the way that the

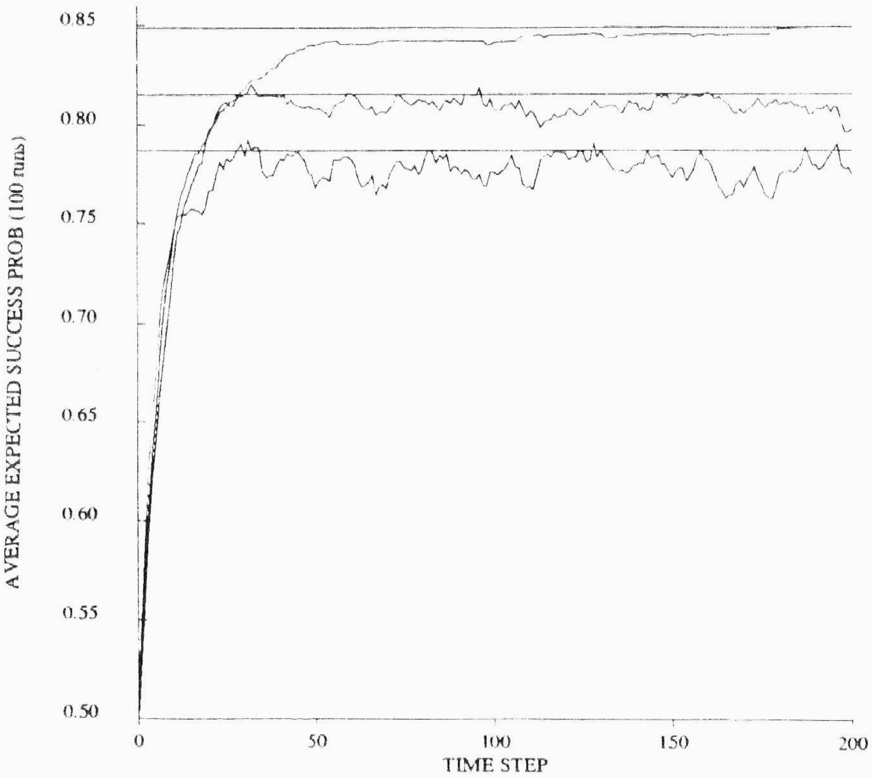


Fig. 3a

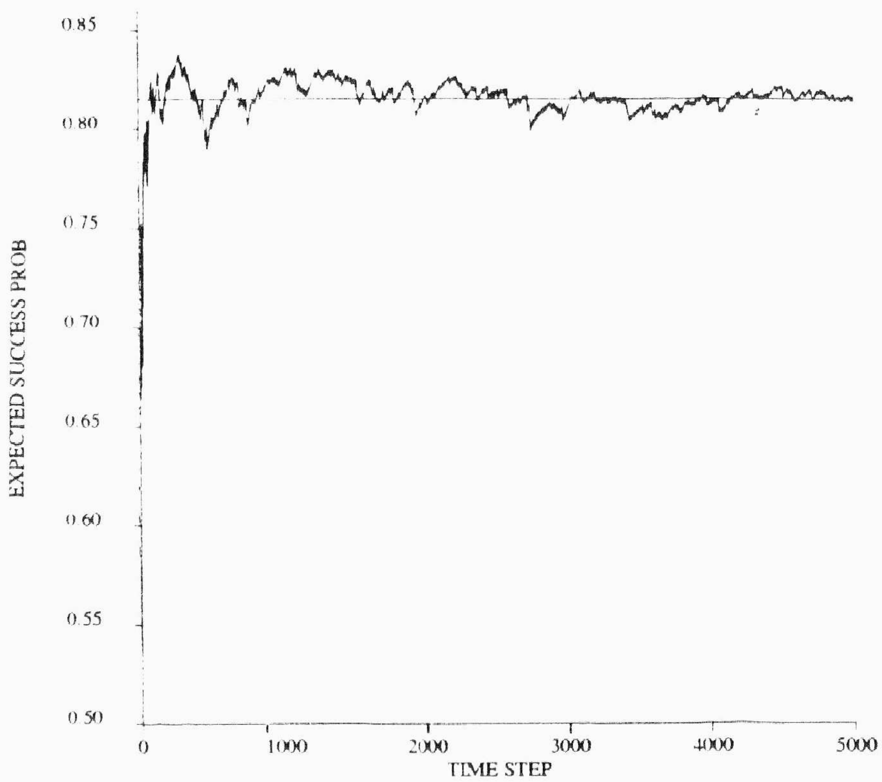


Fig. 3b

Figure 3

Simulation results for Task 1.

(a) Curves showing averages of Q_n over 100 runs of A_{R-P} algorithm for three values of λ and constant ρ.
(b) Curve showing Q_n for a single run of A_{R-P} algorithm with ρ_n decreasing with n.

learning rate slows as λ decreases. However the pRAM A_{R-P} algorithm is around 8—10 times faster than the Barto algorithm. Figure 3(b) is a plot of Q_n, $0 \leq n \leq 5000$, for a single run of the A_{R-P} algorithm with $\lambda = 0.25$, $\rho_n = 1/n^{0.55}$. This ρ-sequence satisfies the requirements of Theorem 3, so this simulation displays the property of strong convergence. The expected success probability after 5000 time steps is 0.8142; this should be compared with the theoretical asymptotic value 0.8152. The pRAM algorithm is again faster than that due to Barto (Figure 1(b) of [3]), but convergence to the asymptote Q_∞ is less smooth.

Task 2

This task is more difficult because the reward probabilities $\beta_{y,\underline{u}}, \beta_{\bar{y},\underline{u}}$ are closer in value:

$$\beta_{0,10} = 0.9,\ \beta_{1,10} = 0.6,\ \beta_{0,11} = 0.2\ \beta_{1,11} = 0.4$$

In this case optimal performance would be obtained from $\alpha_{10} = 0$, $\alpha_{11} = 1$ ($Q_{max} = 0.65$). Figure 4(a) shows the average of Q_n over 100 runs when $\rho_n = 0.5$ for all n, $\lambda = 0.05$. The horizontal line is the theoretical value $Q_\infty = 0.6348$. Figure 4(b) shows results obtained under the same conditions as for 4(a) except that ρ is held constant at 0.1. It can be seen that in this case the average of Q_n approaches the asymptote more closely, although training times are longer. Again, our results are qualitatively similar to those of Barto and Anandan (Figures 2(a), (b) of [3]), but differ in the speed of convergence — the pRAM algorithm is again around 8—10 times faster.

Comparison with the Barto A_{R-P} algorithm on the above classification tasks has indicated that the pRAM algorithm is significantly faster. This may be largely due to the high degree of non-linearity in our system. The role of noise is less clear; in general convergence in our model is less smooth, although we believe it is likely that the stochastic features are also responsible for improving the speed of convergence. It will be important to investigate the scaling behaviour of our model, in order to ensure that these advantages persist at the scale of more realistic problems, and such investigations are in progress.

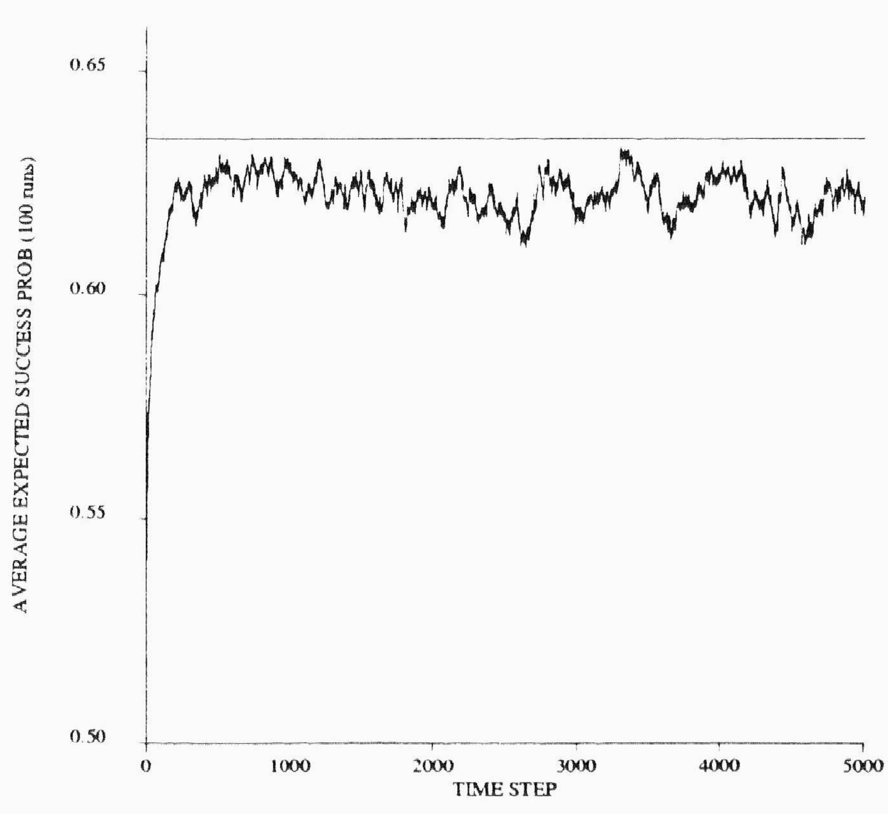


Fig. 4a

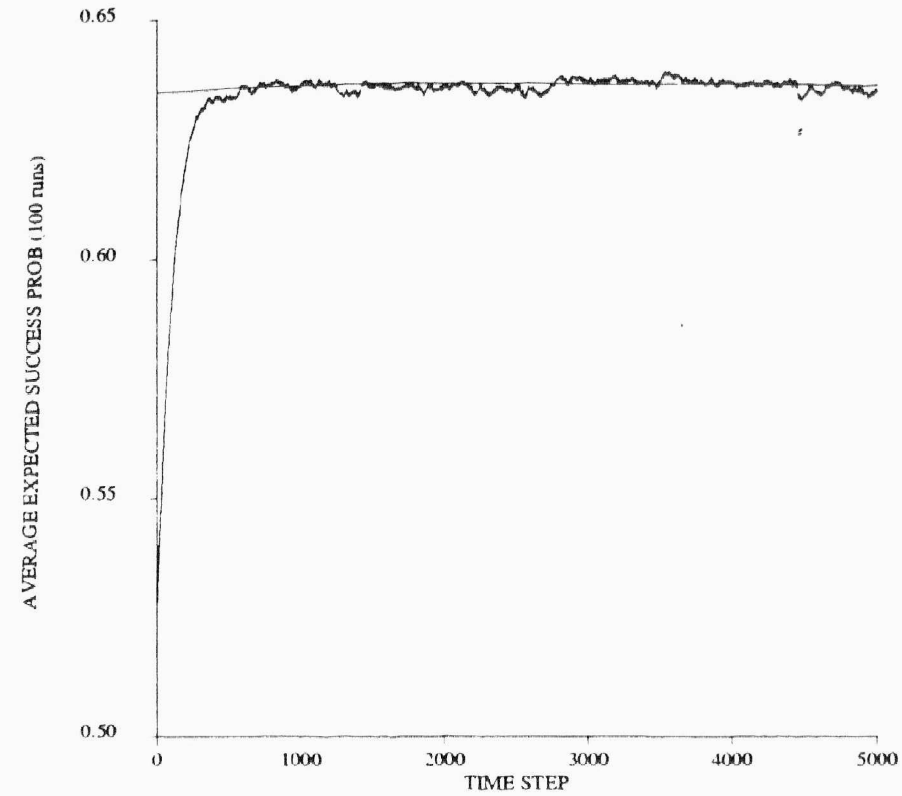


Fig. 4b

Figure 4

Simulation results for Task 2 averaged over 100 runs with constant ρ *where*

(a) $\rho = 0.5$
(b) $\rho = 0.1$

5. Discussions

In this paper we have presented a very general model of stochastic neural automata, in fact the most general such model within the binary domain. This model has been identified with the 'probabilistic random access memory' or pRAM [5]. A particular reinforcement training rule, extending the A_{R-P} rule of Barto [3] to our 'universal' learning automaton, has been investigated in some detail, and conditions under which the behaviour is ergodic or absorbing, and for which strong convergence may be expected, have been determined. Simulation work with our A_{R-P} algorithm has indicated that the maximal degree of nonlinearity and stochasticity associated with the pRAM model allows very effective learning to take place, both in the case of convergence in the mean and when the conditions for strong convergence are fulfilled.

There are several interesting questions on the theoretical side, for example with respect to the ε-optimality of the A_{R-P} training rule when the training rate ρ is held constant. Barto and Anandan [3] raised (but did not answer) this question in the context of their own A_{R-P} rule, having observed simulations which suggested that the constant-ρ system might indeed be ε-optimal. Work on this question is in progress.

In the A_{R-P} rule (21), the punishment signal was identified with the condition r = 0. In fact it is possible (and many in general be advisable) to have separate reward (r = 1) and punishment (p = 1) signals; this allows the possibility of 'neutral' actions which are neither rewarded nor punished, but which may correspond to a useful exploration of the environment. A significant extension of the pRAM model itself would allow it to realise mappings from $[0, 1]^N$ to {0,1} (this condition defines the *integrating pRAM* or *i- pRAM*). Simulation work has indicated that i-pRAMs are very effective in solving a variety of problems of adaptive control, using an appropriate generalisation of the learning rule (21).

The pRAM is a device which has a straightforward hardware implementation (a small pRAM net has already been built [14]). The A_{R-P} training algorithm described above may itself be implemented in hardware using either digital or analog technology, thus making possible the manufacture of self-contained 'learning pRAMs' (it would also be possible to realise the i-pRAM in hardware, together with its A_{R-P} learning rule). Networks of such units could find wide application, for example in the control of autonomous robots. Control need not be centralised; small nets of learning pRAMs could for example be located in the individual joints of a robot limb (such a control arrangement would be akin the neural ganglia found in insects).

One of the questions to be explored is the problem of scaling in pRAM networks, since in an N-input pRAM the number of memory locations $\underline{u}$ increases as 2^N. In some applications we may thus need to look for ways in which the number of locations could be restricted (other than by the simple expedient of putting strict bounds on N, as in the "pyramid neurons" made up of 2-input RAMs which have been proposed by Aleksander). We are investigating ways in which a dynamical list of the current 'most important' locations may be maintained and utilised within advanced pRAM architectures, such as are being developed by Clarkson [16]. The current rapid increase in size of RAM memory chips (for example 64 megabyte chips are now available) indicates that there may be an exponential increase in memory availablity to meet our requirements. In any case we note that scaling is unlikely to be a problem in many control applications, which typically require non-linear discriminations to be made but do not involve large numbers of processors; we consider such applications to be a very promising area for the use of pRAM technologies.

In conclusion, we believe that the identification of our pRAM model with the universal associative stochastic learning automaton described above leads to a learning system of very general utility which is capable of hardware implementation using conventional technologies. The system is trainable using a reinforcement algorithm (not requiring the intervention of a centralised learning controller, and thus suitable for 'on-line' applications), and capable of being extended in various useful ways. We hope both to develop further the mathematical analysis of the system and its extensions, and to apply it to more challenging problems of adaptive control.

References

[1] H Klopf, *The Hedonistic Neuron: A Theory of Memory, Learning and Intelligence*. Washington DC: Hemisphere, 1982.

[2] D E Rumelhart, J L. McClelland and the PDP Research Group, *Parallel Distributed Processing*, Vols 1 and 2, MIT Press, 1986.

[3] A G Barto and P Anandan, "Patern recognising stochastic learning automata", IEEE Trans. Syst., Man, Cyb., Vol SMC-15, 360-375 (1985).

[4] J G Taylor, "Spontaneous behaviour in neural networks", J. Theor. Biol. 36, 513-528 (1972).

[5] D Gorse and J G Taylor, "On the identity and properties of noisy neural and probabilistic RAM nets", Phys. Lett. A131, 326-332 (1988); D Gorse and J G Taylor, "An analysis of noisy RAM and neural nets", Physica D34, 90-114 (1989); D Gorse and J G Taylor, "A general model of stochastic neural processing", Biol. Cybern. 63, 299-306.

[6] P C Bressloff and J G Taylor, "Random interative networks", Phys. Rev. A41, 1126-1137 (1990).

[7] D J Amit, *Attractor Neural Networks*, CUP, 1989.

[8] I Aleksander, "The logic of connectionist system", in *Neural Computing Architectures*, ed I Aleksander, MIT Press, 1989, 133-155.

[9] K Narendra and M Thathachar, "Learning automata — a survey", IEEE Trns. Syst., Man, Cyb., Vol SMC-4, 323—334 (1974).

[10] M Norman, "Some convergence theorems for stochastic learning models with distance diminishing operators", J. Math. Psych. 5, 61-101 (1968).

[11] D Gorse and J G Taylor, "Training strategies for probabilistic RAMs", in *Parallel Processing in Neural Systems and Computers*, eds R Eckmiller, G Hartmann and G Hauske, North-Holland, 1990, 161-164.

[12] S Karlin, "Some random walks arising in learning models I", Pac. J. Math. 3, 725-775 (1953).

[13] P C Bressloff and J Stark, "Neural networks, learning automata and iterated function systems", in Proc. Brit. Comp. Soc. on 'Fractals and Chaos' (to appear).

[14] T G Clarkson, D Gorse and J G Taylor, "Hardware realisable models of neural processing", in Proc. of the First IEE International Conference on Artificial Neural Networks, 1989, 242-246.

[15] K Narendra and M A L Thathachar, *Learning Automata*, Prentice Hall, 1989.

[16] T G Clarkson, C K Ng, D Gorse and J G Taylor, "A serial-update VLSI architecture for the learning probabilistic RAM neuron", to appear in Proceedings of ICANN-91, Helsinki, June 24-28, 1991.

Literature Survey

The literature on neuroscience increases last few years extremely fast. At present some estimations of more than 20000 existing papers, conference and symposium talks, books and research reports are made. Evidently it is not possible to inform the readers about all the interesting publications, which currently appear. However, we would like to use the existence of the computer oriented Scientific Information System of the Institute of Computer and Information Science in Prague for to present here almost regularly the short survey of the last year records of this base.

Of course, the readers are asked for to be so kind and inform the Editors or the Institute about any publication, which they recommend to insert in this literature survey.

Aiyer S. V. B., Niranjan M., Fallside F.: A Theoretical Investigation into the Performance of the Hopfield Model, IEEE Transactions on Neural Networks, Vol. 1, 1990 No. 2, pp. 204-215

Abstract: This paper analyzes the behavior of the Hopfield model as a content addressable memory (CAM) and as a method of solving the traveling salesman problem (TSP).The analysis is based on the geometry of the subspace set up by the degenerate eigenvalues of the connection matrix. For the content addressable memory, it is shown that spurious fixed points can occur at any corner of the hypercube that is on, or near, the subspace spanned by the memory vectors. For the traveling salesman problem, analytic expressions are derived which make the network robust, so it can solve the traveling salesman problem with tour sizes of 50 cities or more.

Aleksander I.: Neural Computing Architectures, the Design of Brain-Like Machines MIT Press, Cambridge 1989, 401 p. ISBN 0-262-01110-7

Allen R. B. ,Alspector J.: Learning of Stable States in Stochastic Asymmetric Networks (Scanning the Issue), IEEE Transactions on Neural Networks Vol. 1, 1990 No. 2, pp.

233-238

Abstract: This paper investigates Boltzmann-based models with asymmetric connections. Although initially unstable, these networks spontaneously self-stabilize as a result of learning. Moreover, pairs of weights symmetrize during learning.However, the symmetry is not enough to account for the observed stability. To characterize the system it is useful to consider how its entropy is affected by learning and by the entropy of the information stream.

Almeida B. L., Wellekens C. L.: Neural Networks ((Lecture Notes in computer Science, vol.412) — Soft cover DM 42,-) Technical report: Berlin, Springer-Verlag, 1990 ISBN 3-540-52255-7

Abstract: The EURASIP workshop contributions collected in this volume have an interdisciplinary character. The authors include psychologists, biologists, engineers and mathematicians as well as computer scientists. The volume starts with two invited papers, by George Cybenko and by Eric Baum, on the formal study of the capabilities of neural networks. The papers are organized into parts dealing with theory and algorithms, speech processing, image processing, and implementation.

Antsaklis P. J.: Neural Networks for Control Systems, IEEE Transactions on Neural Networks, Vol. 1, 1990 No. 2, pp. 242-244

Abstract: This letter describes 11 papers from the April 1990 Special Issue of the IEEE Control Systems Magazine on Neural Networks in Control Systems, Vol. 10, No. 3, pp. 3-5 (ISSN0272-1708).

Atlan H.: Theories of Immune Networks Springer Series in Synergetics vol. 46, 1989, 117p. ISBN 3-540-51678-6

Abstract: Theories of Immune Networks presents various techniques to model the immune system as a network of interacting units (cell and molecule populations). Their respective significance is discussed on the basis of available empirical data (idiotypic-anti idiotypic interactions, receptor-antigen reactions and cross-reactivity) and of the simulation properties exhibited by the models (stability,memory, learning). Neural networks computation techniques, recently developed in cognitive sciences, are introduced into the field of immunology for the first time.

Baum E. B.: The Perceptron Algorithm is Fast for Nonmalicious Distributions, Neural Computation Vol. 2, 1990 No. 2 pp. 248-260

Abstract: The perceptron algorithm is shown to be a distribution-independent learning algorithm. In an appendix we show that, for uniform distributions, some classes of infinite V-C dimension including convex sets and a class of nested differences of convex sets are learnable.

Bienenstock E.: Relational Models in Natural and Artificial Vision In: Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds: Eckmiller Malsburg Ch.), 1989 pp.61-70

Bilbro G. L., White M.,Snyder W.: Image Segmentation with Neurocomputers In: Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds:Eckmiller R., Malsburg Ch.), 1989 pp. 71-80

Bout Van den D. E., Miller III., T.K.: Graph Partitioning Using Annealed Neural Networks, IEEE Transactions on Neural Networks Vol. 1, 1990 No. 2 pp. 192-203

Abstract: This paper presents a new algorithm, called mean field annealing (MFA), which is applied to the graph partitioning problem. The MFA algorithm combines characteristics of the simulated annealing algorithm and the Hopfield neural network. The rate of convergence of MFA on graphbipartitioning problems is ten to one hundred times that of simulated annealing (SA), with nearly equal quality of solutions. The temperature behavior of MFA during graph partitioning is shown to possess a critical temperature at which most of the optimization occurs.

Card H. C., Moore W. R.: Silicon Models of Associative Learning in Aplysia, Neural Networks Vol. 3, 1990 No. 3, pp. 333-346

Key words: analog VLSI models of neural networks; EE-

PROMS; associative learning; invertebrate models; non-Hebbian synapses.

Abstract: We describe analog CMOS circuits which can be used to qualitatively mimic the mechanisms of associative learning in this creature. In particular we model the synaptic plasticity and activity-dependent facilitation in sensorimotorsynapses responsible for classical conditioning of the gill with draw all reflex. The same circuits implement then on associative learning mechanisms of habituation and sensitization. We also discuss considerations in abstracting from the biological mechanisms when developing artificial neural systems. The specific circuits for Aplysia described in this paper are suggestive of way to implement VLSI area-efficient associative learning circuits in general, especially those which perform in situ temporal associations on multiple synaptic inputs.

Cybenko G.: Mathematical Problems in Neural Computing Technical report: CSRD-905 Washington, University of Illinois at Urbana-Champaign, 1989, 18 p.

Key words: applications; networks.

Abstract: Focusing on signal processing applications in pattern recognition and classification, we present some analytic and experimental results that indicate promising directions for future research. At the same time, these results show that in a well defined sense, a certain class of neural computing techniques are universally applicable if feasibility constraints are not an issue.

Devos M., Orban G. A.: Modeling Orientation Discrimination at Multiple Reference Orientations with a Neural Network, Neural Computation, Vol. 2, 1990, No. 2 pp. 152-161

Abstract: We trained a multilayer perceptron with backpropagation to perform stimulus orientation discrimination at multiple references using biologically plausible values as input and output. Hidden units are necessary for good performance only when the network must operate at multiple reference orientations. The orientation tuning curves of the hidden units change with reference. Our results suggest that at least for simple parameter discriminations such as orientation discrimination, one of the main functions of further processing in the visual system beyond striate cortex is to combine signals representing stimulus and reference.

Dontas K., Sarma J., Srinivasan P., Wechsler H.: Fault Tolerant Hashing and Information Retrieval Using Back Propagation, IEEE, 1990 pp. 345-351

Abstract: The paper describes the architecture and performance of neural networks designed and trained to compute hashing functions.The networks described are of the connectionist type and are capable of learning complex mappings using the back propagation algorithm.

Eberhart R. C.: Standardization of Neural Network Terminology, IEEE Transactions on Neural Networks Vol. 1, 1990 No. 2 pp. 244-245

Abstract: It is desirable to move toward commonly accepted terminology in the neural network field.This letter outline the initial activities of an Ad Hoc Standards Committee established by the IEEE Neural Networks Council to pursue this effort.

Ersoy O. K., Hong D.: Parallel, Self-Organizing, Hierarchical Neural Networks (Scanning the Issue), IEEE Transactions on Neural Networks, Vol. 1, 1990 No. 2, pp. 167-178

Abstract: This paper presents a new neural network architecture called the parallel self-organizing hierarchical neural network (PSHNN) which involves a number of stages. At the end of each stage, error detection is carried out, and a number of input vectors are rejected. Between two stages there is a nonlinear transformation of those input vectors rejected by the previous stage. The new architecture has desirable properties such as high classification accuracy, minimized learning and recall times, and truly parallel architectures. Experiments show advantages in comparison to multilayered networks with back-propagation training.

Fanelli R., Raphan T., Schnabolk Ch.: Neural Network Modelling of Eye Compensation During Off-Vertical-Axis Rotation, Neural Networks, Vol. 3, 1990, No. 3 pp. 265-276

Key words: oculomotor system; vestibular system; back propagation; modelling; otoliths; nystagmus.

Abstract: Compensatory eye motion during off-vertical axis rotation of the head in darkness (OVAR), has been modelled with a neural network. The three layered network was trained with back-propagation to simulate the estimation of head velocity during OVAR. The network produced good estimates within its training range and predicted the eye velocity versus head velocity characteristics in the monkey. Invariance of the compensation to changes in tilt angle, not fully addressed in previous models, was demonstrated by the network, along with as booth decline in velocity estimate below a threshold angle.

Farrell J.A., Michel A.N.: A Synthesis Procedure for Hopfield's Continuous-Time Associative Memory, IEEE Transactions on Circuits and Systems, Vol. 37, 1990 No. 7 pp. 877—884

Abstract: The present paper presents a new technique for designing associative memories to be implemented by Hopfield neural networks. This technique guarantees that each desired memory is stored and is attractive. The procedure also guarantees that the resulting network is implementable, a requirement often overlooked by other methods.

Foo S.Y., Anderson L.R., Takefuji Y.: Analog Components for the VLSI of Neural Networks, IEEE — Circuits & Devices Vol.6, 1990 No.304 pp.18-26

Abstract: Artificial neural networks can be implemented with simple analog devices.

Frean M.: The Upstart Algorithm: A Method for Constructing and Training Feedforward Neural Networks, Neural Computation, Vol.2, 1990 No.2 pp.198-209

Abstract: A general method for building and training multilayer perceptrons composed of linear threshold units is proposed. A simple recursive rule is used to build the structure of the network by adding units as they are needed, while a modified perceptron algorithm is used to learn the connection strengths. Convergence to zero errors is guaranteed for any boolean classification on patterns of binary variables. Simulations suggest that this method is efficient in terms of the numbers of units constructed, and the networks it builds can generalize over patterns not in the training set.

TRAINING DISCRETE-TIME FEEDBACK NETWORKS FOR FILTERING AND CONTROL

L. Personnaz, O. Nerrand, P. Roussel-Ragot, G. Dreyfus)*

Introduction

During the 1960's, the most popular learning rules were essentially based on the minimization of quadratic criteria by gradient descent. Among those methods, the Widrow-Hoff rule [1], is widely used in the field of linear signal processing and adaptive control, without reference to neural networks. One of the reasons of the revival of the field was the introduction of a new gradient descent technique, known as backpropagtion [2], which has been widely applied to neural networks having a specific architecture: multi-layer feedforward networks.

In the present paper, we propose a general framework for training discrete-time neural networks by gradient descent. The techniques that we suggest are applicable to any network architecture, including feedback networks, whose behaviour depends on time explicitly. This systematic, explicit introduction of time is important since it casts neural networks in the framework of non-linear recursive filters, which suggests potential applications in signal processing and adaptive control.

In the first section, we shall deal with feedforward networks, in which the outputs of the neurons depend on the inputs at the same instant of time, the necessary delay for information transmission across the layers being negligibly small: these networks simply perform input-output mappings irrespective of the past history of the network. We shall introduce a gradient descent training method termed "forward computation", and we shall describe the advantages and shortcomings of this method as compared to those of backpropagation.

In the second section of the paper, we introduce a general description of feedback neural networks, together with a general methodology for training any network, regardless of its architecture.

We shall distinguish between two kinds of problems:

- "non-adaptive" problems, for which the network is first trained, and subsequently used; this is the case in most neural networks studied to date;
- "adaptive" problems, for which the neural network is trained permanently while it is used.

Feedforward (Static) Networks

At present, a large number of investigations of neural networks deal with the classification of static, non-ordered patterns, with the prediction of time series (ordered patterns), etc. These studies use feedforward networks: in this framework, one considers neurons whose outputs depend only on their inputs at the same instant of time, neglecting the time necessary to compute the outputs as a function of the inputs. Therefore, information flow can take place in one direction only, from the inputs to the outputs.

1. Definitions and notations:

A static neuron is defined as: $x_i = f_i(v_i)$ with $v_i = \sum_{j \in P_i} C_{ij} x_j$

where v_i is its *potential*,
$f_i(.)$ is its *activation function*,
and x_i is the *output of neuron* i.

Tne *inputs of neuron* i are the neurons, or the network inputs, which send their values x_j to neuron i. P_i stands for the set of the indices of the inputs of neuron i, and C_{ij} is the *synaptic weight* or the *coefficient* of the input j of neuron i.

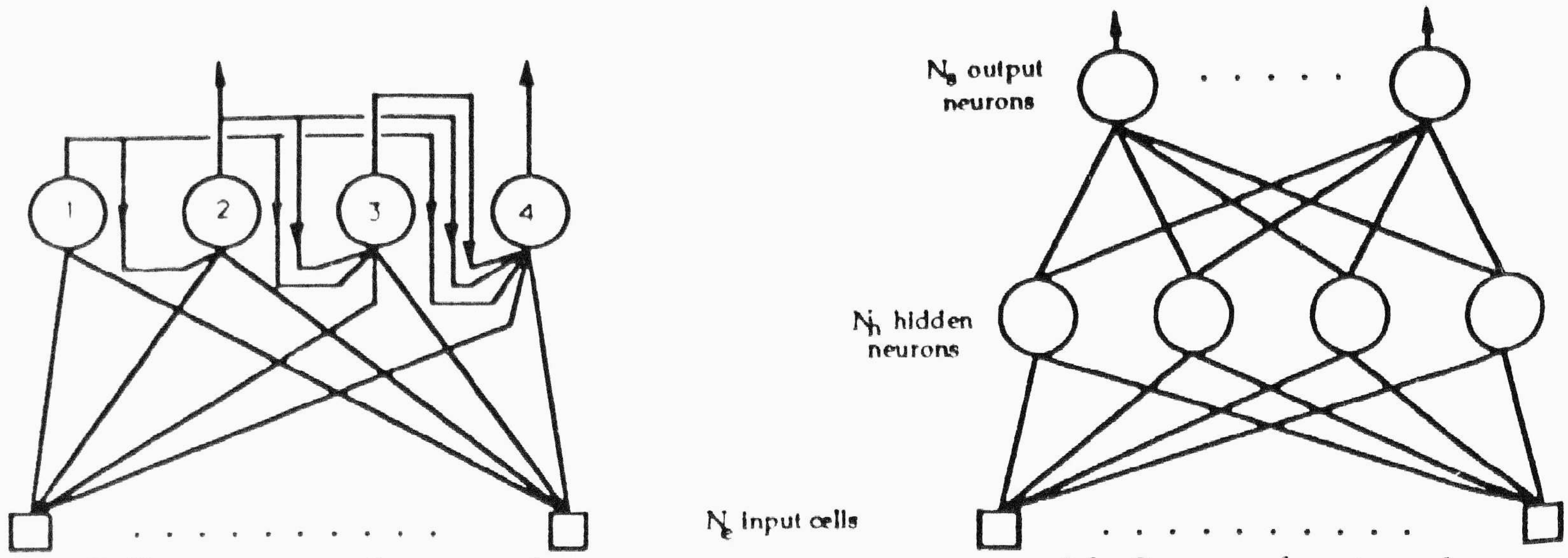
Any differentiable activation function can be used. One often uses the sigmoidal transfer function:

$$f(u) = \frac{2}{1 + e^{2u}} - 1 = \tanh(u).$$

The most general architecture for a static network is that of a *fully connected feedforward network* (Figure 1.a). In such a structure, the neurons can be numbered in a definite order: the first neuron is driven by the inputs of the network only, the second neuron is driven by the network inputs and by the first neuron, . . ., the last neuron is driven by the network inputs and by all other neurons ($C_{ij} \neq 0$ i > j).

Output neurons are the neurons whose output are taken into account in the criterion which is minimized

*) L. PERSONNAZ, O. NERRAND, P. ROUSSEL-RAGOT, G. DREYFUS
Ecole Supérieure de Physique et de Chimie Industrielles de la Ville de Paris
Laboratoire d'Electronique, 10 rue Vauquelin
F-75005 PARIS — FRANCE
S. MARCOS, C. VIGNAT
Laboratoire des Signaux et Systèmes, CNRS-ESE
Plateau du Moulon
F-91192 GIF SUR YVETTE Cedex — FRANCE



1.a. Fully connected network.
Neurons 2 and 4 are output neurons.
Neurons 1 and 3 are hidden neurons.

1.b. Layered network.

Examples of feedforward networks.

Fig. 1

during training (which will be defined in the next section); all other neurons are *hidden neurons.*

Layered networks (Figure 1.b) are a special case of fully connected feedforward networks.

In the following, we consider the general case of a fully connected network.

N_e stands for the number of input units,
N_n stands for the number of neurons,
N_h stands for the number of hidden neurons,
and N_s stands for the number of output neurons ($N_n = N_h + N_s$).

As mentioned above, the neurons can be ordered:

— the N_e network inputs are numbered from $i = 1$ to N_e; we denote the set of their indices by E;
— the N_n neurons are numbered from $i = N_e + 1$ to $N_e + N_n$; we denote the set of their indices by N;

Finally, we denote by H the set of indices of the N_h hidden neurons, and by S the set of the indices of the N_s output neurons.

2. Training:

Supervised training consists in computing the values of the coefficients $\{C_{ij}\}$ so that the network performs the desired task (classification, prediction, etc.). This is done by an iterative procedure whereby each modification of the coefficients is performed according to

$$\Delta C_{ij} = -\mu \frac{\partial J}{\partial C_{ij}}$$

where J is a suitable cost function of the form

$$J(\underline{C}) = \sum_{k \in L} Jk(\underline{C}) = \frac{1}{2} \sum_{k \in L} \sum_{r \in S} (d_r^k - x_r^k)^2,$$

where L is collection of pairs of vectors $\{\underline{\xi}^k, \underline{\mathbf{d}}^k\}$; $\underline{\xi}^k$ is an example vector, with N_e components x_i^k (i belongs to set E); $\underline{\mathbf{d}}^k$ is a vector of desired outputs, with N_s components d_i^k (i belongs to set S).

Typically, in classification, $\underline{\xi}^k$ may be a representation of a pattern (picture, phoneme, ...) and $\underline{\mathbf{d}}^k$ may be the code of the corresponding class, provided by a supervisor; in filtering, $\underline{\xi}^k$ may be a temporal sequence of values of a signal s(t): $\{x_1^k = s(t - N_e + 1), x_2^k = s(t - N_e + 2), \ldots, x_{Ne}^k = s(t)\}$; in the case of a predictor for instance, $\underline{\mathbf{d}}^k$ may be the value of the signal at time $t + 1$: $\underline{\mathbf{d}}^k = s(t + 1)$.

In a classification task, or in non-adaptive transversal filtering, the ordering of the examples is arbitrary, and their number is finite, so that $J(\underline{C})$ may be the criterion which must be minimized after completion of the training phase; in the latter case, L is the whole training set. *In adaptive filtering,* the ordering of the examples is imposed by the temporal nature of the input signal, and their number is infinite, so that L cannot possibly be the whole training set.

Thus, during training, the computations necessary for *one modification of the coefficients* are performed as follows:

— ***propagation:*** during a given step, the weights computed during the last step are available; examples $\{(\underline{\xi}^k, \underline{\mathbf{d}}^k)$, for all $k \in L\}$ are presented to the network, which computes the values of v_i^k and x_i^k for all $i \in N$.
— ***gradient computation:*** $\nabla_{\underline{C}}(J) = \partial J/\partial \underline{C}$ (see below).
— ***modification of the weights:*** the modification of the weights C_{ij} is performed according to:

$$\Delta C_{ij} = \sum_{k \in L} \Delta C_{ij}^k = -\mu \sum_{k \in L} \frac{\partial J^k}{\partial C_{ij}} \text{ for } i \in N \text{ and } j \in E \cup N \text{ with } j < i.$$

The gradient $\partial J^k/\partial \underline{C}$ can be computed in two different ways:

a) **Gradient estimation by forward computation:**

The gradient of J^k is given by (omitting subscript k):

$$\frac{\partial J}{\partial C_{ij}} = -\sum_{r \in S} (d_r - x_r) \frac{\partial x_r}{\partial C_{ij}} \text{ for } i \in N$$

and $j \in E \cup N$ with $j < i$.

The partial derivatives on the right-hand side are obtained by the following relations:

$$\text{if } i = r \text{ then } \frac{\partial x_r}{\partial C_{rj}} = f_r'(v_r)\, x_j,$$

$$\text{if } i \neq r \text{ then } \frac{\partial x_r}{\partial C_{ij}} = f_r'(v_r) \sum_{m \in P_r} C_{rm} \frac{\partial x_m}{\partial C_{ij}}.$$

Thus, we get recursively:

$$\frac{\partial x_m}{\partial C_{mj}} = f_m'(v_m)\, x_j \text{ for all } m \in N,$$

$$\frac{\partial x_m}{\partial C_{ij}} = f_m'(v_m) \sum_{h \in P_m} C_{mh} \frac{\partial x_h}{\partial C_{ij}} \text{ for all } m \in N,\ i \neq m,$$

with $\frac{\partial x_m}{\partial C_{ij}} = 0$ for all $i > m$, and $\frac{\partial x_m}{\partial C_{ij}} \neq 0$ for all $m \in E$.

P_m stands for the set of the indices of the inputs of neuron m. Thus, the above relations show clearly that the computation must be performed in the forward direction, i.e. from the inputs of the network to the last neuron.

b) **Gradient estimation by backpropagation** [2]:

The backpropagation technique consists in using the partial derivatives of the criterion with respect to the potentials as intermediate variables:

$$\frac{\partial J}{\partial C_{ij}} = \frac{\partial J}{\partial v_i} \frac{\partial v_i}{\partial C_{ij}} = \frac{\partial J}{\partial v_i} x_j \text{ for } i \in N \text{ and } j \in E \cup N$$

with $j < i$.

Defining: $y_i = -\frac{\partial J}{\partial v_i}$, one obtains

$$y_m = f_m'(v_m)\left[(d_m - x_m) + \sum_{h \in R_m} C_{hm}\, y_h\right] \text{ for } m \in S,$$

$$y_m = f_m'(v_m) \sum_{h \in R_m} C_{hm}\, y_h \text{ for } m \in H.$$

R_m stands for the set of the indices of the neurons to which neuron m transmits its output: $R_m = \{h \mid h > m\}$. This set is empty for the last neuron. In this case, computations are performed from the outputs to the inputs (backpropagation).

3. Computational complexity:

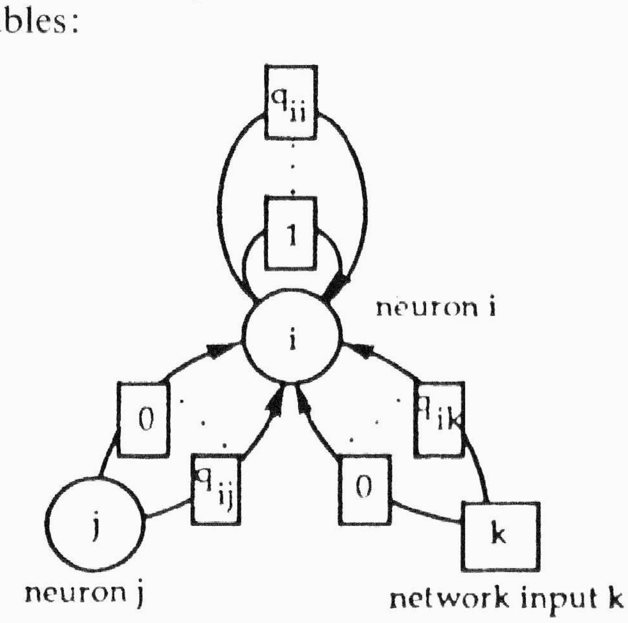
The number of multiplications required for a modification of all the weights of a fully connected network scales like N_n^4 in the case or forward computation and like N_n^2 with backpropagation.

Thus, for a given network and a given criterion (hence for the same modifications of the weights), training by backpropagation is less expensive than by forward computation.

However, training by forward computation gives a measure of the sensitivity of the outputs of the neurons to the modifications of the weights, whereas backpropagation gives the sensitivity of the criterion to the variations of the potentials of the neurons only. The analysis of the values of the sensitivities may lead to rational alterations to the architecture of the network.

Feedback (Dynamical) Networks

We now come to the networks which keep a memory part of their past behaviour : the evolution of the neurons is driven by difference equations (discrete-ti-



Discrete-time neuron i in a network; synaptic delays are shown within rectangles

Fig. 2

me networks) or by differential equations (continuous-time networks). In the following, we present a general, rigorous framework for training discrete-time feedback networks with arbitrary architectures.

1. Definitions and notations:

Model of discrete-time neuron:

Its equation is defined by the activation function $f_i(.)$ and its synaptic weight $\{C_{ij,\tau}\}$, where τ is the (discrete) delay of the synapse ij,τ transferring information from neuron j to neuron i:

$$x_i(t) = f_i\left[\sum_{j \in P_i} \sum_{\tau = 0}^{q_{ij}} C_{ij,\tau} x_j(t - \tau)\right]$$

where x_j can be either the output of neuron j or the value of an external input j.

It should be clear that several synapses can transfer information from neuron (or network input) j to neuron i, each synapse having its own delay τ and its own weight $C_{ij,\tau}$ (Figure 2).

Clearly, one must have $C_{ii,0} = 0\ \forall i$, otherwise the output of neuron i cannot be computed.

If neuron i is such that: $i \notin P_i$ and $q_{ij} = 0\ \forall j \in P_i$, neuron i is static.

If $\tau = 1$ for all synapses, the neurons are of first order. This specific type of network has been investigated by other authors [3, 4, 5].

Definitions of the state variables and of the order of the network:

The computation of the *outputs of the network* requires the knowledge of the values of a minimal number of *neuron outputs*, in addition to the knowledge of the values of the *network inputs*. The required values of the neuron outputs are the *state variables* of the network, and their number is the *order of the network*. The *state of the network* is the set of values of the state variables.

Any network can be represented as a graph whose nodes are the neurons and whose edges are the connections between neurons, weighted by the synaptic delays. The order of the network is the weight of the cycle of maximum weight in the graph.

The canonical representation of a discrete-time feedback network:

Any discrete-time feedback network can be cast into a canonical form made of a feedforward (static) network

— whose outputs are the outputs of the network and the outputs of the state neurons,
— whose inputs are the inputs of the network and the outputs of the state neurons, *the latter being delayed by one time unit.*

Example: a third-order feedback network:

Figure 3.a. shows a third-order network. Its equations are:

$$x_2(t + 1) = f_2[C_{22,1}x_2(t) + C_{21,0}x_1(t + 1) + C_{23,0}\, x_3(t + 1) + C_{23,1}\, x_3(t)]$$

$$x_3(t + 1) = f_3[C_{33,1}x_3(t) + C_{31,0}x_1(t + 1) + C_{32,2}\, x_2(t - 1)].$$

The output of neuron 2 at times $t - 1$ and t, together with the output of neuron 3 at time t, can be chosen as state variables.

Thus, the following variables can be defined:

$z_1(t) = x_1(t + 1)$; $z_2(t) = x_2(t)$; $z_3(t) = x_2(t - 1)$; $z_4(t) = x_3(t)$.

The state variables are $z_2(t)$, $z_3(t)$, $z_4(t)$.

The corresponding canonical form is shown on Figure 3.b.

A feedback network is completely defined by its associated static network, its state variables, its outputs, and its external inputs. The general canonical form of a feedback network is shown on Figure 4. A feedback network may have several canonical forms.

2. Training:

Just as in the case of static networks, training consists in computing the weights so that the network performs the desired task (filtering, prediction, control, etc.). In the present case, for each modification of the coefficients, the training algorithm estimates the gradient of the following function, on a temporal horizon T:

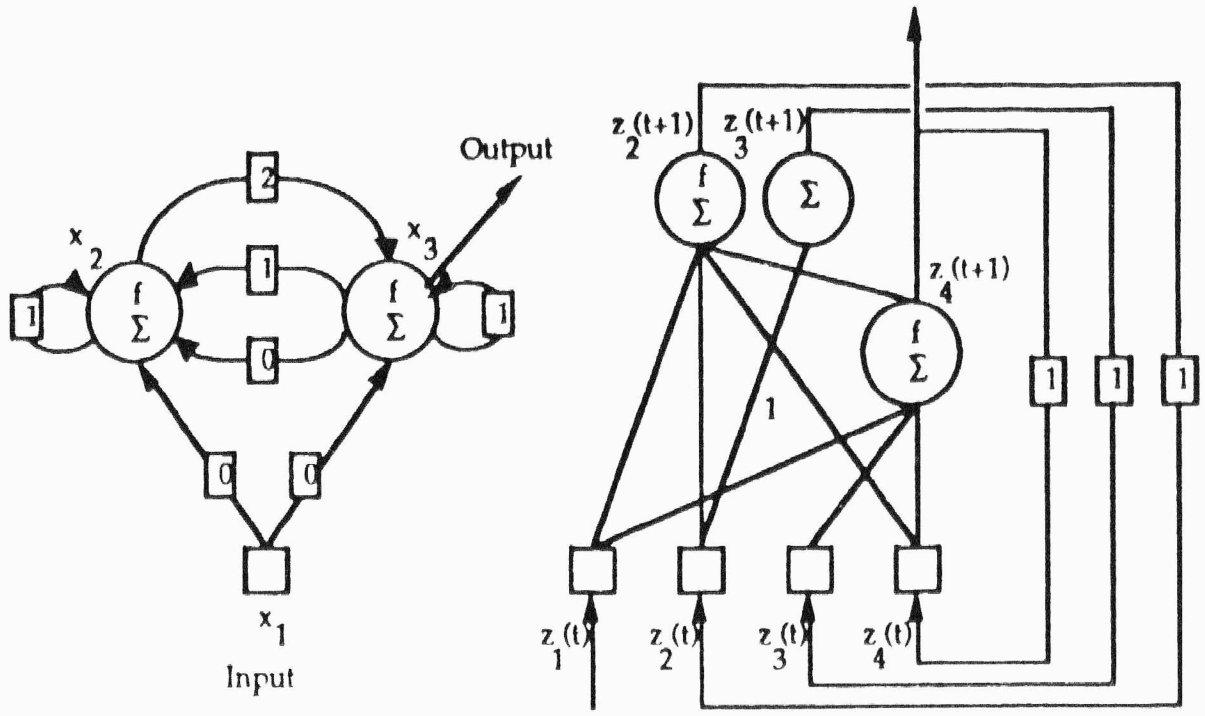
$$J_T(\underline{C}) = \sum_{t \in T} J(t) \text{ where } J(t) = \frac{1}{2} \sum_{r \in D_t} [d_r(t) - x_r(t)]^2,$$

where D_t is set of neuron outputs for which there exists a desired value $d_r(t)$ at time t (D_t is included in S), and where $x_r(t)$ is the corresponding output. Various cases of interest may arise: in problems of the type investigated in [10], the temporal horizon and the initial conditions are defined by the task itself (i.e., the number of time steps in the trajectory), and $d_r(t)$ exists at the last time step of the temporal horizon only; in typical adaptive filtering tasks, the desired values are taken into account on a sliding window, and both the temporal horizon and the initial conditions are parameters of the training algorithm.

To summarize, the problem of training a feedback network is defined by the following elements:

— a function, whose gradient is estimated on a temporal horizon T, from a set of desired values;
— the inputs at all times on that horizon;
— the values of the state variables at the beginning of the horizon.

Just as in static networks, the modification of the



3.a. 3.b.

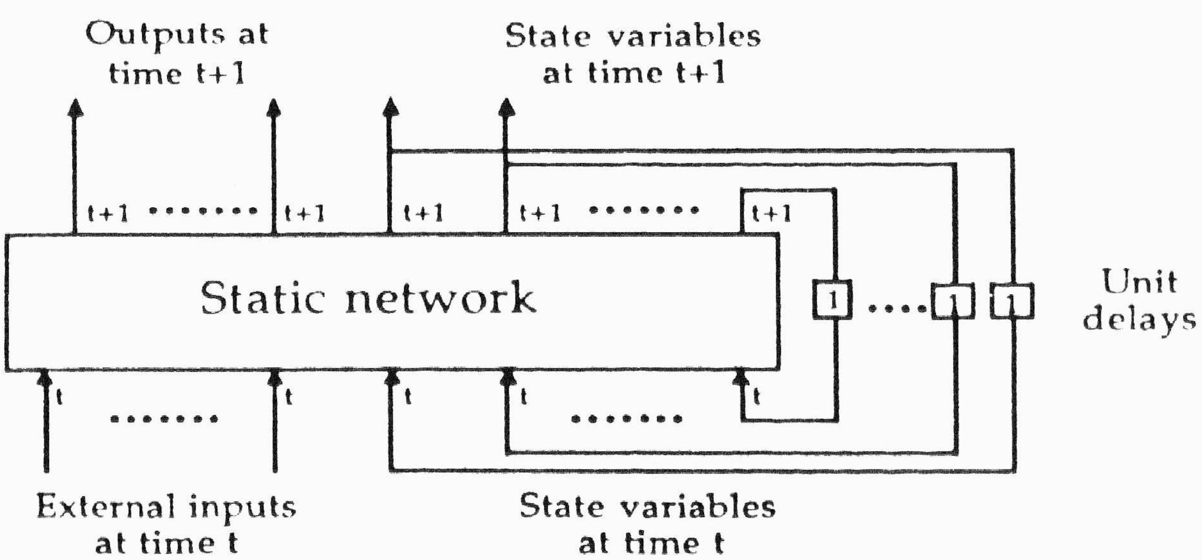
Symbol $\left(\begin{smallmatrix} f \\ \Sigma \end{smallmatrix}\right)$ denotes a neuron which computes f(v).

Symbol (Σ) denotes a "neuron" which performs a weighted sum only (here, the neuron which computes $z(t_3+1)$ is just the identity operator).

Delays are in boxes.

Example: a third-order feedback network.

Fig. 3



General canonical form of a feedback network.

Fig. 4

coefficients consists in modifying iteratively the weights $\{C_{ij,\tau}\}$ in the direction opposite to that of the gradient:

$$\Delta C_{ij,\tau} = -\mu \frac{\partial J_T(C)}{\partial C_{ij,\tau}} = -\mu \sum_{t \in T} \frac{\partial J(t)}{\partial C_{ij,\tau}} \text{ for } i \in N,$$

$$j \in E \cup N \text{ and } \tau \cup [0, q_{ij}].$$

Training proceeds as follows:

— ***initialization:*** at the initial time t_0 of horizon T, the state variables, which are necessary for computing the values of the outputs at time $t_0 + 1$ and beyond, are initialized. For the network shown on Figure 3, one must know the values of $x_2(t_0)$, $x_2(t_0 - 1)$ and $x_3(t_0)$ in order to be able to compute $x_3(t_0 + 1)$ and $x_2(t_0 + 1)$.

— ***propagation:*** the network is presented with the values of the external inputs, which allows the computation of the values of $v_i(t)$ and $x_i(t)$ for all neurons $i \in N$ and all times $t \in T$.
— ***gradient estimation:*** once the outputs are known, the gradient is estimated (see below).
— ***weight modifications:*** the modifications of the weights $\{C_{ij,\tau}\}$ are given by the above relation.

a) **Gradient estimation by forward computation:**

The gradient to be estimated is given by:

$$\frac{\partial J(t)}{\partial C_{ij,\tau}} = \sum_{r \in D_t} (d_r(t) - x_r(t)) \frac{\partial x_r(t)}{\partial C_{ij,\tau}} \text{ for } i \in N,$$

$$j \in E \cup N \text{ and } \tau \in [0, q_{ij}].$$

It can be shown that one has:

$$\frac{\partial x_m(t)}{\partial C_{ij,\tau}} = f'_m[v_m(t)] \cdot$$

$$\left[\delta_{im}\, x_j(t-\tau) + \sum_{h \in P_m} \sum_{\xi=0}^{q_{mh}} C_{mh,\xi} \frac{\partial x_h(t-\xi)}{\partial C_{ij,\tau}} \right]$$

for all $m \in N$,

with $\frac{\partial x_m(t)}{\partial C_{ij,\tau}} = 0$ for all $m \in E$, for all t.

It should be noted that the computations are performed in the forward direction (from the inputs to the outputs of the network), from the beginning to the end of the temporal horizon. A similar approach has been proposed [2, 5], restricted to the case where all synapses have unit delays.

b) **Computation by backpropagation:**

First,the network is put into a canonical form. In general, the resulting static network has additional "linear neurons" (Figure 3.b); thus, neurons are renumbered as described above for feedforward networks. Since all synapses of the static network have zero delays ($\tau = 0$), we denote their weights by C_{ij}.

Thus, the successive outputs of the feedback network can be computed by developing the operation of the network with a number of *copies* which is equal to the length of the horizon (Figure 5). The copy related to time t computes the state at time t + 1 from the values of the inputs and of the state variables at time t. The set of copies builds up a static network on which the gradient is estimated by backpropagation. Figure 6 shows an example in which the chosen horizon is of length 4, and in which the function J_T takes into account the desired values of the output at the last two instants of the horizon.

Depth of a network.

During training of a feedback network by backpropagation, several copies are necessary, in general, to compute the modifications of all the weights of the network. This minimal number of copies is termed the *depth* of the network. In the previous example, the depth is equal to 3. If the length of the chosen horizon is smaller than the depth, some weights are not affected by training.

Comparison between the two gradient methods.

The above methods compute the same gradient, hence yield identical coefficient modifications for the same initial conditions. Backpropagation is less computationally expensive, but it is less flexible with re-

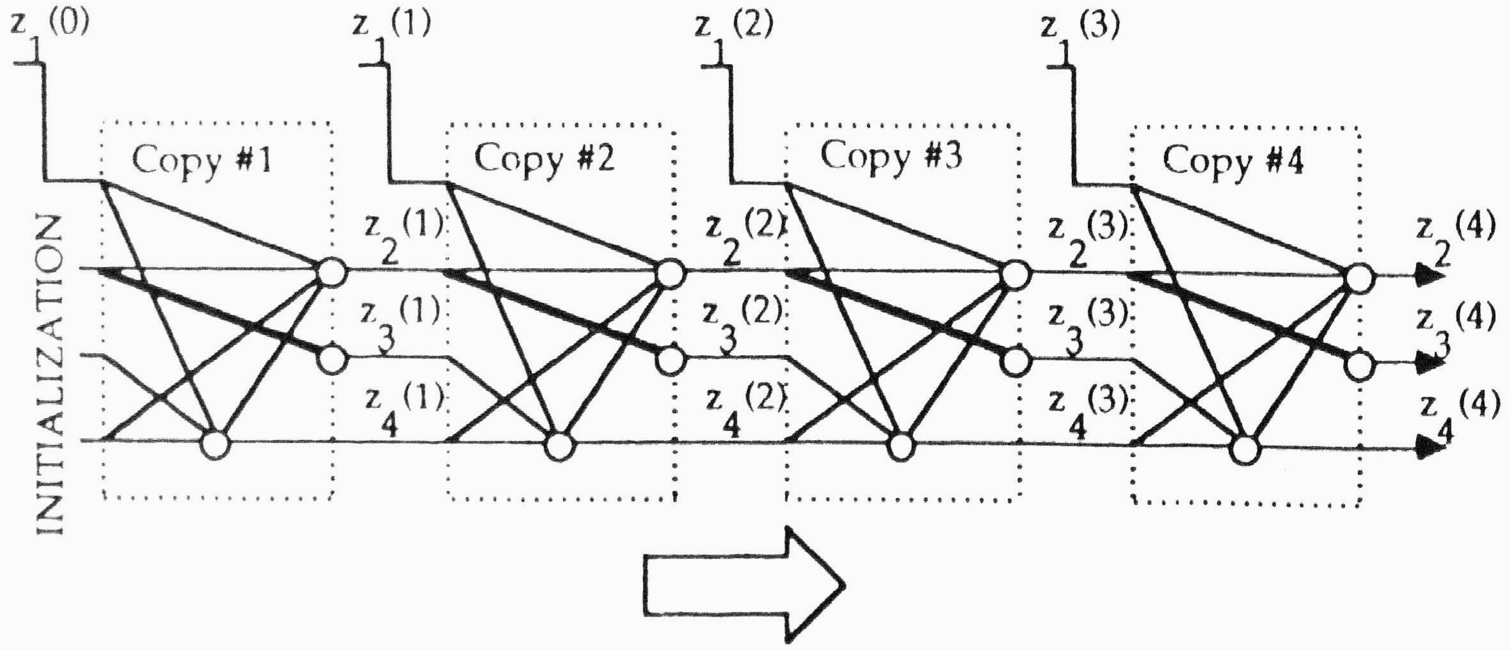
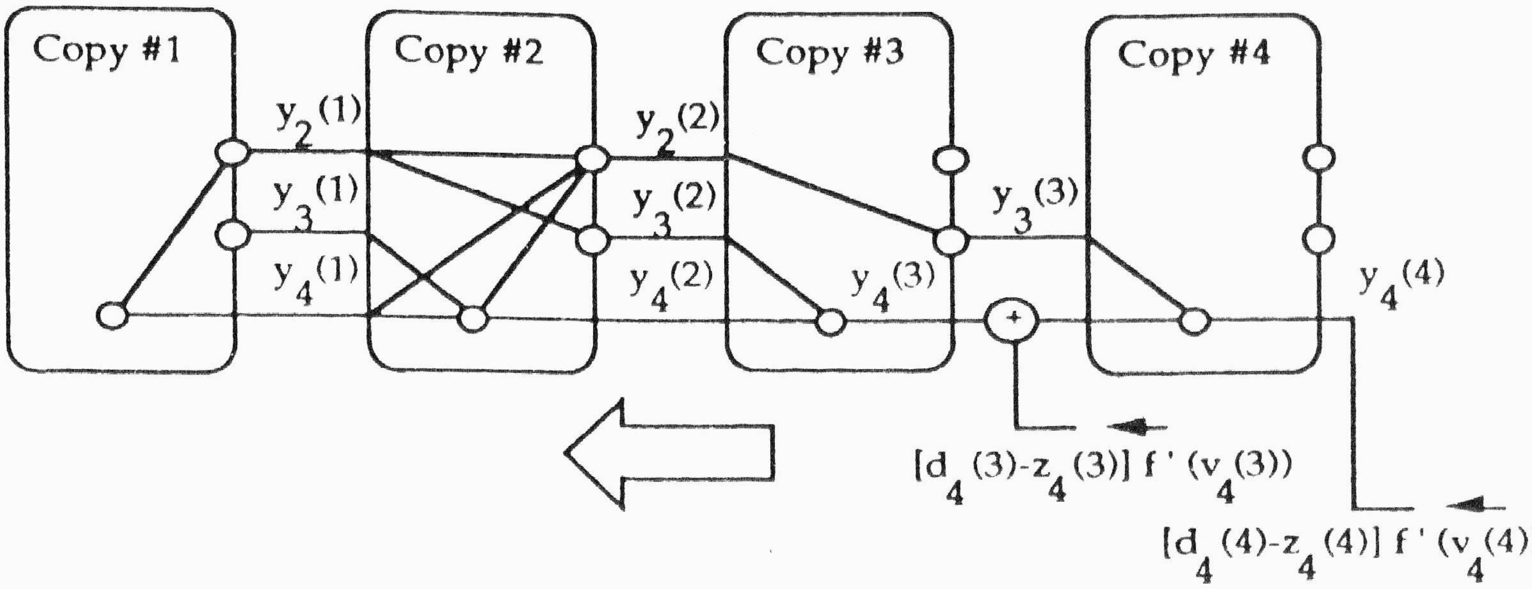
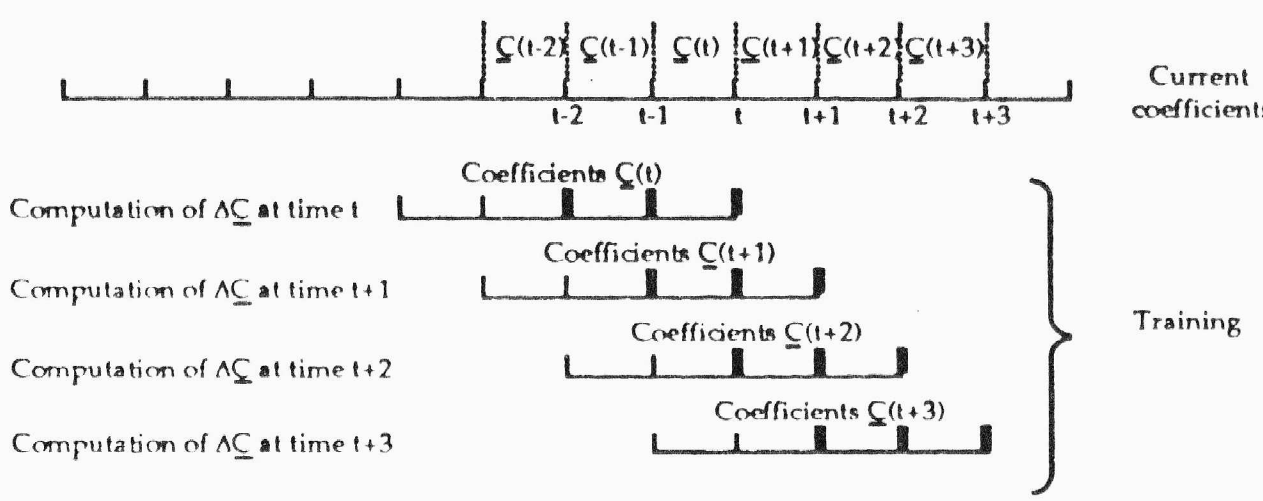


Fig. 5



Backpropagation of two desired values.

Fig. 6



Permanent training of an adaptive system.

Fig. 7

spect to the initialization of the derivatives of the feedback inputs [6].

Adaptiv *vs.* non-adaptive networks:

Feedback networks have been investigated mainly as non-linear dynamical systems operating as associative memories [3, 4, 7, 8]; it is only recently that they have been used for attacking problems in adaptive filtering and in adaptive control [6, 9, 12]. In this framework, two types of networks must be considered: (i)*non-adaptive* networks, whose weights are computed during a training phase, and are subsequently used, and (ii) *adaptive* networks which are trained permanently.

In an adaptive system, the coefficients of the network are updated periodically with period T_a. Between t and t + T_a, modifications of the weights are computed, possibly by performing several iterations of the gradient descent.

Figure 7 illustrates such an adaptive operation. Training is performed on a given horizon (of length four in the example), with a given set of desired values (in this example, desired outputs are defined on the last three instants of the horizon). The updating period is equal to 1.

Conclusion

In the present paper, we have described a general, rigorous framework for training any discrete-time network. Within this framework, a variety of training algorithms can be derived; some of them are described in more detail in [6, 11]. This approach may open new perspectives for the use of neural networks in adaptive signal processing and adaptive control.

Literature References:

[1] Widrow B., Hoff M. E.: Adaptive Switching Circuits. IRE WESCON Convention Record, New York: IRE, 96—104 (1960).

[2] Rumelhart D. E., Hinton G. E., and Williams R. J.: Learning Internal Representations by Error Propagation. In Parallel Distributed Processing, vol. 1, pp. 318—362 (M.I.T. Press, 1986).

[3] Hopfield J. J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. Natl. Acad. Sci. USA 79,2554—2558 (1982).

[4] Almeida L. B.: A Learning Rule for Asynchronous Perceptrons with Feedback in a Combinatorial Environment. Proc. of the First International Conference on Neural Networks, pp. 609—618 (San Diego, 1987).

[5] Williams R. J., Zipser D.: Experimental Analysis of the Real-Time Recurrent Learning Algorithm. Connection Science 1, 87—111 (1989).

[6] Nerrand O., Roussel-Ragot P., Personnaz L., Dreyfus G., Marcos S., Macchi O., Vignat C.: Neural Network Training Schemes for Non-linear Adaptive Filtering and Modelling. International Joint Conference on Neural Networks (Seattle, 1991). Marcos S., Macchi O., Vignat C., Dreyfus G., Personnaz L., P. Roussel-Ragot: A Unified Framework for Gradient Algorithms Used for Filter Adaptation and Neural Network Training. International Journal of Circuit Theory and Applications, in press.

[7] Pineda F.: Generalization of Backpropagation to Recurrent Neural Networks. Phys. Rev. Lett. 59, 2229—2232 (1987).

[8] Rohwer R., Renals S.: Training Recurrent Networks. In Neural Networks from Models to Applications, Personnaz L., Dreyfus G., eds., pp. 207—216 (I.D.S.E.T., Paris, 1989).

[9] Robinson A. J., Fallside F.: Static and Dynamic Error Propagation Networks with Application to Speech Encoding. In Neural Information Processing Systems, Anderson D. Z., ed., pp. 632—641 (American Institute of Physics, 1988).

[10] Nguyen D., Widrow B.: The Truck Backer-Upper: an Example of Self-Learning in Neural Networks. In Neural Networks for Control, Miller III W. T., Sutton R. S., Werbos P. J., eds, pp. 287—299 (MIT Press, 1990).

[11] Williams R. J., Peng J.: An Efficient Gradient-Based Algorithm for On-Line Training of Recurrent Network Trajectories. Neural Computation 2, 490—501 (1990).

[12] Narendra K. S., Parthasarathy K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. on Neural Networks 1, pp. 4—27 (1990).

Literature Survey

Fukushima K.: A Hierarchical Neural Network Model for Selective Attention, In:Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG,Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds: Eckmiller R.,Malsburg Ch.), 1989 pp.81-90

Abstract: A neural network model, which has the function of selective attention in visual pattern recognition and in associative recall, is proposed and simulated on a digital computer. The model has the function of segmentation and pattern-recognition. When a composite stimulus consisting of two patterns or more is presented, the model focuses its attention selectively to one of them, segments it from the rest, and recognizes it. After that, the model switches its attention to recognize another pattern. The model also has the ability to restore and imperfect pattern, and can recall the complete pattern in which the defects have been corrected and the noise eliminated. These functions are performed successfully even if the input patterns are deformed in shape or shifted in position.

Gallant S.I.: Perceptron-Based Learning Algorithms, IEEE Transactions on Neural Networks, Vol.1, 1990, No.2 pp.179-191

Abstract: This paper examines several supervised learning algorithms for single-cell and for network models. The heart of these algorithms is the pocket algorithm, a modification of perceptron learning that makes it well behaved with nonseparable training data, even if that data is noisy and contradictory. Features of these algorithms include: 1) speed; 2) network scaling properties; 3) analytic tractability; 4) on-line learning; and 5) winner-take-all groups or choice groups.

Gelenbe E.: Stability of the Random Neural Network Model, Neural Computation, Vol.2, 1990, No.2, pp.239-247

Abstract: The signal flow equations of the network, which describe the rate at which positive or negative signals arrive at each neuron, are nonlinear we show that whenever the solution to these signal flow equations exists, it is unique. We then examine two sub classes of networks — balanced and damped networks — and obtain stability conditions in each case. In practical terms, these stability conditions guarantee that the unique solution can be found to the signal flow equations and therefore that the network has a well-defined steady-state behavior.

Granger R., Ambros-Ingerson J., Lynch G.: Derivation of Encoding Characteristics of Layer II Cerebral Cortex,Journal of Cognitive Neuroscience Vol.1, 1989 No.1 pp.61-87

Abstract: We offer the hypothesis that the network function of superficial cerebral cortical layers may simultaneously acquire and hierarchically organize information about the similarities and differences among perceived stimuli. Experimental manipulation of the simulation has generated hypotheses of direct links between the values of specific biological features and particular attributes of behavior,generating testable physiological and behavioral predictions.

Granger R., Ambros-Ingerson J., Staubli U., Lynch G.: Memorial operation of multiple, interacting simulated brain structures In: Neuroscience and Connectionist Models Hillsdale, Erlbaum associates, 1989

Abstract: We constructed simulations of the superficial layers of ol factory cortex that received inputs at the theta frequency.

Hampshire J.B., II, Waibel A.H.: A Novel Objective Function Function for Improved Phoneme Recognition Using Time-Delay Neural Networks, IEEE Transactions on Neural Networks Vol.1,1990 No.2 pp.216-228

Abstract: This paper presents single and multiplexer recognition results for the voiced-stop consonants/b, d, g/using time delay neural networks. A new objective function is introduced for training these networks which seeks to maximize the differences between the output activation of the node representing the correct classification and all other nodes (representing incorrect classifications). A simple arbitration mechanism is used to achieve a median 30 time reduction in the number of misclassifications.

Hartman E., Keeler J.D., Kowalski J. M.: Layered Neural Networks with Gaussian Hidden Units as Universal Approximations, Neural Computation Vol.2, 1990 No.2 pp.210-215

Abstract: A neural network with a single layer of hidden units of gaussian type is proved to be a universal approximator for real-valued maps defined on convex, compact sets of Rn.

Hartmann G.: Mapping Images to a Hierarchical Data Structure -A Way to Knowledge-Based Pattern Recognition In:Neural Computers(Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf),FRG, Held:September 28-October2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds:Eckmiller R.,Malsburg Ch.), 1989 pp.91-100

Abstract: The Hierarchical Structure Code (HSC) provides a transition between the signal space of a gray-scale image and the space of symbolic description. Continuous objects are mapped to code-trees of the HSC-network by hierarchical linking operations. The HSC-network is controlled by a system of inhibition mechanisms, extracting invariant feature a from the code trees.Extracted features are compared with modelled features in the knowledge based.

Hassoun M. H., Sanghvi A. J.: Fast Computation of Optimal Paths in Two— and Higher-Dimension Maps, Neural Networks, Vol. 3, 1990, No. 3, pp. 355-363

Key words: combinatorial optimization; artificial neural networks; optimal path; distributed cost map; potential surface;fine-grained processor.

Abstract: Highly interconnected networks of relatively simple processing elements are shown to be very effective in solving difficult optimization problems. Problems that fall into the broad category of finding a least cost path between two points, given a distributed and sometimes complex cost map,are studied in this paper. A neural like architecture and associated computational rules are proposed for the solution of this class of optimal path finding problems in two— and higher-dimensional spaces.

Hecht-Nielsen R.: Neurocomputing New York, Addison-Wesley Publishing Company, 1990, 433 p. ISBN 0-201-09355-3

Hecht-Nielsen R., Evans K. M.: A Method for Error Surface Analysis, In: NEURONET 90, Prague, CSFR, 10.-14. 9. 1990

Key words: backpropagation; error surfaces.

Heeger D.J.,Jepson A.: Visual Perception of Three-Dimensional Motion, Neural Computation Vol. 2, 1990 No. 2 pp. 129-137

Abstract: Nonlinear equation describing the optical-flow field can be split by an exact algebraic manipulation to yield an equation that relates the image velocities to the translation al component of the 3D motion alone. Thus, the depth and the rotational velocity need not be known or estimated prior to solving for the translational velocity. The algorithm applies to the general case of arbitrary motion with respect to an arbitrary scene.

Hinton G. E.: Deterministic Boltzmann Learning Performs Steepest Descent in Weight-space, Neural Computation, Vol. 1, 1989, pp. 1-10

Abstract: By using the appropriate interpretation for lathe way in which a „deterministic Boltzmann machine" DBM represents the probability of an output vector given an input vector, it is shown that the DBM performs steepest descent in the same function as the original „stochastic Boltzmann machine" SBM, except at rare discontinuities. A very simple way of forcing the weights to become symmetrical is also described, and this makes the DBM more biologically plausible than back-propagation (Werbos, 1974; Parker, 1985; Rumelhart, Hinton and Williams, 1986).

Hirai Y., Tsukui Y.: Position Independent Pattern Matching by Neural Network, IEEE Transaction on systems, Man and Cybernetics vol. 20 , 1990 No. 4 pp. 816-825 S

Abstract: A new pattern matching neural network is proposed.The network matches an input to multiple candidates of the stored templates in parallel.It can find the best matching template, whose features are arranged in the same order as those of the input, regardless of positional differences between corresponding features.

Húsek D., Pokorný J.: Spreading Activation Methods in Information Retrieval — A Connectionist Approach In: NEURONET'90, Prague, September 10-14, 1990, ČSFR 1990, pp.134-139

Abstract: An improving of spreading activation methods own information retrieval systems is designed. As a new tools an appropriate type of a neural network was developed. The neural network, RETRIEVALNET called here, is a composition of two subnets (recursive and feed forward) extended by a level of inhibitory neurons.

Irvine D. R. F., Phillips D. P.: Polysensory „Association" Areas of the Cerebral Cortex In:Cortical Sensory Organization Clifton, Humana Press, 1982 pp.111-156

Izui Y., Pentland A.: Analysis of Neural Networks with Redundancy, Neural Computation Vol. 2, 1990 No. 2 pp.226-238

Abstract: Biological systems have a large degree of redundancy, a fact that is usually thorough to have little effect beyond providing reliable function despite the death of individual neurons. We have discovered, however, that redundancy an qualitatively change the computations carried out by a network. We prove that for both feedforward and feedback networks the simple duplication of nodes and connections results in more accurate, faster,and more stable computation.

Ji Ch., Snapp R. R., Psaltis D.: Generalizing Smoothness Constraints from Discrete Samples, Neural Computation Vol. 2, 1990, No. 2, pp. 188-197

Abstract: We study how certain smoothness constraints, for example, piecewise continuity, can be generalized from a discrete set of analog-valued data, by modifying the error backpropagation, learning algorithm. Numerical simulations demonstrate that by imposing two heuristic objectives — (1)reducing the number of hidden units, and (2) minimizing them altitudes of the weights in the network — during the learning process, one obtains a network with a response function that smoothly interpolates between the training data.

Jones E. G.: Anatomy of Cerebral Cortex: Columnar Input-Output Organization In: The Organization of the Cerebral Cortex Cambridge, The MIT Press (Ed.: Steph. G. Dennis), 1981 pp. 199-235

Abstract: In the cerebral cortex, the search for manageable circuit elements comparable, say, to the mossy fiber-granule cell-Purkinje cell circuit of the cerebellum has been hampered by lack of knowledge regarding the meaning of cortical lamination and a lack of agreement regarding the classes and distribution of interneurons. Recent work indicates that in higher primates, each lamina of the cortex to a large extent represents an aggregation of pyramidal-cell somata whose axon project to the same cortical or subcortical site. There is even evidence for a sublaminar organization along these lines inlayers III and V.

Kanerva P.: Self-Propagation Search In: CSLI-84-7, Cent. for the Study of Language and Information Palo Alto, Standford University, 1984

Abstract: At issue is the ability of humans to retrieve inform. from memory according to content (recalling and recognizing previously encountered objects) and temporal sequence(perform. a learned sequence of act.). Retrieval times indicate the direct retrieval of stored information. A mathem. theory of memory is developed. Memory items are represented by n-bit binary words(points of the space 0,1.). The unifying principle is that the address space and the datum space are the same.As in the conventional random-access memory of a comp., any stored item can be accessed directly, and sequent. retrieval is achieved by storing the memory record as a pointer chain. Accessing of many locations at once accounts forrecognit. Three main results are obtained: (1) The propert. ofneur. allow their use as address decoders for a generalized random-access memory; (2) distrib.the storage of an item in a set of locat.makes very large addr. spaces (2.,000) pract.; (3) struct.similar to those suggested by the theory are found in the cerebellum.

Karnin E.D.: A Simple Procedure for Pruning Back-Propagation Trained Neural Networks, IEEE Transactions on Neural Networks vol., 1, 1990 No. 2 pp.239-242

Keller J. M., Hunt D. J.: Incorporating Fuzzy Membership Functions into the Perceptron Algorithm, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, 1985 No. 6 pp. 693-699

Abstract: The perceptr. algorithm, one of the class of gradient descent techniques, has been widely used in pattern recognition to determine linear decision boundaries. While this algorithm is guaranteed to converge to a separating hyperplane if the data are linearly separable,it exhibits perratic behavior if the data are not linearly separable. Fuzzy set theory is introduced into the perceptron algorithm to produce a „fuzzy algorithm“ which ameliorates the convergence problem in the nonseparable case. It is shown that the fuzzy perceptron, like its crisp counterpart, converges in the separable case. A method of generating membership functions.

Klop H.: The Hedonistic Neuron Washington, Hemisphere Publishing Corporation, 1982

Koch Ch.: Computing Motion in the Presence of Discontinuities: Algorithm and Analog Networks In:Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds: Eckmiller R., Malsburg Ch.), 1989 pp. 101-110

Abstract: In this paper we will describe recent developments in the theory of early vision which lead from the formulation of the motion problem as ill-posed problem to its solution by minimizing certain „cost“ functions. These cost or energy functions can be mapped onto very simple analog and binary resistive networks. Thus, we will see how the optical flow can be computed by injecting currents into „neural“networks and recording the resulting stationary voltage distribution at each node. These networks can be implemented inc MOS VLSI circuits and represent plausible candidates for biological vision systems.

Koenderink J. J.: Design Principles for a Front-End Visual System in: Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG, Held:September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds: Eckmiller R., Malsburg Ch.), 1989, pp. 111-118

Abstract: In this paper I outline some of the a priori principles relating to the very from and of vision systems and try to correlate these with current knowledge concerning the primate visual system.

Kohonen T.: The Role of Adaptive and Associative Circuits in future Computer Designs In: Neural Computers, Neuss(Dusseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag Eds.: Eckmiller, Malsburg, 1989 pp. 1-8

Abstract: The fifth-generation computers have essentially been designed around knowledge data bases. Originally they were supposed to accept natural data such as speech and images; these functions, however, have new been abandoned. Some vistas to sixth— and later-generation computers have already been present, too. The latter are supposed to be even more „natural“. Certainly the fifth-generation computers have applied very traditional logic programming but what could the alternative be? What new functions can evolve from the „neuralcomputer“ principles? This presentation expounds a few detailed questions of this kind. Also some more general problem areas and motivation for their handling are described.

Konishi M., Knudsen E. I.: A Theory of Neural Auditory Space In: Cortical Sensory Organization Clifton, Humana Press (Ed.: Woolsey C. N.), 1982 pp. 219-229

Korn A. F.: Towards a Primal Sketch of Real World Scenes in Early Vision In: Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds: Eckmiller R., Malsburg Ch.), 1989 pp. 119-128

Abstract: The problem of symbolic represent.of intensity variations in gray value pictures of real scenes is studied. The goal is to relate the responses of a filter bank of different gradient filters to the structure of the picture which is determined by the physics of the image generation process. A simple criterion is proposed for the selection of a suitable center frequency of the involved bald-pass filters. The gradient vectors of the image function give the direction of maximal intensity changes with high resolution (8 bit) which can be used for an invariant shape description by corner points of a contour. The picture is segmented by closed contour lines in to regions which form a topographic representation in the picture domain.

NEURAL NETWORKS SIMULATOR

P. Bitzan)*

Abstract:

Programmed models of neural nets sometimes exhibit slow training and uncertain testing which can fail if the system converges into improper minima. Subsequently, they face strong competition from standard algorithms. When one is interested only in applications, neural nets can be simulated highly effectively on the base of non-adaptive algorithms. We have described here a so-called neural networks simulator, which provides simulation of neural nets by means of a pseudo-metric which is constructed in the space of training samples.

1. Introduction

The basic general property of neural nets as used in computer science is the ability to learn a mapping on the base of some finite number of examples of such a mapping (training samples). This means that there is no need to specify any additional rules or principles and the only thing the user must provide are these examples.

Generally this mapping, denoted by O, can be defined from the n-dimensional real space R^n to the k-dimensional real space R^k_j; hence O: $D \subset R^n \rightarrow R^k$. Training samples in this case are formed by a finite number of vector pairs, where the first is n-dimensional and the second k-dimensional. Let us denote the set of training samples by

$$L = \{(x_i, y_i),\ i = 1,2..K;\ x_i \in D \subset R^n,\ y_i \in R^k, \quad (1)$$
$$O: x_i \rightarrow y_i\}$$

A neural net learns a mapping usually by repeatedly setting training samples on its input and output neurons. In this way inner weights of the net are adapted and after the learning is finished, the neural net is able to realize the mapping from the whole domain $D \rightarrow R^k$. We can say that the mapping formally defined on the finite set of points (training samples) is enlarged into the whole domain D. This enlarging is not arbitrary of course, but represents some hidden principles of the mapping being learned.

The kind of information processing just described can be called neural network-like information processing. Obviously, this is a simple procedure and it is appropriate particularly in cases where it is difficult to specify some attributes of the information being processed.

Unfortunately programmed models of neural nets are somewhat time consuming, particularly in the training phase of perceptron-like paradigms. Depending on the architecture and on the set of training samples these weights can converge very slowly or don't converge at all. The testing phase of a neural net has similar problems; the system can converge to some improper point.

Let us elucidate some problems of programmed models of neural nets a bit more precisiously. From the general point of view programmed models of neural nets suffer from the improper connection between naturally parallel neural nets and existing computer systems, which don't usually support higher levels of parallelism. Natural neural nets are systems consisting of a great number of elements (neurons). This is not exceptional; everything is composed of a large number of elements, molecules, atoms, etc., and there are algorithms which can acceptably model such systems on computers. However, in these cases it is possible to group some elements into classes according to their properties and hence we are able to simplify the whole description. In the case of neural nets, these techniques are hardly acceptable, because each neuron has its own important function and when modeling a neural net, we must take every neuron into account. The case where a large number of neurons are present differs from common physical systems and much parallel computing is necessary, something that current computers are not prepared to master. Special co-processors can only increase the low efficiency of possible algorithms, but they can't change it substantially. Thus, the computation of neural networks on current computers resemble an army attacking through a tube.

If we omit physical models of neural nets, then this problem of low efficiency can be solved in three ways.

The first one is the improvement of existing algorithms for neural networks. Nevertheless, such an improvement, though possible, has its limit. Considering the very low efficiency of available algorithms one could predict that any possible speed up won't be significant.

The second way is based on large parallel computer architectures such as, e.g., CM1 and CM2 [1], which consist of many thousands of processor units. Systems of such a high parallelism might be appropriate for effective neural networks computing.

The third way to cope with the low efficiency of neural networks computing is through simulation.

*) *Pavel Bitzan*
The Institute of Computer and Information Science
Czech Academy of Sciences, P.O. Box 5, 182 07 Prague 8, Czechoslovakia

This simulation can be done regardless of their real structure and can substantially accelerate the learning phase, which is the most time consuming part of this kind of processing. For this purpose we have developed a neural networks simulator [2] which is described in the next section.

2. Neural Networks Simulator

We could imagine that every type of a neural network uses some inner pseudo-metric on the domain D. If we have determined this pseudo-metric , then we could interpolate our resulting mapping using pseudo-distances among training samples. Let us propose how such an appropriate pseudo-metric can be determined.

For each $x_i \varepsilon L_D$ *(projection of* L into D) let us define vectors $x_i^n, x_i^{n-1}, .. x_i^2$ *as follows:*

$$\begin{aligned}
x_i^n &= x_i, & x_i^n &\in R^n,\\
x_i^{n-1} &= [x_{i2}^n - x_{i1}^n, x_{i3}^n - x_{i2}^n, .., x_{in}^n - x_{in-1}^n], & x_i^{n-1} &\in R^{n-1},\\
x_i^{n-2} &= [x_{i2}^{n-1} - x_{i1}^{n-1}, x_{i3}^{n-1} - x_{i2}^{n-1}, .., x_{in-1}^{n-1} - x_{in-2}^{n-1}], & x_i^{n-2} &\in R^{n-2},\\
&\;\;\vdots & &\;\;\vdots\\
x_i^2 &= [x_{i2}^3 - \mathbf{x}_{i1}^3, x_{i3}^3 - x_{i2}^3], & x_i^2 &\in R^2.
\end{aligned} \tag{2}$$

The vectors defined above represent $0,1 \ldots n-1$ differences of vector x_i and we call them moments of $0,1 .. n-1$ order of data pre-processing.

The principles of physical systems are usually characterized by means of differential description. Their behavior mostly does not depend on current values of parameters, but rather on their relations to each other. Hence we can establish the following generalization for j vectors x_i^j.

Let us define for every vector x_i^j, $i = 1,2..K$; $j = n, n-1,.2$, a binary matrix $m_i^j = \{m_{ikl}^j;\ k, l = 1,2,..j\}$ as follows:

$$\begin{aligned}
&if(x_{ik}^j < x_{il}^j) \rightarrow m_{ikl}^j = 1,\\
&if(x_{ik}^j > x_{il}^j) \rightarrow m_{ikl}^j = 0,\\
&if(x_{ik}^j = x_{ib}^j) \text{ then } if\,(k \leq 1) \rightarrow m_{ikl}^j = 1,\\
&if(x_{ik}^j = x_{ib}^j) \text{ then } if\,(k > 1) \rightarrow m_{ikl}^j = 0.
\end{aligned} \tag{3}$$

For the transformation $(0 \rightarrow 1)$ and $(0 \rightarrow 1)$ m_i^j is a anti-symmetric matrix which contains ordering among components of a vector x_i^j.

Hence, every vector $x_i \varepsilon L_D$ can be characterized by at most $n-1$ binary matrices (3); this matrices we formally denote by M_i. This description is appropriate for computer processing because many compilers can operate in bit arithmetic mode. It also considers the fact that physical principles can be expressed by means of differential (difference) description, which is, in fact, a kind of a description of data ordering.

Let τ: $M^n \times M^n \rightarrow R^+ + [0]$ be a symmetric, non-negative function of the form as follows:

$$\tau(M_1^n, M_2^n) = \text{abs}\,)\sum_{j=2}^{n} \sum_{k=1}^{j} \sum_{l=1}^{j} \alpha_{kl}^j \delta_{kl}^j) \tag{4}$$

$$\delta_{kl}^j = \begin{matrix} 1 & if(m_{1kl}^j \neq m_{2kl}^j) \\ 0 & if(m_{1kl}^j = m_{2kl}^j), \end{matrix}$$

where $\alpha_{kl}^j \varepsilon R$, $k,l = 1,2,..j$ *are arbitrary, in* k,l symmetric coefficients. Considering the mapping (3), then (4) establishes a pseudo-metric in R^n as follows:

$$\rho(x_1, x_2) = \tau(M_1^n, M_2^{n/}). \tag{5}$$

Hence, this pseudo-metric is defined in D by (2), (3), (4), (5) and coefficients α_{kL}^j. In contrast to a regular metric, (5) accepts zero distance between different points and in general, does not satisfy triangle inequality.

The natural way of establishing these metric coefficients α_{kL}^j is to require mapping O (See (1)) to be "as continuous as possible". This requirement implies the triangle inequality is to be satisfied „as much as possible". Hence we can require the following expression

$$R = \sum_{i=1}^{N} \sum_{j=i+1}^{N} (\rho(x_i, x_j) - e(y_i, y_j))^2 \tag{6}$$

to be minimal; e denotes Euclidean distance.

We can say that we seek for such α_{kl}^j that corresponding pseudo-distances among x_i in R^n are closed "as possible" to corresponding Euclidean distances among y_i *in* R^L in the sense that (6) reaches its minimum.

Example.

Let us take the following set of training samples, which map symmetric 3-dimensional vectors to 1 and non-symmetric ones to 0:

$x_1 = [1,2,1] \rightarrow y_1 = [1]$ $\qquad$ $x_2 = [2,1,2] \rightarrow y_2 = [1]$

$x_3 = [1,2,3] \rightarrow y_3 = [0]$ $\qquad$ $x_4 = [3,2,1] \downarrow y_4 = [0]$.

Let us limit ourselves to the zero moment of data pre-processing (See (2)) and according to (3), we can process vectors x_1, x_2, x_3 and x_4 to the form of binary matrix:

$$\begin{matrix} 0\,1\,1 \\ 0\,0\,0 \\ 0\,1\,0 \end{matrix} \rightarrow [1] \quad \begin{matrix} 0\,0\,1 \\ 1\,0\,1 \\ 0\,0\,0 \end{matrix} \rightarrow [1] \quad \begin{matrix} 0\,1\,1 \\ 0\,0\,1 \\ 0\,0\,0 \end{matrix} \rightarrow [0] \quad \begin{matrix} 0\,0\,0 \\ 1\,0\,0 \\ 1\,1\,0 \end{matrix} \rightarrow [0].$$

In order to find $R_{\min}$ we can directly require $\rho(x_1, x_2) = 0$, $\rho(x_1, x_3) = 1$, $\rho(x_1, x_4) = 1$, $\rho(x_2, x_3) = 1$, $\rho(x_2, x_4) = 1$, $\rho(x_3, x_4) = 0$, which implies the following

equations for the upper triangle coefficients α_{kl}^0:

$\text{abs}\,(\alpha_{12}^0 + \alpha_{23}^0) = 0$ $\qquad$ $\text{abs}\,(\alpha_{23}^0) = 1$

$\text{abs}\,(\alpha_{12}^0 + \alpha_{13}^0) = 1$ $\qquad$ $\text{abs}\,(\alpha_{12}^0) = 1$

$\text{abs}\,(\alpha^0_{13} + \alpha^0_{23}) = 1 \qquad \text{abs}\,(\alpha^0_{12} + \alpha^0_{13} + \alpha^0_{23}) = 0.$

The solution of this set of equations is $\alpha^0_{12} = 1$, $\alpha^0_{13} = 0$ and $\alpha^0_{23} = -1$. Because the whole matrix α^0_{KL} is symmetric, and diagonal coefficients don't change the pseudo-metric, ρ is complete.

In general, the evaluation of metric ρ is a problem of unconstrained optimization and of course it may assume higher moments of data processing.

While training the system we store the matrices M_i with corresponding vectors y_i in some I/O file, which e.g., we can denote MEMORY. Once the pseudo-metric is determined, it is possible to exclude those samples which are sufficiently close to some former ones (with respect to the chosen pseudo-metric), so as to reduce the stored redundant information. Then the training is finished.

Once ρ is determined, we can test our simulator, i.e., for some testing $x \varepsilon D \subset R^m$ we can construct its mapping. First we determine some number of „nearest" samples and then on this set we can interpolate the mapping of x, for instance by the formula as follows:

$$y = \text{NORM}^{-1} \sum_i \rho(x_i, x)^{-\gamma} y_i, \qquad (7)$$

$$\text{NORM} = \sum_i \rho(x_i, x)^{-\gamma},$$

where $\gamma > 0$ is a shape factor. If for some i we have $\rho(x_i, x) = 0$, then we define $y = y_i$.

In order to find these „nearest" samples, it would be ineffective to evaluate distances from all memorized samples. For a particular pre-selection we can use appropriate global quantities which could be joined to stored training samples and which would provide their global characterization.

By the method just described, we are able to process information in the same way that neural networks do, that is without any specification of rules or principles. Because most of the data processing concerns binary matrix operations, this method is advantageous particularly when it is implemented on a computer system with binary arithmetic mode. If this isn't available it is appropriate to code binary matrix into real digits.

In contrast to neural nets the learning phase of this method is sped-up quite markedly. This speed-up is more significant for greater numbers of neurons. We shall dismiss the actual descriptions of problems which have been solved and compared; instead we shall demonstrate relative performance in the two following exemplary time comparisons of the back-propagation algorithm. All tests have been processed on a standard PC AT-286.

A back-propagation network with 10 input, 6 hidden and one output neurons spent more than 2 hours learning 44 training samples. The described simulator took only 10 seconds.

A back-propagation network with 5 input, 80 hidden and 180 output neurons spent about 10 hours learning 62 training samples using the ANZA co-processor. Without this co-processor, we can estimate that the time needed for learning could be more than 10 days. In this same case, the simulator took less then 25 seconds.

According to the description of the simulator, it is clear that when training, there is no need to adapt any weights. Besides the pseudo-metric optimization which can be solved using standard optimization methods, the only thing we must do is to store binary matrices in MEMORY provided they are not redundant. This is the reason for such an extreme speed-up in comparison to standard adaptive algorithms. We should also admit that in the testing phase, the simulator might take a bit longer than the back-propagation algorithm, since the program must seek „near stored samples" and evaluate their pseudo-metric distances from the testing sample.

It is necessary to also emphasize that the described simulator could fail when using a small number (5 or less) of input neurons. Due to transformations on binary matrices, many different samples might degenerate improperly; that is, they could be improperly represented by the same record. This instance can be removed by special pre-processing which is not described in this paper. This pre-processing assumes the addition of an appropriate number of components to input vectors.

We could also study the relations among various types of neural networks and various types of the above-mentioned pseudo-metrics. We can't prove the theorem that every neural network with arbitrary training samples can be substituted for the afore-mentioned simulator, but we can verify this statement for many cases when testing it on computers.

Nevertheless, we consider this type of study to be a bit unnecessary, because our goal is not to simulate exactly various neural networks, but to simulate their properties that are useful in information processing.

3. Conclusion

We have introduced the basic principles of the neural networks simulator. Considering training and testing difficulties of programmed models of neural nets, the proposed simulator might be expected to find a general application, especially in large problems of neural network-like information processing.

The neural networks simulator has been used for recognizing indications of heart diseases. As was described, it learns only on the basis of training vectors whose dimensions vary within the range 50 to 800 input neurons. There is no need to specify any additional rules or principles and the only thing the expert must provide is a specification of what disease the sample under consideration represents. In addition, the above-mentioned I/O file MEMORY can be easily modified by adding new samples or by cancelling some of the old ones. This can be done even during the testing phase and hence the system can be continuously updated. For this problem, it is possible to use a standard PC AT with an acceptable response time.

References

[1] Hockney J.: Parallel Computers
Adam Hilger, Bristol and Philadelphia ,1988

[2] Bitzan P.: Neural networks simulator
Research Report V-481, Institute of Computer and Information Science, Prague, 1990

[3] Clark G.C.,Jr., Cain J.B.: Error-Correction Coding for Digital Communications , Harris Corporation
Melbourne, Florida, 1986

[4] Tsuda I.,.Koerner E., Shimizu H.: Progr. Theor. Phys.,Vol.78, 1987, 51-71

[5] Reggia J.A., Sutton G.G.: Proceedings of the IEEE,Vol.76, 1988, 680-692

[6] Feldman J.,.Ballard D.: Connectionist models and their properties Cognitive Science,vol.6, 1982, 205-254

[7] Hopfield J.: Neural networks and physical systems with emergent collective computational abilities
Proceedings of the National Academy of Sciences Vol.79, 1982, 2554-2558 [8]... Ikegami T., Suzuki M.: Prog.Theor.Phys. Vol.78, 1987, 38

[9] Small S., Shatri L., Brucks M., Kauffman S., Cottrell G., Addanki A.: ISCON-A network construction aid and simula tor for connectionist models
Tech. Rep. No.109, Department of Computer Science, University of Rochester

INTERESTING AND COMING EVENTS

In this section of our Journal the information on some interesting and coming conferences, symposiums and seminaries are given:

JUNE 1991

Fourth Int'l Conf. on Industrial and Eng. Applications of Artificial Intelligence and Expert Systems, June 2—5, Kauai, Hawaii. Sponsors: ACM et al. Contact Moonis Ali, Univ. of Tennessee Space Inst., MS15, B. H. Goethert Pkwy., Tullahoma, TN 37388-8897, phone (615) 455-0631, ext. 236, fax (615) 454-2354, e-mail alif@utsiv l. bitnet.

Fifth Supercomputing Symp. June 3—5, 1991, Fredericton, N.B., Canada. Cosponsors: Canadian Special Interest Group on Supercomputing, Univ. of New Brunswick. Contact Virendra C. Bhavsar or Uday G. Gujar, Faculty of Computer Science, Univ. of New Brunswick, Fredericton, N. B., E3B 5A3, Canada, phone (506) 453-4566, fax (506) 453-3566.

Parle 91, Conf. on Parallel Architectures and Languages Europe, June 10—13, 1991, Einhoven, The Netherlands. Consponsors: Commission of European Communities et al. Contact F. Stoots, Philips Research Labs, PO Box 80.000, 5600 JA Einhoven, The Netherlands, fAX 31 (40) 744-758,e-mail: stoots@dooma. prl. philips. nl.

1991 IEEE International Symposium on Circuits and Systems — ISCAS '91, June 11—14, 1991, Raffles City, Singapore. Contact: ISCAS '91 Secretariat c/o Communication Int'l Associates Pte Ltd. 44/46 Tanjong Pagar Road, 0208 Singapore (65) 226-2838.

1991 ACM International Conference on Supercomputing , June 17 — 21, Cologne, Germany.
Cosponsors: Gesellschaft fuer Informatik; contact Ruediger Esser, FKA-ZAM, D-5170 Juelich, Germany, phone (0049) 2461 61 6588, fax: (0049) 2461 61 6656, e-mail: zdv003@djukfa11. bitnet

First Int'l Conference on AI in Design, June 25—27,1991, Edinburgh, UK; contact Helen Hodge, Butterworth Scientific Ltd., Westbury House, Bury St., Guildford, Surey GU2 %BH, UK; 04 (83)300966.

AIME-91 AI in Medicine, June 24—27, Maastricht NL. The scope ranges from theoretical research to practical applications. Main themes: knowledge representation and modelling, knowledge acquisition and explanation, temporal and spatial reasoning, uncertainty management, expert databases, KBS evaluation, clinical applications and integration, technology assessment of KBS. Cosponsors: AIME-91, Univ. of Limburg, PO Box 616, 6200 MD Maastricht, The Netherlands. Fax+31 43 436080, phone ... 43888409. AIME 1991@HMARL%. Bitnet.

ICANN 91, International Conference on Artificial Neural Networks, June 24—28, Espoo, Finland
Cosponsors: IEEE Neural Network Council, International Neural Network Society. Contact: Congress Management Systems, P. O. Box: 151, SF — 00141 Helsinki, Finland, phone: (003580) 175 355, fax: (003580) 170 122

6th Czechoslovak Conf. on AI, June 25—27, Prague CS. Contact: Vladimir Marik, Czech Technical University, K 335, Technicka 2, 166 27 Prague 6, Czechoslovakia. Fax +422 290159, phone ... 2 295664.

JULY 1991

PDK '91, International Workshop Processing Declarative Knowledge. Representation and Implementation Methods, July 1—3, 1991, Kaiserslautern, Germany. Sponsors: PDK, DFKI Gmbh, P. O.Box 2080, 6750 Kaiserslautern, F.R.Germany, fax:+49-631-205-3210. email: pdk@informatik. uni-kl.de.

ACAI-91, Adv. Course on AI, July 1—12, Bilbao ES. Theme of the course: Issue on Reasoning. Topics: Trends in Cooperative Distributed Problem Solving, Temporal reasoning, Case Based Reasoning and Learning, General Cognitive Architectures, Validation of Knowledge Based Systems, Knowledge Elicitation Techniques. Sponsors: Mikel Emaldi, LABEIN, IDEIA Dep. Cuesta de Olabeaga 16, E-48013 Bilbao, Spain. fax +34 4 441 1749, phone ... 441 9300, emaldi@labein.es.

1991 International Joint Conference on Neural Networks IJCNN '91, July 8—14, 1991, Seattle, WA., USA
Cosponsors: IEEE, International Neural Network Society, Contact: IJCNN 91, UW Extension, GH-25, 5001 25th Ave, NE, Seattle, WA. 98185, phone: (001 206) 543 2310, fax: (001 206) 685 9359, Diana Nielsen (001 206) 543 0888.

Third Conference on Innovative Applications of AI '91, July 15—17,1991, Anheim, CA. Contact IAAI '91, AAAI, 445 Burgess Drive, Menlo Park, CA 94025-3496; (405)328-3123.

International Congress of Psychology, July 19—24, 1991, Brussels, Belgium. Address: Brussels International Conference Centre, Parc des Expositions, Place de Belgique, B-1020 Brussels, Belgium. Tel.: +32 2 478 48 60. Fax: + 32 22 478 80 23. Telex: 234643 foire b.

AUGUST 1991

IEEE International Conference on Systems Engineering, August 1—3, 1991, Wright State University, Dayton, Ohio, USA
Contact: Dr. B. A. Shenoi, Electrical Engineering Departament, Wright State University, Dayton OH 45435, phone: (001 513) 873 3527, fax: (001 513) 873 3301

CRYPTO '91, Aug. 11—15, Santa Barbara, Calif. Cosponsors: Int,l Assoc. for Cryptologic Research et al. Contact Burt Kaliski, Crypto 91, RSA Data Security, 10 Twin Dolphin Dr., Redwood City, CA 94065, phone (415) 595-87822, fax (415) 595-1873, Internet: bur@trsa.com.

6th IEEE International Symposium on Intelligent Control, August 12—16, 1991, Key Bridge Mariott, Arlington VA, USA,
Contact: Prof. A. H. Levis, Dept. of E. C. E., George Mason University, Fairfax, VA 22030, phone: (001 703) 764 6282

IFAC Symposium on Distributed Intelligence Systems, DIS' 91, Aug. 13—15, 1991, Washington DC. Contact: A. H. Levis, M. I. T., 35-410/LIDS, Cambridge, MA 02139.

Twelfth International Joint Conference on Artificial Intelligence, IJCAI '91, Aug. 24—30. Sydney, Australia. Contact: Prof. J. Mylopoulos or Prof. R. Reiter, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4; Phone: + 1-416-978-1455; e-mail: ijcai@cs.toronto.edu.

DEXA '91 Int'l Conf. Database and Expert Systems Applications, August 28—30, Berlin-Potsdam DE. Contact: Dr.Dimitris Karagiannis, FAW-Ulm, AI-Lab, Helmholtzstrasse 16, P.O.Box 2060, D-7900 Ulm, Germany. Fax +49 731 501-999, phone ... 501-540. karagian@dulfawla.bitnet.

SSD 91, Second symp. on Large Spatial Databases, Aug. 28—30, Zurich, Switzerland. Contact: H. J. Schek, Inst. für Information Systeme, Eth Zentrum, 8092 Zurich, Switzerland, phone 41 (1) 254-7240.

SEPTEMBER 1991

14th Annual Meeting of the European Neuroscience Association, ENA, Sept. 8—12, Cambridge, England. Address: Conference Contact, 42 Devonshire Road, Cambridge CBI 2BL, UK. Tel: +44 223 323427. Fax: +44223 460396

17th Int'l Conf. on Very Large Data Bases, Sept. 3—6, Barcelona, Spain. Sponsors: IEEE Computer Soc. Tech. Committee on Data Eng. et al. Contact Guy Lohman, IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120, e-mail lohmani@bm.com.

2nd Ws Uncertainty Processing in Expert Systems, Sept. 9—12, Bechyne CS. The attendance of the workshop is limited to about 50 persons, lectures presenting unfinished work, opening new problems and provoking active discussion are welcome. Sponsors: WUPES, Czechoslovak Cybernetic Society, Pod vodarenskou vezi 4, 18208 Prague 8, CSFR. fax +42 2 815 2171, phone ... 815 2384.

1991 Annual International Course Neural Networks. Adaptive Neural Networks, Sept. 9—13, Garmisch-Partenkirchen, Germany. Course Director: Bernad Widrow. CEI-Europe/ Elsevier, BOX 910, S-61225.

Compsac 91, 15th Int'l Computer Software and Applications Conf., Sept. 11—13, Tokyo. Cosponsor: Information Processing Soc. of Japan. Contact Stephen S. Yau, Univ. of Florida, CIS Dept., Rm.301, Gainesville, FL 32611, phone (904)335-8006.

XS '91 BCS Conf. on Expert Systems, Sept. 17—19, London, UK. Cosponsor: Clearway International, attn mrs. Fiona Pearson, Conference House, 9 Pavilion Parade, Brighton BN2 1RA. fax +44 273 57 1224, phone ... 273 695 811.

First IEEE-SP Workshop on Neural Networks for Signal Processing, Sept. 29-Oct. 2, 1991, Sponsors: IEEE Signal Processing Society in cooperation with the IEEE Neural Networks Council. Contact Gary Kuhn, Publicity Chair, fax (609) 924-4600 or (609) 924-3061.

OCTOBER 1991

EPIA-91 5th Portuguese Conf on AI, Oct. 1—3, 1991, Albufeira, Algarve PT. Info: Prof. Pedro Barahona, Dep de Informtica, Univ Nova de Lisboa, 2825 Monte da Caparica, Portugal. fax +351 1 2955641, phone ... 295 4464. pb@fctunl.rccn.pt.

IEEE Workshop on Visual Motion, Oct. 6—9, 1991, Princeton, N. J. Contact Thomas S.Huang, Coordinated Science Lab, Univ. of Illinois, 1101 W. Springfield Ave., Urbana, IL 61801, phone (217) 333-6912.

11th IEEE Symposium on Mass Storage Systems, October 7—10,1991, Monterey, California,
Sponsor: IEEE Computer Society Technical Committee on Mass Storage Systems and Technology, Contact: B. T. O'Lear, NCAR, P. O. Box: 3000, Boulder, CO 80307, phone: (001 303) 497 1268, fax: (001 303) 497 1137

Sixth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Oct. 6—11, 1991, Banff, Canada. Contact John Boose, Advance Technology Center, Boeing Computer Services, 7L-64, PO Box 24346, Seattle, WA 98124; (206) 865-3253.

11th IEEE Symp. on Mass Storage Systems, Oct. 7—10, 1991, Monterey, Calif. Sponsor: IEEE Computer Soc. Technical Committee on Mass Storage Systems and Technology. Contact Bernard T. O.'Lear, NCAR, PO Box 3000, Boulder, CO 80307, phone (303) 497-1268, fax (303) 497-1137.

First Int'l Conf. on Artifical Intelligence Applications on Wall St., Oct. 9—11, 1991, New York City. Sponsor: Polytechnic Univ., Brooklyn NY 11201, phone (718) 260-3360, fax (718) 260-3136.

Workshop on Experimental Distributed Systems, Oct.12, 1991, Huntsville, Ala. Contact Raif M Yanney, TRW, 1 Space Park, DH2/2328, Redondo Beach, CA 90278, phone (213)764-6033.

ICCD '91. IEEE International Conference on Computer Design: VLSI in Computers & Processors, Oct. 14—16, 1991, Hyatt Regency Cambridge, Cambridge, Mass. Sponsor: IEEE Computer Society and IEEE Circuits and Systems Society. In Cooperation with: IEEE Electronic Device Society. Contact Dwight Hill, AT&T Bell Laboratories 3D-446, Murray Hill, NJ 07974, phone 201-582-7766, E-mail: dwight@research.att.com.

European Conference on Industrial Applications of Knowledge-Based Diagnosis, Oct. 17—18, 1991, Milano, Italy. Contact A. Camnasio, CISE, PO BOX 12081, 20134 Milano, Italy; (39) 2-21672400, fax (39) 2-26920587.

1st Int Conf Practical Application of Prolog, Oct. 28—31, 1991, Edinburgh UK. Info: Al Roth, 31 Bexley Avenue, Blackpool, Lancs FY2OTE, UK. fax +44 253 53811, phone ... 253 58081. alroth@cix.compulink.co.uk.

AI*IA 2nd Nat Congres on AI, Oct. 29—31, Palermo, Italy. Info: Prof. Salvatore Gaglio, CRES, Centro per la Ricerca Electronica in Sicilia, Viale Regione Siciliane 49, 90046 Montreale (Palermo), Italy. fax +39 91 640 6200; phone ... 640 6192/619/4501.

NOVEMBER 1991

TAI 91, Third IEEE Computer Soc. Conf. on Tools for Artifical Intelligence, Nov.5—8, 1991, San Jose, Calif. Contact Benjamin Wah, Coordinated Science Lab, MC 228, Univ. of Illinois, 1101 W. Springfield Ave., Urbana, IL 61801-3082, phone (217)333-3516, fax (217)244-1764, e-mail wah@aquinas%cso.unicu.edu; or Nikolaus G. Bourbakis, 4138 Moonflower Ct., San Jose, CA 95135, phone (4080284-6494.

First Int'l Conf. on Artifical Intelligence Applications on Wall St., Oct. 9—11, 1991, New York City. Sponsor: Polytechnic Univ., Brooklyn. Contact Mary Bianchi, Polytechnic Univ., 333 Jay St., Brooklyn NY 11201, phone (718)260-3360, fax (718)260-3136.

Int'l Joint Conference on Neural Networks '91, Nov. 18—22, Singapore. Contact Teck-Seng Low, Communicationn Int'l Associates, 44/46 Tanjong Pagar Rd., Singapore 0208; (65)226-2838; fax (65) 226-2877; e-mail mpeangh@nusvm.

Supercomputing 91, Nov. 18—22, 1991, Albuquerque, N.M. Cosponsor: ACM. Contact Raymond L. Elliott, Computing and Comm. Div., MS B260, Los Angeles Nat'l Lab, Los Alamos, NM 97545; or Supercomputing 91, IEEE Computer Soc., 1730 Massachusetts Ave. NW, Washington, DC 20036-1903, phone (202)371-1013.

DECEMBER 1991

Int'l Conf. on Parallel and Distributed Information Systems, Dec. 4—6, 1991, Miami Beach, Fla. Cosponsors: IEEE Computer Soc. et al. Contact Amit Sheth, Bellcore, IJ-210. 444 Hoes Ln., Piscataway, NJ 08854, phone (908)699-9011, e-mail amit@ctt.bellcore.com.

1991 IEEE Workshop on Speech Recognition, Dec. 15—18, 1991, Harriman, NY. Contact Jay G. Wilpon (201)582-3559.

World Congress on Expert Systems, Dec. 16—19, 1991, Orlando, Fla. Cosponsors: Int'l Assoc. of Knowledge Engineers et al. Contact World Congress on Expert Systems, c/o Congress Secretariat, Congress (USA), Inc., 7315 Wisconsin Ave., Suite 404E, Bethesda, MD 20814, phone (301)469-3355, fax (301)469-3360.

JANUARY 1992

25th Annual Hawaii International Conference on System Sciences (HICSS — 25), January 7—10, 1992
Kauai, Hawaii; Contact: Dr. Bhushan Saxena, Department of Computing, Hong Kong Polytechnic, Hung Hom, Kowloon, Hong Kong, e-mail: cssaxena @ hkpcc.hkp.hk

MAY 1992

IEEE International Symposium on Circuits and Systems — ISCAS '92, May 10—13, 1992, Sheraton Harbor Island Hotel, San Diego, California; Contact: Dr. Stanley A. White, 433 Avenida Cordoba, San Clemente, CA 92 672, phone: (001 714) 498 5519

SEPTEMBER 1992

IEEE International Workshop on Robot and Human Communications, September 1—3, 1992, Hosei University, Tokyo, Japan; Contact: Prof. Hisato Kobayashi, Dept. of Electrical Engineering, Hosei University, Kajinocho, Koganei, Japan, phone: (0081) 423 87 6187, fax: (0081) 423 87 6122

JANUARY 1993

IEEE International Symposium on Information Theory, January 10—15, 1993, San Antonio, Texas, Contact: Mr. Costas N. Georghiades, Texas A&M University, Department of Electrical Engineering, College Station, TX 77843-3128, phone: (001 409) 845 7408

MAY 1993

IEEE International Symposium on Circuits and Systems — ISCAS '93, May 2—6, 1993, Sheraton Chicago Hotel and Towers, Chicago, Illinois; Contact: Dr. W. Kenneth Jenkins, Coordinated Science Lab., University of Illinois, 1101 Springfield Ave.,Urbana, IL 61801

SYSTEM IDENTIFICATION USING THE SYMMETRIC LOGARITHMOID AS AN ACTIVATION FUNCTION IN A FEED-FORWARD NEURAL NETWORK

Abhay B. Bulsari † *and Henrik Saxén* *)

Abstract

In most applications, the sigmoidal activation function is used without questioning its limitations. The sigmoid restricts the outputs from feed—forward neural networks to between —1 and 1, or 0 and 1. However, there are systems whose outputs are not constrained within —1 and 1, or 0 and 1, and for reasons of loss in sensitivity, it is not desirable to map the output range to 0 to 1. In such cases, the symmetric logarithmoid provides a viable alternative to the sigmoid, while preserving many characteristics of the sigmoid.

This paper illustrates the applicability of the symmetric logarithmoid activation function in a feed—forward neural network, exemplified by a system identification problem of a biochemical reactor. The inputs to the networks were the three state variables at a time, and the process input variables (control variables and disturbances) from that time to the time for which the state variables are to be predicted. This duration was 0.1 hour, and the characteristic time for the process was 2.9 hours under normal circumstances.

Levenberg—Marquardt method was used to train the neural networks by minimising the sum of squares of the residuals. In most cases, the symmetric logarithmoid resulted in lower error square sum values than the sigmoid. The predictions were quite accurate.

The symmetric logarithmoid is continuous, first—order differentiable and a simple, monotonically increasing algebraic function. Convergence is generally faster compared to the sigmoidal activation function. Extremely large weights are not commonly generated by the training process, but is a usual feature with the sigmoids.

1. Introduction

A lot of work has been done on feed—forward neural networks taking the sigmoid for granted. The sigmoid, however, has its limitations and its applicability is not universal. Sigmoids are applicable for outputs contrained between 0 and 1, or —1 and 1. A linear mapping can extend this range. But many variables do not have such limits, and it is not always desirable to map them to a range of 0 to 1 since sensitivity can be lost in the process of mapping. Feedforward neural networks can be used for system identification of processes [1], and one often comes across variables like temperature, pressure, viscosity, concentration, etc. which have no upper limits, although for a system under consideration, they may stay in a particular range.

System identification is an important task for chemical, biochemical and metallurgical processes, especially where the mathematical models are not accurate enough for control purposes. This is more true of biochemical processes, where models based on first principles are not so common.

Although back propagation has become popular on grounds of simplicity and its capability to learn sequentially from training instances, we have used the Levenberg—Marquardt method [2—5] for minimising the sum of squares of errors.

The sigmoid is very flat when the absolute value of its argument $|a_i| > 10$. In other words, its derivative is extremely small, and has poor sensitivity to its argument. This is the root cause of the very slow rates of convergence during the training phases of neural networks, and relative insensitivity of the network to a fairly wide range of weights.

The symmetric logarithmoid, given by the following equation overcomes these limitations despite, perhaps, creating some others of its own.

$$x_i = \frac{\beta a_i}{|\beta a_i|} \ln(1 + |\beta a_i|)$$

The sigmoid and the symmetric logarithmoid can be considered to be in a continuum of activation functions. One extreme of the activation functions is the switch (the sign function), a network based on which cannot be trained by any of the optimisation methods meant for continuous functions. The sigmoid alleviates this problem by smoothening the switch near its discontinuity. The symmetric logarithmoid is continuous and first—order differentiable. It is a monotonically increasing function with maximum sensitivity

*) Abhay B. Bulsari†, and Henrik Saxén
Värmeteknik, Kemisk—tekniska fakulteten
Åbo Akademi
SF 20500 Turku, Finland

† on leave from Lappeenranta University of Technology

near zero and monotonically decreasing sensitivity away from zero, as with the sigmoid. However, the symmetric logarithmoid never becomes insensitive to the argument, and its output is not limited to between −1 and 1. Networks using this function are a bit easier to train, and the convergence is better. The derivative of this function can be expressed as a function of its output, and can thus be used in an algorithm like back-propagation also. The other extreme of activation functions is the linear (identity) function, which finds limited use in our work for statistical purposes. This function, obviously poses no problems to the usual optimisation methods, and the Levenberg-Marquardt method, which we use, converges in very few iterations.

2. The biochemical process

Biochemical processes have highly non-linear characteristics, and have operability in limited domains. In the process considered here, *Saccharomyces cerevisiae*, a yeast, is grown on glucose substrate in a chemostat (a biochemical continuous stirred tank reactor) producing ethanol as a product of primary energy metabolism. There are three state variables: microbial concentration, X; substrate concentration, S; and product concentration, P. The kinetic and stoichiometric parameters were taken from a recent study on kinetics of this system [6]. The feed to the chemostat is sterile, *i.e.* there are no microorganisms in the feed. The feed concentration of substrate, glucose, is S_0, and D is the dilution rate (volumetric flow rate per volume of the chemostat.) The dynamics of this system can be described by the following equations.

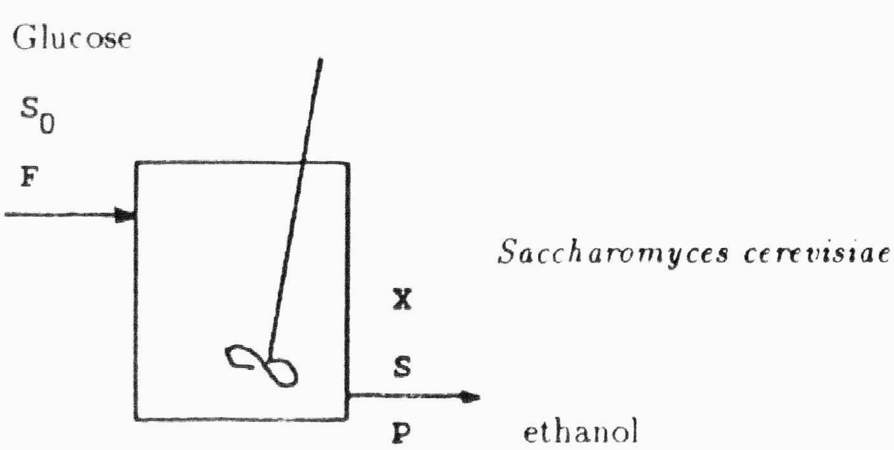


Fig. 1. A bioreactor

$$\frac{dX}{dt} = (\mu - D)X$$

$$\frac{dS}{dt} = D(S_0 - S) - Y_{S/X}\mu X$$

$$\frac{dP}{dt} = -DP + Y_{P/X}\mu X$$

where the growth rate, μ and the yield coefficients, $Y_{X/S}$ and $Y_{P/X}$ are given by

$$\mu = \frac{0.427S}{0.245 + S}(1 - (P/101.6)^{1.95})$$

$$Y_{P/X} = 3.436, \qquad Y_{X/P} = 0.291$$

$$Y_{X/S} = 0.152\,(1 = P/302.3), \qquad Y_{S/X} = 1/Y_{X/P}$$

The growth rate, μ cannot exceed 0.427, and that is also the upper limit for the dilution rate, D. If the dilution rate is higher than the growth rate, microbial concentration in the bioreactor decreases to zero. Table 1 shows typical steady state values for the process parameters.

Table 1. Typical steady state values of the process parameters.

S_0	101.416 gm/lit
D	0,345 hr^{-1}
X	12.0 gm/lit
S	10.0 gm/lit
P	41.232 gm/lit
μ	0,345 hr^{-1}
$Y_{X/S}$	0,1313
$Y_{X/P}$	0.291

3. The Levenberg-Marquardt method

Levenberg-Marquard methodt [2—5] was used to determine the weights in the neural networks by minimising the sum of squares of errors (the difference between the calculated output and the desired output), which is the aim of network training. Back propagation by the generalised delta rule, a kind of a gradient descent method is one popular method for such training. Most algorithms for least-squares optimisation problems use either steepest descent or Taylor-series models. The Levenberg-Marquardt method is a restricted step method, which uses a interpolation between the approaches based on the maximum neighbourhood (a „trust region") in which the truncated Taylor series gives an adequate representation of the non-linear model. The method has been found to be advantageous compared to other methods which use only one of the two approaches.

4. Results

Several simulations were carried out using Simnon [7] to generate the data for the system's dynamics. Results from 16 simulations were used to tabulate the training instances. The few instances which had outputs exceeding 1 were excluded to permit usage of the sigmoid. The total number of training instances was 1562.

The 9 inputs to the networks were the state variables X, S, P at a time t, S_0 and D at times t, $t + 0.5\Delta t$ and $t + \Delta t$. The outputs of the neural networks were the changes in the state variables X, S, P between times $t + \Delta t$ and t. All training instances had outputs between -1 and 1 to permit usage of the sigmoid. Δt was 0.1 hour and the process had a characteristic time of 2.9 hours.

4.1 Training the feed-forward neural networks

A network without hidden layers is not applicable. The minimum sum of squares of errors (SSQ) was found to be about 14 with the sigmoid as well as the symmetric logarithmoid activation functions.

Two nodes in the hidden layer were not sufficient. The SSQ were about 11.5 for both the activation functions. The (9,3,3) network performed much better resulting in SSQ of 1.498 and 2.154 for the symmetric logarithmoid and the sigmoid respectively. It was much easier to get converged results with the symmetric logarithmoid than with the sigmoid. For (9,4,3) and (9,5,3), the SSQ were 0.9487 and 0.3516 respectively, using the symmetric logarithmoid activation function. The rms error for (9,5,3) with 68 weights was 8.66×10^{-3}. A linear correlation between the outputs and the inputs results in a SSQ of 13.56.

Networks with two hidden layers were difficult to train. (9,2,2,3) and (9,3,3,3) had SSQ od 11.971 and 1.762 respectively with symmetric logarithmoid activation functions.

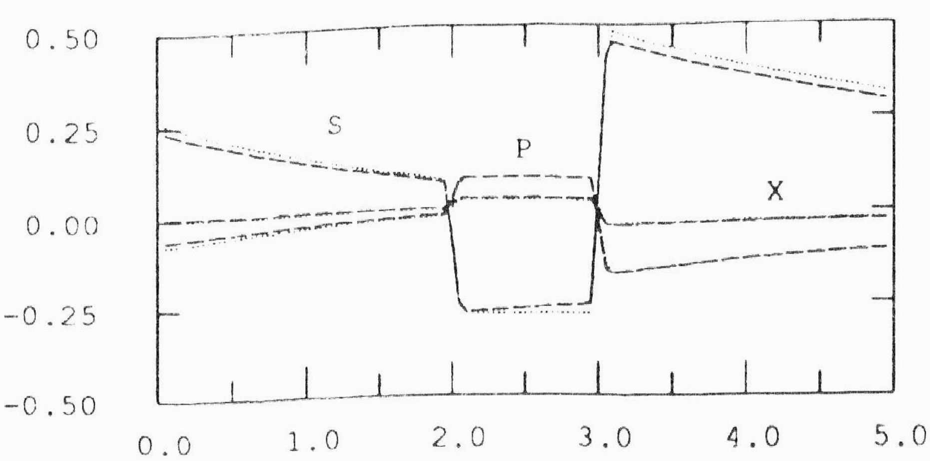


Fig. 2. Comparison of predicted changes in the state variables X, S and P in 0.1 hr.

This trained neural network (9,5,3) was tested on another set of conditions, which it was not trained for. There were two step changes in S_0 and in D in the test conditions. The predicted and actual changes in the state variables are shown in Fig. 2. The prediction is very good throughout. The sum of squares of these errors is 0.04096, and the root mean square error is 0.01174. This small error in the change of state variables is hardly visible on the plot of the state variables, shown in Fig. 3.

It is sometimes better† to use the logarithms of the inputs instead of their actual values, especially if the inputs span more than an order of magnitude. However, it did not improve the results in this case. The SSQ for (9,3,3), (9,4,3) and (9, 5, 3) were 2.103, 0.940 and 0.470 respectively using the symmetric logarithmoids, which are not better than the ones obtained without the logarithmised inputs. It may therefore be concluded that logarithms of inputs help only when the range of inputs is more than a few orders of magnitude.

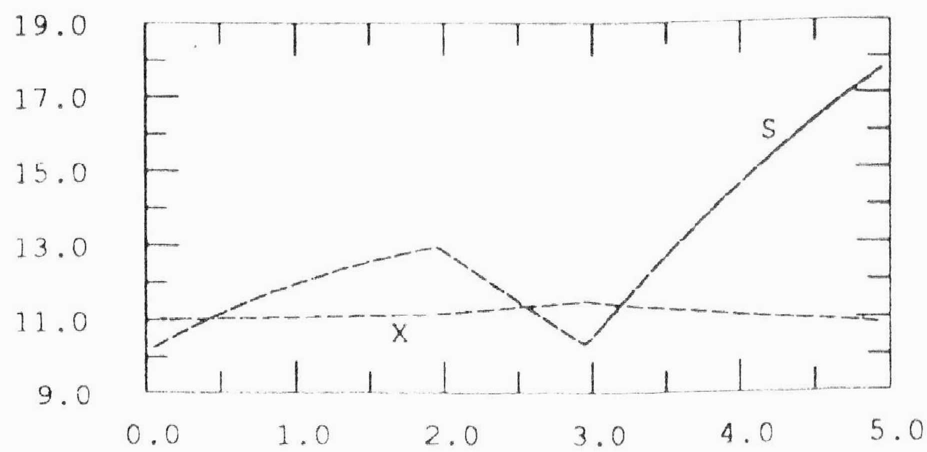


Fig. 3. Comparison of predicted and actual values of the state variables X and S (Differences in P are also not discernible.)

4.2. Dominant weight analysis

Dominant weight analysis is a rationalisation of the nature of dominant weights. This method is not applicable in a straightforward manner when there are no singular dominant weights. Its applicability is limited to monotonously increasing or decreasing activation functions, and is thus not applicable to activation functions like the radial basis (normal distribution, or the derivative of the sigmoid) functions. For a network without hidden layers, every weight is a dominant weight. For single hidden layer networks, identify the hidden node with the largest weight in magnitude from the first output, if the second largest weight in magnitude is substantially smaller than the largest weight. The sign of the product of this weight with the weights from that node to the input nodes indicates whether the relationship between corresponding input node and the first output is positive or negative. For more hidden layers, identify the dominant weights for all the hidden layers to the layer above, and take the product all those weights with the weights from that node in the first hidden layer to the input nodes. The sign of the product indicates whether the relationship between corresponding input node and the first output is positive or negative. This can be done for all the outputs, and should work for all configurations of the networks.

In this chemostat system, the first output ΔX hat a negative relationship with D; the second output ΔS has a positive relation with S_0 and D, and negative with X and S; and the third output ΔP has a negative relation with D and P, and a positive relation with X.

This could be confirmed with (9,3,3) and (9,5,3). (9,4,3) did not have dominant weights for any of the outputs, but considering that both the large weights were of the same sign for all the three outputs, products were taken for both the "quasi-dominant"

† as experienced with the CSTR system identification problem in [1], unreported

weights and added. The resultant explained the expected nature of the relations between the output and input variables. (Similarly, dominant weights from the hidden layer to the inputs should also be identified to consider if they could influence the upper layers.) One more modification was needed with this system. We have three values for S_0 and for D. Only one representative node for S_0 and one for D were considered. These were the middle nodes among the three, representing averages of S_0 and D during that Δt.

The fault tolerance of the neural networks, which is being considered as a desirable or useful by-product of their architectures, is limited to the non-dominant weights. If one of the nodes with dominant weights is removed, the network cannot function properly. The fault tolerance is very good when nodes with very small weights are removed. This technique can be used to trim a network. However, those small weights are by no means useless. They perform effective curve-fitting in small regions by small amounts.

A corollary of this analysis is that replacing the weights by their multiplicative inverse (-1 times the weights) except the biases on the output nodes and second, fourth, etc. hidden layers, leaves the outputs unchanged for networks with an odd number of hidden layers.

5. Conclusions

System identification of the biochemical system could be performed successfully using the symmetric logarithmoid activation function in feed-forward networks, and the results were often better than with the sigmoid. Convergence during training was faster than is usually encountered with sigmoids. The weights generated after training were never very large, although that happens often with sigmoids.

The symmetric logarithmoid, thus provides a feasible activation function for the neurons, when the outputs are not in a well-defined limited range. It may not be correct to consider this function as an alternative to the sigmoid. The sigmoid is applicable when the output is in limited ranges, and this function is applicable when the output is not limited, and hence their areas of application do not really overlap.

References

[1] Bulsari, A. and H. Saxén,
"Applicability of an artificial neural network as a simulator for a chemical process", Proceedings of the fifth International Symposium on Computer
and Information Sciences, Nevsehir, Turkey (October 1990) 143—151.

[2] Levenberg, K.,
"A method for the solution of certain nonlinear problems in least squares", *Quart. Appl. Math.*, 2 (1944) 164—168.

[3] Marquardt, D. W.,
"An algorithm for least-squares estimation of nonlinear parameters", *J. Soc. Indust. Appl. Math.*, 11 (June 1963) 431—441.

[4] Fletcher, R.,
"Practical methods of optimization, Vol. 1, Unconstrained optimization", John Wiley and Sons, Chichester (1980) 82—88.

[5] Gill, P. E., W. Murray and M. H. Wright,
"Practical optimization",
Academic Press, London (1981) 136—140.

[6] Warren, R. K., G. A. Hill and D. G. Macdonald,
"Improved bioreaction kinetics for the simulation of continuous ethanol fermentation by *Saccharomyces cerevisiae*",
Biotechnology Progress, 6 (1990) 319—325.

[7] Elmqvist, H. *et al.*,
"Simnon user's guide for MS-DOS computers, Version 3"
SSPA Systems, Göteborg (January 1990).

Book Review

Forrest Stephanie: Parallelism and Programming in Classifier Systems.
Research Notes in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, California 1991, pp. 213.

The book is based on the concept of general-purpose learning algorithms applied to parallel rule-based systems, as proposed by J.H.Holland.

It is an updated version of the author's 1985 PhD thesis, extended by broad discussion on parallelism, intelligent systems and emergent computation. It is intended both for graduate students in AI and cognition theory and for specialists in the classifier systems, which represent an alternative model to connectionism (such as neural networks) from one hand and to knowledge representation systems on the other hand, but unlike them they operate on a symbolic/sub-symbolic level.

The schema of the system is the usual one: performance system, which is a rule based, message passing and highly parallel system of classifiers [over the alphabet 0,1,], credit assignment level, based on bucket brigade algorithm and the rule discovery level using a genetic algorithm.

The first part of the booklet [chapters 1,2 and 3] may serve as an introduction to all above mentioned components including a brief description of the KL-ONE family of languages (with a good source of references). What is to be appreciated is a nice discussion on parallelism, which can be of benefit to a broad range of readers.

The second part is more specialized and requires familiarity with KL-ONE and patience to swallow all of the many formulas. On the other hand, general remarks are pleasing to read by anyone. This part brings some implementation details, study of various complexity measures of parallel KL-ONE operations and examples ending with 50 pages of Most Specific Subsumer and, above all, a theoretically supported general compiler for producing classifiers implementing any semantic net specified by a KL-ONE expression.

The book is provided by a well selected bibliography [but no subject index]. The formulas are typed in two fonts, which contributes to the clarity, but needs good eyes or glasses.

Jiří Hořejš

Literature Survey

Kuffler S. W., Nicholls J. G.: From Neuron to Brain Massachusetts, Sinauer Associates, Inc., Sunderland, 1977

Kufudaki O., Hořejš Húsek D.: Threshold-Controlled backpropagation Learning In:Proceedings of the International Symposium on Neural Networks and Neural Computing — NEURONET'90,Prague, September 10-14, , 1990 pp.211-213

Abstract: Using geometrical ideas outlined previously we propose a learning strategy for layered networks which turned out to be fairly efficient when when compared with classical forms of backpropagation; the strategy helps considerably also when by passes [connections skipping the hidden layer] are used. The considerations concern the hidden neuron problem as well.

Lansner A.: Associative Processing in Brain Theory and Artificial Intelligence In: TRITA-NA-8505, Dept. of Numerical Analysis and Computing Science, Stockholm, Sweden, Held:, The Royal Institute of Technology, 1985

Abstract: In this paper some theories of associative brain functions are reviewed and elaborated further on. A distributed and self-organizing (learning) computational model and associative net, is proposed.

Lansner A.: Prototype Extraction and Matching in Associative Nets In: TRITA-NA-8516, Dept. of Numerical Analysis and Computing Science, Stockholm, Sweden, Held:, The Royal Institute of Technology, 1985

Lansner A., Ekeberg O.: A Program Package Implementing a System of Interacting Adaptive Associative Networks In: TRITA-NA-8302, Dept. of Numerical Analysis, Stockholm, Sweden, Held:, Royal Inst. of Technology, 1983

Lansner A., Ekeberg O.: Reliability and Speed of Recall in an Associative Network, IEEE — Transactions on Pattern Analysis and Machine Intelligence Vol. PAMI-7, 1985 No. 4 pp. 490-498

Legendy C. R.: On the Scheme by Which the Human Brain Stores Information, Math. Biosciences Vol. 1, 1967 pp. 555-597

Abstract: The assumption is explored that the effects of information on the brain is to cause the neurons to form groups. Each group can „ignite" to reverberatory firing in response to a certain kind of stimulation, similar to that which originally formed it.

Luttrell S. P.: Derivation of a Class o Training Algorithms, IEEE Transactions on Neural Networks Vol. 1, 1990 No. 2 pp. 229-232

Abstract: This paper presents a novel derivation of Kohonen's topographic mapping training algorithm, based upon an extension of the Linde-Buzo-Gray (LBG) algorithm for vector quantizer design. A vector quantizer is designed by minimizing an L with subscript 2 construction distortion measure, including an additional contribution from the effect of code noise which corrupts the output of the vector quantizer.

MacKay D. J. C., Miller K. D.: Analysis of Linsker's Simulations of Hebbian Rules, Neural Computation Vol. 2, 1990 No. 2 pp. 173-187

Abstract: Linsker has reported the development of center-surround receptive fields and oriented receptive fields in simulations of Hebb-type equation in a linear network. The dynamics of the covariance matrix of cell activities. Analytic and computational results for Linsker's covariance matrices, and some general theorems, lead to an explanation of the emergence of center-surround and certain oriented structures. We estimate criteria for the parameter regime in which center-surround structures emerge.

Mallot H. A., Seelen W. von, Giannakopoulos F.: Neural Mapping and Space-Variant Image Processing, Neural Networks Vol. 3, 1990 No. 3 pp.245-263

Key words: visual cortex; space-variant image processing; retinotopic mapping; optical flow analysis; uniformity of visual cortex; columnar organization; ocular dominance columns; visual receptive fields.

Abstract: We present a mathematical framework for describing the interaction of neural mappings and local image processing operations which allows functional interpretations. In an example from visual navigation, we show that neural maps are powerful tools for the parallel processing of visual information. Since mappings of the various types result in a spatial encoding of arbitrary stimulus information, image processing operations can be applied to information from other modalities or for higher-level problems as well. For this aspect of neural mapping, we adopt the term parametric mapping.

Malsburg C. v. d.: Goal and Architecture of Neural Computers In: Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds:Eckmiller R., Malsburg Ch.), 1989 pp. 23-28

Abstract: The ultimate goal of neural computers is the construction of flexible robots, based on massively parallel structures and of self-organization. This goal includes construction of a generalized scene. Some of the necessary subgoals have been demonstrated on the bases of neural architecture, but this architecture has to be further developed. Major issues are the reduction of learning times, the integration of subsystems and the introduction of syntactical structure.

Marko H.: The Controlled Information Source and the Communication, In: Stochastische Modelle und Methoden in der Informationstechnik, Nürnberg, April 12-14, 1989

Key words: information theory; communication theory; controlled information source; statistical signal theory; biological cybernetics.

Marom E.: Associative Memory Neural Networks with Concatenated Vectors and Nonzero Diagonal Terms, Neural Networks Vol. 3, 1990 No. 3 pp. 311-318

Key words: neural networks; associative memories; vector concatenation; memory capacity; error correction; nonzero diagonal matrix.

Abstract: The effects of vector concatenation as well as that of eliminating the zero diagonal restriction of associative memory neural network matrices is analyzed. Extensive computer simulations seem to indicate that the use of concatenated vectors increases the storage capacity of the association matrices, beyond the accepted practical limit of 0.14 N, N being the length of the stored vectors.

Matsumoto T., Chua L. O., Furukawa R.: CNN Cloning Template: Hole-Filler, IEEE Transactions on Circuits and Systems Vol. 37, 1990, No. 5, pp. 635-638

Abstract: A CNN template for hole-filling is reported.

Matsumoto T., Chua L. O., Suzuki H.: CNN Cloning Template: Connected Component Detector, IEEE Transactions on Circuits and Systems, Vol. 37, 1990, No. 5, pp. 633-635

Abstract: A CNN template for connected component detection is reported.Using this template a handwritten character recognition system is proposed.An initial test result already shows 94-100. recognition rates for numerals.

Matsumoto T., Chua L. O., Yokohama T.: Image Thinning with a Cellular Neural Network, IEEE Transactions on Circuits and Systems Vol. 37, 1990 No. 5 pp. 638-640

Abstract: Image thinning can be achieved in real time using a cellular neural network with eight planes, each one defined by a set of „peeling templates", and a set of "stopping templates".

Mead C., Ismail M. (Eds.): Analog VLSI Implementation of Neural Systems Boston, Kluwer Academic, 1989, 248 p. ISBN 0-7923-9040-7

Moore W. R.: Conventional Fault-Tolerance and Neural Computers In: Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds:Eckmiller R., Malsburg Ch.), 1989 pp. 29-38

Abstract: Fault-tolerance is used in conventional computer systems and in VLSI circuits in order to fulfill reliability, dependability and/or cost of manufacture objectives. A wide range of techniques have been used according to the particular objectives and the system architecture. Almost all of these techniques can be observed in biological neural networks and may even be in use simultaneously. This paper suggests that VLSI designers may wish to incorporate several of these approaches into digital neural computers.

Murch A. R., Bates R. H. T.: Colored Noise Generation Through Deterministic Chaos, IEEE Transactions on Circuits and Systems Vol. 37, 1990 No. 5 pp. 608-613

Abstract: The apparently stochastic properties of deterministic-chaos sequences, generated by hierarchies of nonlinear recursive loops, are illustrated,as is the capability of such sequences to exhibit seemingly arbitrary spectral coloring. An approximate, though usefully accurate approach to estimating probability density functions of sequences is developed. The synthesis of sequences having prescribed statistics is examined. The tendency of power spectra of sequences to exhibit an increasingly excess low frequency character, as the number of loops in a hierarchy is increased is illustrated. New insights are presented into a)the generation of 1/f noise and b)whether any species of noise need have a random origin.

Negrini R., Sami M. G., Scarabottolo N., Stefanelli R.: Fault-Tolerance in Imaging-Oriented Systolic Arrays In:Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Düsseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds: Eckmiller R., Malsburg Ch.), 1989 pp. 39-50

Abstract: Image processing often involves convolutions and Fourrier Transforms (DFT and FFT): these specific operations are well implemented by means of a systolic multipipeline structure. Practical implementations require large pipelines, adopting highly integrated circuits that are prone to production defects and run-time faults; efficient fault-tolerance through reconfiguration is then required. Still, the basic problem of concurrent (or semi-concurrent) testing must be solved prior to any reconfiguration step. Here, we prove how these structures allow to perform testing by a simple technique (based on the classical LSSD method) so that added circuits required due to testing functions is kept very limited.

Norton M. J.: Motion Control of an Unmanned Undersea Vehicle Using a Simple Neural Network, Intelligent Systems Review Vol. 1, 1989 No. 4 pp. 39-50

Abstract: The kinematic motion control of multiple simulated autonomous vehicles is described. This control is accomplished via the use of a simple neural network which accepts range, bearing, and heading information to produce a speed and rudder angle. The controller is explained and used to show several kinds of behavior in two autonomous vehicles (UUV and Intelligent Torpedo). Early results are presented in a sample UUV combat mission. These results are discussed and examined for problems. Solutions to these problems are suggested and will be incorporated into future work.

Occhinegro P., Tagliaferri R.: Classes of Efficiently Computable Linear Neural Nets, Neural Networks Vol. 3, 1990 No. 3 pp. 347-353

Key words: threshold neural nets; stable and cyclic states; inverse problem; associative memories; exact solution; analytical approach.

Abstract: The synthesis of a linear Boolean net is still an open problem in the study of neural nets.The aim of this paper is to introduce a class for which the solution is possible in polynomial computational time.In this class,by considering a neural net of n elements,once the evolution of n independent states is fixed a priori,we compute the algorithms to construct the synaptic matrix and the evolution of the remaining 2 to n-th — n states.These algorithms work in polynomial computational time;the class contains 2 to n-th to2-nd matrices.The main result is the proof that for the class under examination the complete evolution of the net is fixed by an n x n permutation matrix Po which connects the n independent states with their next successors. This reduces the time needed to study an arbitrary linear Boolean net from exponential to polynomial in the number of neurons.

Petsche T., Dickinson B. W.: Trellis Codes, Receptive Fields, and Fault Tolerant, Self-Repairing Neural Networks , IEEE Transactions on Neural Networks Vol. 1, 1990 No. 2 pp. 154-166

Abstract: This paper explores some of the relationships between neural networks and trellis or convolutional codes that lead to fault tolerant behavior. A neural network patterned after the trellis graph description of convolutional codes is able to detect and correct errors in its inputs in a well-behaved way. This network incorporates learning to add failure tolerance capability, and it is able to modify its connection weights and internal representation so that spare neurons can replace neurons which fail. A brief review of trellis coding concepts is included.

INVARIANT SPEECH PERCEPTION AND RECOGNITION BY NEURAL NETS

V. Chudý), L. Chudý**), V. Hapák***)*

Abstract:

In the paper is described one of the possible approaches to speech recognition in the family of modern Indo — european languages, esspecially in Slovak. The model describes the processes of perception by an invariant feature neural net and the learning and recognition by the probabalistic neural net. The model requires maximum 256 bits for a pattern of any word what corresponds to compression of the information rate from some 2^{16} bits per second to 2^8 bits per second for an isolated word. The aim of the model is recognition which is independent (invariant) of a speaker, and the redundant physical and phonetic parameters. An underlying Group symmetry approach is implicitly involved in the model.

1. Introduction

Despite the fact that from the 1960's it was becoming clear that the final goal of speaker independent connected speech recognition is one of the hardest recognition problems, work in this field has been continued by investigation of the particular speech recognition tasks. In the 1980's were applied three different concepts to deal with these tasks. The first two were strongly statistical ones:

Dynamic Time Warping and the Hidden Markov Model. The last one is connected with the re-emergence of the Artificial Neural Networks (ANN). In spite of the great potential benefits of ANN, which are based on strong motivation by the nature-like approaches and solutions, the mainstream of the ANN speech recognition investigations still has more engineering-like fashion.

The usual strategy here is the following one:

1. to apply some standard acoustic preprocessing, which transforms or encodes the speech signal into some coefficients: spectral, cepstral, LPC etc.
2. to use some learning and recognition methods for the classification of the obtained coefficient representation of speech sound.

*) Dr. V. Chudý,
Dept. of Phonetics, Philosophical Faculty of Comenius University, Gondova 2, 818 01, Bratislava, Czechoslovakia
**) Dr. L. Chudý,
Institute of Measurement Science, Slovak Academy of Sciences, Dúbravská cesta 9, 842 09, Bratislava, Slovakia, CSFR
***) Dr. V. Hapák,
Dept. of Phonetics, Philosophical Faculty of Comenius University, Gondova 2, 818 01, Bratislava, Czechoslovakia

The main effort is oriented solely in the investigation and improvement of the learning and recognition methods.

Conversely relatively very little attention is focused on the signal preprocessing stage. It is known, however, that the biological auditory system is sensitive not only to the spectral representations of speech but also to various transient features in the time domain, which are responsible for the highly nonlinear dynamical processing of the auditory system. Widespread opinion is that for the time being this nonlinear preprocessing is not understood well enough to warrant the design of artificial speech preprocessing methods or standard electronic analogies.

The approach proposed in this paper is just focused on the understanding and design of the biologically motivated and plausible preprocessing stage of speech recognition. We think that the key for further deeper understanding of the human speech recognition lies in this field. However, we investigate here also the subsequent learning and recognition methods.

Our speech processing approach is crucially based on the invariant feature perception principles. These are incorporated in a fundamental way in the construction of the processing mechanisms, whereas invariant features (which correspond to words or phonemes) emerge naturally. As a consequence the relevance and significance of the Group symmetry approach naturally arises and is involved in our model.

The proposed approach can be considered as the ANN model of speech perception and recognition, but also as the design of an artificial speech recognition system.

The paper is organized as follows:

In Sect. 2 we introduce the concept based on the in variant feature principles. We describe the model of the phonetic preprocessor of the neural network form. We discuss this model from the point of view of the isolated word recognition in time domain.
In Sect 3 are introduced and discussed the learning and recognition methods by a probabilistic neural net related to the above task.
In Sect. 4 we discuss the results of computer simulations on the real speech data.
Sect. 5 contains some discussion and comments about human perception in the comparison with that of parrots.

Proposals of the more human-like perception and recognition methods are discussed.

Some technicalities and fundamental concept proposals related to the general field theory approach are deferred to the Appendix.

In this paper we prefer a physically oriented treatment to a mathematically rigorous one.

2. Neural Phonetic preprocessor — NP 4

In the first part of this section we want to make some general remarks about speech perception from the point of view of abstract ideas of information processing and about the role of invariant feature recognition mechanisms. The aim is to see the motivation and background of our argumentation in the proposed construction of the speech perception model.

In the second part we mathematically describe in detail particular processing stages of the neural perception phonetic preprocessor model. In the following text we will address this model shortly as NP 4.

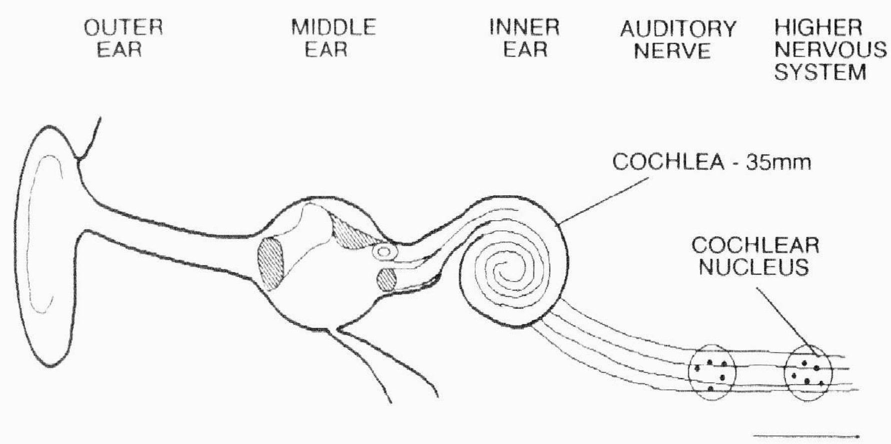


Figure 1

We now present a short remark about the information processing aspect of human speech recognition. From the point of view of information processing the human speech perception and recognition mechanism can be considered as a special device which

> *compresses the information rate from about 2^{16} bits/s to 2^{8} bits/s, for isolated words.*

These estimations follow simply from the following expressions for the information rate in analog or digital form

$$\frac{\Delta I}{\Delta t} \sim BW \log_2 \{1 + S/N\} \text{ for the analog form,}$$

$$\frac{\Delta I}{\Delta t} \sim 2BW \log_2 \{2^{A/D}\} \text{ for the digital form,}$$

where $\Delta I/\Delta t$ is the information rate (bits/s); BW is the frequency bandwidth of the input signal (Hz); S/N is the signal to noise ratio of the speech signal and A/D is the number of bits used for analog to digital conversion.

The upper limit is the input information rate which is transferred from the outer space to the ear via sound waves. For this outer speech signal this gives about 2^{16} bits per second.

The lower limit follows from the next considerations. When we want to describe a phoneme-like unit in terms of binary distinctive features (see [4]), we need for this about six bits (whereas the number of phonemes is about 40). We then have about 10 phoneme-like units for a single word one second long. This approximately leads to the information rate of about 2^8 bits per second. At this point we want to note that our model final representation of the word is a 32 dimensional vector of 1-byte numbers, what is just the reguired information rate 2^8.

Thus we can conclude that the speech signal has a relatively high redundancy, so when designing the recognition system we must be sure that the compression of the information rate does not change in a considerable way any of the essential phonetic properties of the speech signal, which are necessary for the recognition of the phonemes, words, or continuous speech. In practice it means that the artificial recognition system is as effective and optimal as is the value of the obtained compression information rate of the original speech sound to the final bit representation of the phoneme or word.

Our approach is based on the conviction that the crucial role in this process is played by the invariant feature mechanisms. In the next section we therefore introduce some basic facts about invariant speech recognition.

2.2 Invariant feature recognition requirements

In speech perception and recognition it is possible and useful to take into account two kinds of invariances. The first of these deals with the configuration space or outer space, space-time, frequency, intensity, etc.

Invariances *under*

- *the intensity of speech signal*
- *the acoustical and phonetic noise*
- *the pitch of the speech signal*
- *the rapidity of speaking and the total time duration of a word*
- *the relative location and motion of the observing system to the source of speech signal*

are called the outer ones of the recognition mechanism.

All these invariances we use in every-day life and they are enormously important for effective and robust speech communication. On the more fundamental level they point to the possible internal mechanisms of speech perception and recognition which can be used in the design of the recognition system. That is analogous to the situation when from the invariance under group symmetry transformations we can construct differential equations (invariant under these transformations) [15].

The proposed NP4 model in the next section can be considered just as the invariant (under above listed transformations) feature extraction mechanism.

The second kind of invariances are the so called inner ones and are reflected in the symmetries of the phonemic system (what will be discussed in the Discussion). At the present stage of the investigations we have not incorporated these inner invariances in the construction of the proposed NP4 model. However, we consider it incorporation in the model as crucially important and in final effect responsible for the human-like speech perception and recognition (see again Discussion and also Appendix).

And finally we require a feature of beauty, which means to be simple, compact, small and realizable in a simple way in analog or digital hardware fashion.

We presume the languages satisfying the invariance requirements to be almost all Indo — european languages.

2.3 The NP4 model

To design our invariant feature system we strongly follow the physiological mechanisms of speech perception. So every physiological part making an important operation with the speech signal (illustrated in *fig. 1*) corresponds roughly to the subsequent part of NP4 recognition system. Thus we can construct the NP4 schematically according to *fig. 2*, which follows the sketch in *fig. 1*. In the upper line of this figure we have denoted physiologically significant parts artificial processing analogies of which are in the middle line. In the lower line are presented the information rates of the individual levels of the processing.

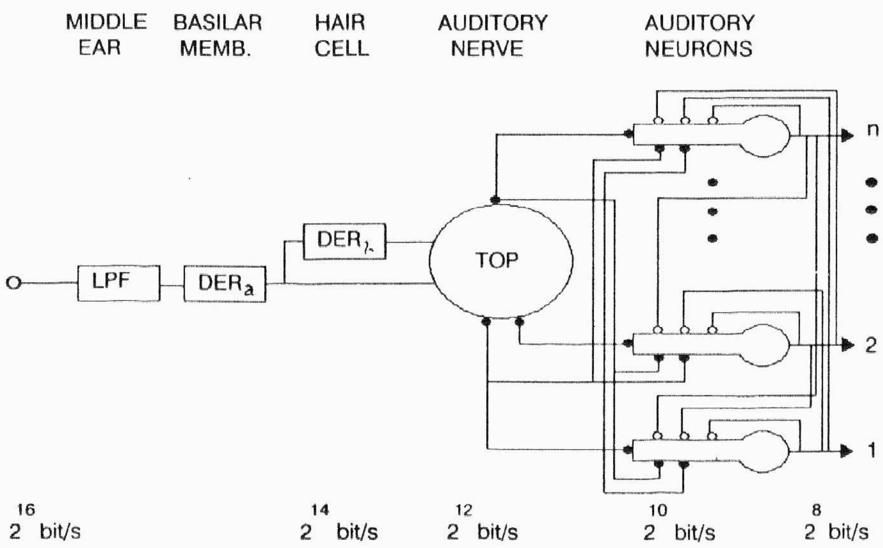


Figure 2

Here it must be pointed out however, that the physiological resemblance of the proposed model is a qualitative one, of course. Moreover, the NP4 model as a preliminary model does not include frequency analysis processing (as it appears in human chochlea). That is the main reason for considering the proposed model as a more parrot-like perception model. We will discuss this topic in detail in Sect. 5, where there are proposed more consistent and elaborate approaches for modeling of human-like speech perception and recognition. A hypothesis of a more general field theory concept of this topic is introduced in the Appendix.

In what follows we will not deal in detail with the physiological background of the particular processes. More information related to our model can be found in [1, 2, 16, 17]. In this section we do not introduce any concrete values of parameters in a question. This is done in Sect. 4 where we will present the numerical results on the real speech data.

Now we mathematically describe the particular parts of the NP4 model.

1. Let us consider the input speech signal s(t) as a function of time.The first part of the NP4 is low-pass filter

$$s_1(t) = \int h_1(t, t')\, s(t')\, dt', \qquad (2.1)$$

where $h_1(t, t')$ is the impulse response of the low-pass filter. This impulse response must be linear, stationary and its phase characteristic must be linear, too. The system realization of this filter is the so called non-recursive filter. Output from this filter is $s_1(t)$.

This part models the function of the middle ear.

2. The following part describes the function of the basilar membrane. We model this function in a very approximative fashion, simply by the certain approximation of the first derivative of $s_1(t)$

$$s_2(t) = \int h_a(t, t')s_1(t')\, dt' \sim (ds_1 t)/dt. \qquad (2.2)$$

The requirements of the impulse response $h_2(t, t')$ are similar to the previous ones. In simulations we have chosen a physically realizable, causal response, discrete version of which is given in Sect. 4. The frequency characteristic is illustrated in *fig. 6.*, which has similar properties as the measured ones of basilar membrane (with the exception of the presence of unimportant nonlinear effects).

3. The third part describes exactly the functioning of one hair cell, which has two types of outputs. The first one $x(t)$ copies exactly the output of the basilar membrane potential $s_2(t)$. The second output $y(t)$ makes approximately the derivative of this potential

$$x(t) = s_2(t) = \int dt'' \int h_2(t, t')h_1(t', t'')s(t'')dt',$$

$$y(t) = \int dt''' \int dt'' \int h_2(t, t')h_1(t', t'')(t', t'')$$
$$h_1(t'', t''')s(t''')dt'. \qquad (2.3)$$

4. The effect of the auditory nerve and cochlear nucleus we describe by the nonlinear operator θ which operates on both functions $x(t)$ and $y(t)$ and its outputs are values 1 or 0, only. To understand better the behavior of this operator we construct the so called

phase plane of $x(t)$ and $y(t)$ illustrated in *fig. 3.* We now define $2n$ equally spaced axes each of them beginning in the intersection of coordinate axes. Then we can define the assumed output by a so called elementary spikes generator

$$q_i(t) = \lim_{\delta t \to 0} \Theta[-f_i(t+\delta t) f_i(t)], \quad (2.4a)$$

where Θ is the unit piecewise operator defined as

$$\Theta(a) = \lim_{\delta t \to 0} \frac{-1}{2\pi i} \int \frac{e^{-i\omega a}}{\omega + i\varepsilon} d\omega = \begin{cases} 1 \text{ for } a > 0, \\ 0 \text{ for } a < 0, \end{cases}$$

here $i = (-1)^{1/2}$, the time δt can be interpreted as the refraction time of the nerve and in the digital model it is equal to the sampling period. Variable $f_i(t)$ is defined by

$$f_i(t) \operatorname{arctg} \left(\frac{x(t) - \alpha_{2i} y(y)}{y(t) - \alpha_{2i} x(t)} \right) \quad \text{for } i = 1, \ldots n,$$

where $x(t)$, $y(t)$ are given in (2.3) and arctg (α_{2i}) is the angle of a $2i$-th axis. We see that the physical meaning of this generator is to generate a unit impulse when the trajectory intersects the given (even only) axis. We will explain the meaning of the factor $2i$ later.

The physically important meaning of the above type of processing is to make some quasi-topological feature extraction, which is invariant to intensity and noise, For a more detailed physical explanation of this we refer to Appendix *A.* However, this meaning is also clear from the comparison of *fig. 3a, b, c.* The number of spikes produced inthese three situations is the same or in other words this operation is invariant under some continuous deformations of the phase trajectory. It is interesting to realize that the above mechanism explicitly models the firing of neurons.

A "cochlear nucleus" has two kinds of outputs. The first one we have just described, the second one deals with the time duration which the system spends in the single sector of the phase plane.These sectors are illustrated in *fig. 3.*

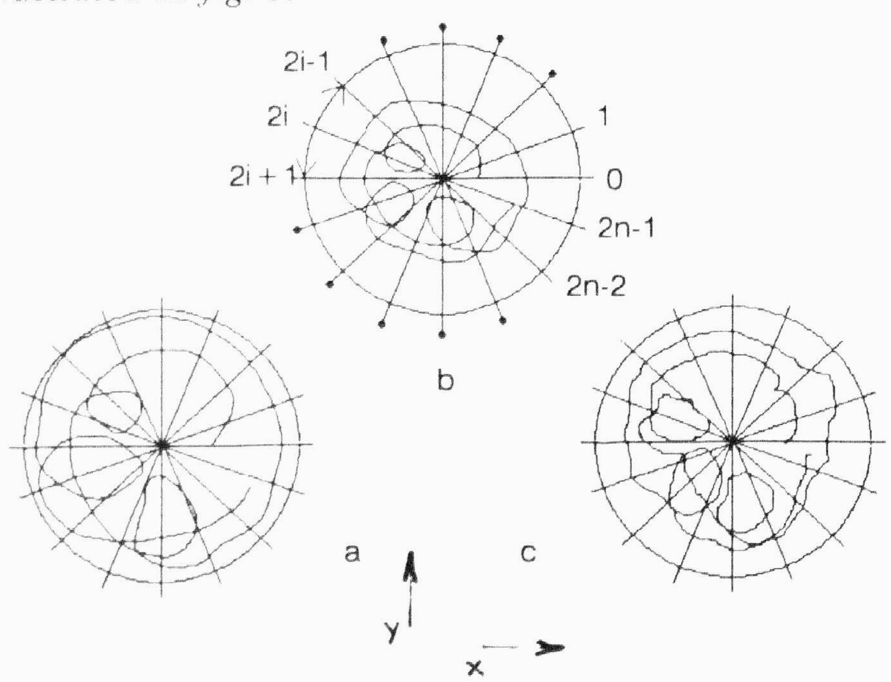


Figure 3

Every (even) axis which is used for the counting of intersections has its sector which is staked out by adjacent (odd) axes. The mathematical treatment of this generator is quite similar to the previous one. We define the so called elementary time generator as

$$m_i(t) = \Theta[-g_i(t) g_{i+1}(t)]. \quad (2.4b)$$

The meaning of Θ is the same as above and the function g_i is defined as

$$g_i(t) = \operatorname{arctg} \left[\frac{x(t) - \alpha_{2i-1} y(t)}{y(t) - \alpha_{2i-1} x(t)} \right], \text{ for } i = 1, \ldots n.$$

The physical meaning of the elementary time generator is to be the weight factor of the spike generator $q_i(t)$ in the input of n neuron-like units as we will see in the next part. The purpose of this weight is to normalize in some sense the time duration and rapidity of speaking. On the more local level it takes into account the relative probability of the input firing of the i-th neuron-like unit relative to the other ones.

5. We will now discuss the fifth part of the NP4 model, which reflects the activity of the network of auditory neurons, which consists of n neuron-like units, with inputs from the cochlear nucleus. In the description of the dynamics of this network we will follow some common ideas from Kohonen, [5]. Thus we describe the activity of n neurons by the following system of dynamic equations

$$d_{\eta_i}/dt = \sum_{j=1}^{n} \varphi_j(\zeta_{ij}) - \rho_i(\eta_i), \quad (2.5)$$

where η is the output firing activity of i-th neuron, ζ_{ij} is the frequency of the firing delivered by some other neuron to the j-th input of the i-th neuron. Functions Φ and ρ have some general form. Assuming that the inverse function ρ exists and the stationary input-output matching condition holds we can obtain

$$\eta_i(t) = \sigma_i[I_i(t) + \gamma_i(t)],$$

where $I_i(t)$ is the input activity of the i-th neuron, γ is the offset value, the hypothetical "threshold", σ is called the sigmoid function, (for more details see Kohonen). As was mentioned by Kohonen the true triggering threshold depends on the collective behavior, interactions of neurons. In this paper we define this threshold function in a similar way to the lateral feedback of Kohonen [5]. So we define the output activity of the i-th neuron — like unit as

$$\eta_i(t + \Delta t) = \sigma_i \left[\Delta I_i(t) + \sum_{j=1}^{n} \Delta s_{ij}(t) \eta_i(t) \right], \quad (2.5a)$$

where one of the possible forms of $\Delta I_i(t)$ may be written as

$$\Delta I_i(t) = \Delta\zeta_i(t) \Delta\mu_i(t), \quad (2.5b)$$

where $\Delta\zeta_i(t)$ and $\Delta\mu_i(t)$ are the integral or sum activities of the i-th elementary spikes and time generators during the time duration Δt defined as

$$\left.\begin{aligned} \Delta\zeta_i(t) &= \int_t^{t+\Delta t} dt\, dq_i/dt \\ \Delta\mu_i(t) &= \int_t^{t+\Delta t} dt\, dm_i/dt \end{aligned}\right\} . \tag{2.5c}$$

In this paper we do not discuss whether the integrals are in the standard form or in some generalized form. The quantities $\Delta s_{ij}(t)$ are defined as

$$\Delta s_{ij}(t) = \frac{\langle \Delta\zeta_i(t)\rangle - \Delta\zeta_j(t)}{\langle \Delta\zeta_i(t)\rangle + \Delta\zeta_j(t)}, \tag{2.5d}$$

where $\langle \zeta_i(t)\rangle$ denotes the back-time mean value of $\zeta_i(t)$, generally weighted. In our case we have then

$$\langle \Delta\zeta_i(t)\rangle = \sum_{k=1} \Delta\zeta_i(t - k\Delta t)\, w(k\Delta t), \tag{2.5e}$$

where the weight function w is generally a nonincreasing function of $k\Delta t$ and expresses some forgetting effects. The basic form of this function is

$$w(k\Delta t) = \Theta[(a-k)\Delta t]/a.$$

The decay time of this function, a Δt can be called the retardation time. In Sect. 4 we shall also show the physical meaning of Δt as the adaptation time of the auditory nerve and neuron-like units. The meaning of Δs_{ij} follows from the following considerations. There exist three limit values of this function

$$\Delta s_{ij}(t) = \begin{cases} 1 & \text{for } \Delta\zeta_j(t) \ll \langle \Delta\zeta_i(t)\rangle \,, \\ 0 & \text{for } \Delta\zeta_j(t) \sim \langle \Delta\zeta_i(t)\rangle \,, \\ -1 & \text{for } \Delta\zeta_j(t) \gg \langle \Delta\zeta_i(t)\rangle \,. \end{cases}$$

We see that this function describes whether something happened in the i-th neuron relatively to the j-th neuron. In this way the neurons "communicate", interact with other neurons and with themselves. The idea of choosing just this function is coming from some abstract ideas of vision perception and recognition. We believe this idea to be rather general and applicable in the neuron-like systems. We have also seen that these interconnections or synapses are not symmetric ones in the indexes i and j.

The sigmoid-like function $\sigma(.)$ in (2.5a) is defined as

$$\sigma(x) = \begin{Bmatrix} 0 & \text{if } x \le 0 \\ x & \text{if } x \ge 0 \end{Bmatrix}$$

As we will see this leads in practice to the $\sigma(x) = x$.

At the end of this section we summarize the main "architecture" features of the NP4 model. These consist of

- 3 frequency filter-like units,
- 1 spikes-like generator unit with n outputs,
- 1 time generator unit with n outputs,
- n neuron-like units of the fully connected neural network in which each neuron-like unit has,
- 1 direct external input from the particular spikes and time generator (see 2.5b),
- $(n-1)$ cross inputs.

The NP4 model network makes analysis of the speech signal in the time domain. It performs information-like processing and not energy-like processing. This processing is differential in time but the result of this processing given by the output activity of neurons is integral in time. The parameters of the model are self adjusted which is very important for the so called generalization problem (see Sect. 4). The optimal number of neuron-like units we will discuss also in Sect. 4. Not mentioned here is the problem of the timing control or the problem of the determination of the beginning and the end of the word. We proposed one possible neural net approach to this problem in [16]. However, in this paper we will not discuss this problem in detail and it will be supposed that the time of beginning and end, t_B resp. t_E, were sufficiently precisely determined.

REMARK

As we have just shown, the behavior and the description of the NP4 net is a little complex but still deterministic. The best indication of this is the possibility to express the whole network by exactly one equation of the analytical form (2.5a). This is consistent with the common deterministic trend in neural net theories (see [5]). The probabilistic ideas are incorporated only in the framework of classical probability theory or classical physics. Thus we can ask what will be happened if we incorporate the ideas of quantum theory, for example the principle of nondistinguishing of neuron states, into already existing neural theories. A more general question can be as follows. Does the quantum theory play any important or even fundamental role in the neuro-business? Or vice-versa. Does the neuro-business play any important or even fundamental role in the quantum theory?

3. The learning and classification method

In the previous section we have shown that the NP4 produces n real values which represent output activities of neural assembly of n neurons. It means that the original speech sound (scalar) signal is finally transformed into the n — component real valued vector $\eta(t) \in R^n$.

Final representation of the single word is then given by evolved neural activity of the assumed assembly at the end of a word given by final vector $x \in R^n$.

At this stage there remain two chronic problems to be solved. The first one is to design the learning mechanism for producing of the prototypes of assumed single words. The second one is to propose an adequate classification mechanism of vector patterns to the one of the prototype class pattern (from the learned prototype pattern set).

However, both mechanisms function in the human simultaneously, although the main learning process takes place in the first years of age. So, adults usually use an effectively elaborated prototype word set, which is adjusted in fine details during time, only. In this paper we deal with the similar situation, when we model both mechanisms in the frame of single neural net structure — Probabilistic Neural Net (PNN). Instead of dealing directly with prototype class patterns belonging to particular key words, this schema works implicitly with complete distribution sets of the prototype patterns (given by sets of pattern realizations belonging to a particular class word).

In the following part of this section we briefly discuss the main features of a PNN learning and classification method which we think to be proper for our problem.

3.1 The probabilistic neural net

Let us assume pattern vectors $\{x^{jk}\}$, where k denotes word class and j is assigned to the particular j-th component output vectors from the NP4 model.

In the framework of the so called Bayesian philosophy we can write for the cost of classifying pattern into class k (assuming the statistical distribution into classes 1, 2, . . ., M)

$$C(x, k) = \sum_{l=1}^{M} P(l \mid x)\, L(k, l), \tag{3.1}$$

where $P(l \mid x)$ is the conditional probability of pattern x belonging to class l and $L(k, l)$ is the unit cost of one decision expressing the miss classification of a pattern x to the class k when it was actually from l. For the equally significant patterns we can write $L(k, l)$ as

$$L(k, l) = 1 - \delta_{kl}. \tag{3.2}$$

The conditional probability $P(l \mid x)$ can be expressed as

$$P(l \mid x) = p(x \mid l)\, P(l), \tag{3.3}$$

where $P(l)$ is the probability that any observed pattern is really belonging to the class l. $A\, p(x \mid l)$ is then the probability distribution of the pattern vectors of the class l. Now following Specht [6], we assume that this probability distribution can be expressed by the multivariate gaussian distribution as

$$P(x \mid l) = \frac{1}{(2\pi)^{n/2}\, \sigma^n} \frac{1}{N_l} \sum_{j=1}^{N_l} \exp\left[-(x - x^{lj})^2 / 2\sigma^2\right], \tag{3.4}$$

which means that we estimate the probability distribution or probability density function from the so called training patterns x^{lj}, n is the dimensionality of the patterns and σ is the "dispersion" of the distribution with the meaning of a smoothing parameter. With the help of this distribution we can define the discriminant function $D_k(x)$ as

$$D_k(x) = -\sum_{l=1}^{M} R(l)\, L(k, l) \frac{1}{N_l} \sum_{j=1}^{N_l} \exp\left[-(x - x^{lj})^2 / 2\sigma^2\right]. \tag{3.5}$$

Because in the classification are important only relative costs we can drop all the factors in (3.5) which are common to all classes. Then (3.5) we can redefine as

$$D_k(x) = -\sum_{l=1}^{M} R(l)\, L(k, l) \frac{1}{N_l} \sum_{j=1}^{N_l} \exp\left[-(xx^{lj} - x^{lj}x^{lj}) / 2\sigma^2\right]. \tag{3.6}$$

The basic case of this discrimination function is defined by the condition (3.2), then we obtain from (3.6)

$$D_k(x) = P(k) \frac{1}{N_l} \sum_{j=1}^{N_l} \exp \left[-(xx^{kj} - x^{kj}x^{kj}) / 2\sigma^2\right] \tag{3.7}$$

Now we can propose the multilayer neural network modeling of the discriminant function (3.7) according to [6], illustrated in *fig. 4*. In this neural net, the so called Probabilistic Neural Net or PNN, the values flow from the input layer to the middle layer. Every middle node is assigned to one training pattern, x^{kj} and the weights from the input layer to this node are equal to the components of this training pattern vector. The weight from the bias-node is proportional to the squared euclidean norm of the particular training pattern. The middle nodes have the nonlinearity of the type

$$\exp\left[(.) / \sigma^2\right].$$

The weights from the middle layer to the summing layer are equal to one and every sum node is assigned to the one of the classes by the way that the every middle node is connected only to the sum node of its class. The weights from the sum node to the first layer of the MAXNET are equal to $P(k)/N_k$ and the MAXNET produces the index of the sum node with maximum value or the index of the recognized class, the word.

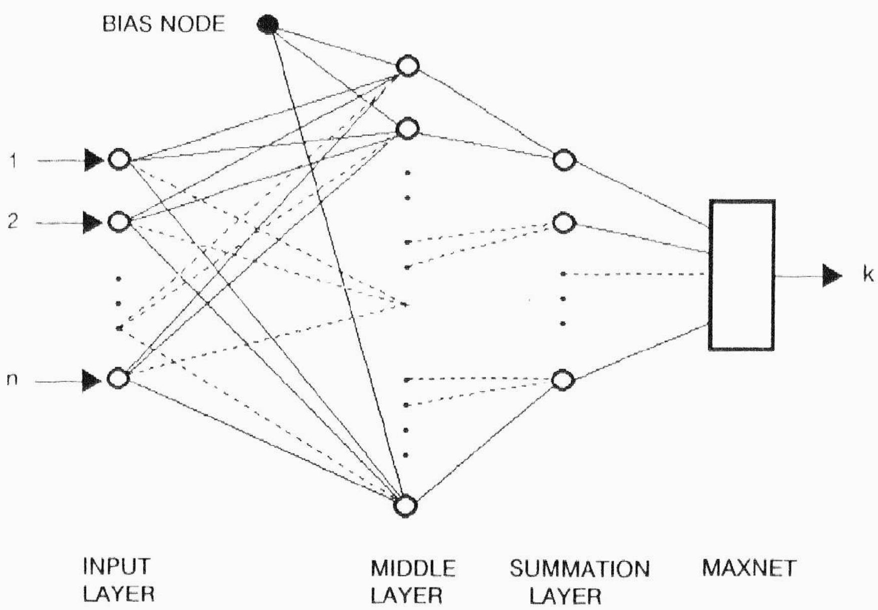


Figure 4

We have just described the learning and recognition of the PNN. It can be seen that the training is easy (without any iterations), instantaneous and can be very easily adopted to the time-varying pattern statistics, [6]. Further, as Specht, [6] has mentioned the PNN can asymptotically approach any probability density function under some general conditions. This can be done simply by the changing of the smoothing parameter σ. The second advantage of this net is the speed gain relative to the back-propagation perceptron (for example in practice this gain is about 200 000 for the same or better performance of the nets in a question).

3.2 Speech Recognition System — NP5

In this section we want to summarize the above proposed approaches into one coherent picture of the unique speech processing and recognition system.

The NP4 model here represents the lower preprocessing stages of the speech signal recognition.

The PNN makes final processing of the complex neuronal activity resulting from the preceding preprocessing and represents high skill level operations such as learning and classification.

Both main modules NP4, PNN create a single artificial neural network speech recognition system, which will be referred in what follows as NP5.

This system can be implemented into any higher structured network system and can function in a small, flexible and compact arrangement in the situations of the segment-like, the whole word-like, and the syntax including processing. The principal system solution of NP5 (NP4 + PNN), is illustrated in *fig. 5*

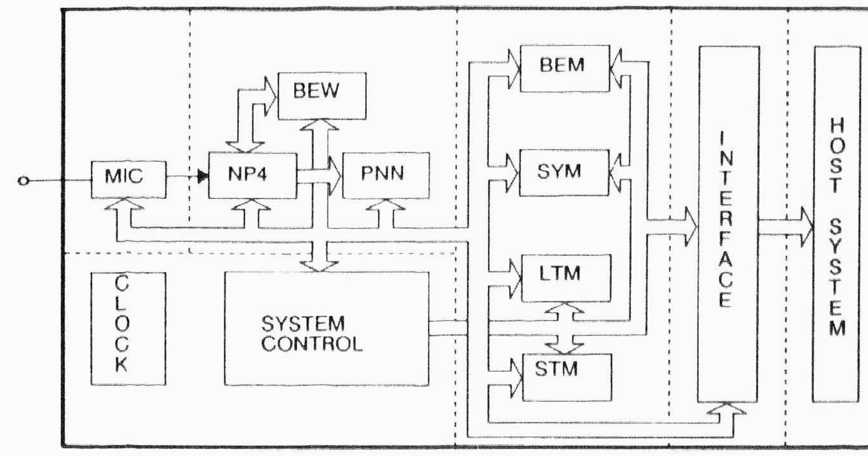


Figure 5

where MIC denotes the microphone input control — it adjusts a maximum of the speech sound intensity. BEM is the pattern memory for the BEW (begin and end of a word time control) processing, SYM is the syntax matrix memory, STM is the short-time pattern memory, in this memory are memorized the "current" patterns. LTM is the long-time pattern memory for the patterns "far" in the past; it is suitable as the preparation mode of the NP5. The remaining notations are clear from *fig. 5*. We believe that the whole system can be realizable on the small European format as the autonomous system or together with the host system.

At the end of this part we briefly discuss an implementation of some syntactic structures into our system. This can be done in two ways. The first one, the so called explicit implementation can be defined by the reinterpretation of (3.7). Thus we postulate the existence of the syntax or lexical matrix $S(k, l)$ of the system of classes, words. Now the interpretation of the first term in (3.7) is the conditional probability of occurrance of the pattern from class k when the previous pattern was classified as belonging to the class l. We postulate that the syntax matrix is proportional to this conditional probability. Then the discriminant function can be redefined as

$$d'_k(x) = S(k, l)\, d_k(x), \tag{3.8}$$

where $d_k(x)$ is defined in (3.7). For this it is necessary to have a knowledge about the artificial or natural syntax of the language system in a question. The second way to do this is the using of the Kohonen self-organizing feature map instead of the PNN. This way we call the implicit syntax implementation. We have done some preliminary experiments with the last one and we consider this approach to be useful.

4. Results of the computer simulations

In the first part of this section we will explain our motivations and goals and will make a description of the experimental conditions. In the second part we present preliminary results on the NP4 and the NP5 models.

The main goal of the following simulation experiments was to investigate the key properties of the proposed model of the speech perception and recognition mechanisms. Especially the testing of the invariance properties (see Sect. 2) of the above mechanisms we consider as the most important and decisive for the design of the optimal and biologically plausible recognition system. For this reason and our time & material limitations we preferred study of a very small set of class words, however recorded in a relatively great number of different realizations. To test the limit condition properties we chose very near class words.

We used 5 words which differ only in one phonetical distinctive feature: lama, lamu, lame, lami, lamo. We used the eight, 5 male and 3 female, speakers. We recorded 500 realizations of these words. Vocabulary of these words was denoted as TTV. We can characterize this set by the following parameters. The maximum of the absolute magnitude of the realizations ranged from about 3 Volts to 8 Volts. The time duration of the realizations in our vocabulary ranged from about 0.2 s to 0.7 s. The maximum of the pitch of the realizations has ranged from about 100 Hz to 350 Hz. The beginning and the end of word-realizations was determined by "hand" for the impossibility of managing this task by some automatic routine on our computer configuration.

The acoustic preprocessor of our system consists of the following stages: a commercial microphone, without any special properties, preamplifier, amplifier, 5 KHz low-pass filter of the 3-rd order with the slope of the frequency characteristic equal to 24 dB/octave, 12-bit analog to digital convertor with 10 KHz sampling frequency. Further parts are described in Sect. 2.

In the numerical simulation we have approximated the derivation by the Stirling formula of the 6-th order

$$x(n) = \{[s_1(n) - s_1(n-6)] - 9[s_1(n-1) - s_1(n-5)] + 45[s_1(n-2) - s_1(n-4)]\}/60,$$

where $s_1(i)$ is the digitized speech signal and $x(n)$ is the Stirling approximation of the derivation with the frequency characteristic shown in *fig. 6*.

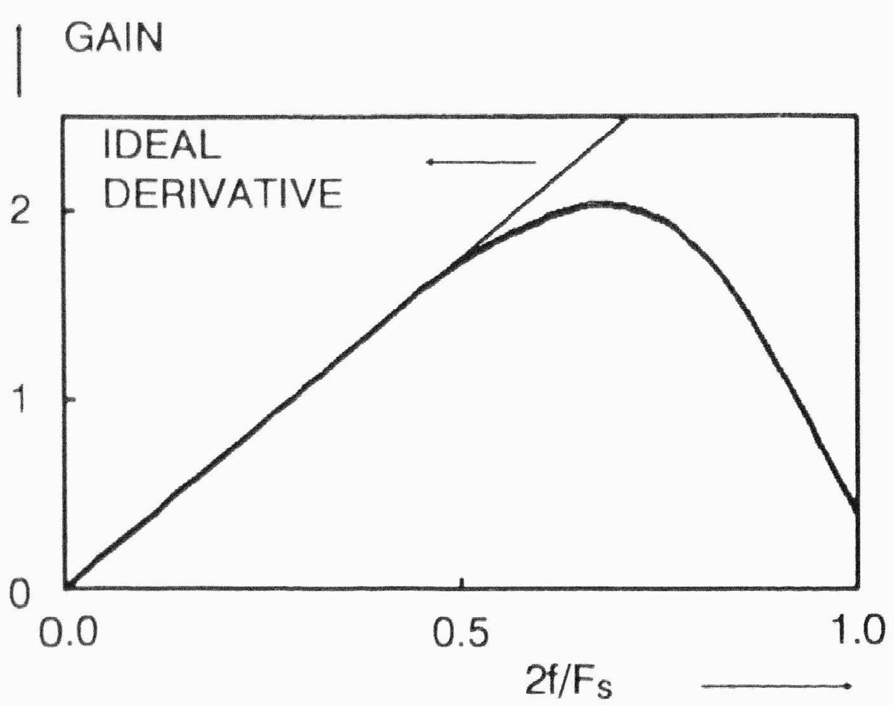


Figure 6

4.1 Word clustering of the NP4 model

The goal of the first experiment was to test the clustering properties of the NP4 model.

We used 100 realizations (20 realizations of every word) of the TTV for this purpose. As a similarity measure we used the euclidean distance

$$d_k(x) = \left(x - x^k\right)^{1/2},$$

where x is the output vector from the NP4 to be tested and the x^k is the mean arithmetic vector of the learning vectors for the k-th class-word. The learning set is the same as the testing one.

We will now show results for different functional forms of the processing in *Tab. 1*.

In the left column we present particular functional forms of the processing, i.e. dynamical rules of the (auditory) neuron network of the type (2.5a).

We used the following parameter setup: the number of neurons is equal to 32, the adaptation time (or the window length, see Sect. 2), is equal to 12.8 ms and the retardation time is equal to 51.2 ms.

In the middle column are matching scores for unnormalized vectors and in the last column for normalized ones (all divided by the total time duration of the given realization). The purpose of this experiment is to investigate experimentally the influence of different factors on the neural network dynamics.

In the first, second and fourth row we present functional forms which have no network-like cooperative behavior (i.e. the dynamics of a particular neuron is not influenced by other neurons). We see from these results that processing without network-like cooperative behavior has also some classification, clustering ability. Comparing the matching scores we see that network-like processing not only "normalizes" vectors according to the time duration of the words, but it makes also some "homotopic" mapping which is local in time. For this compare the matching scores for normalized and unnormalized vectors in the third and the last line.

A remarkable influence has also the weight factor $\Delta\mu_i(t)$ in the external input ΔI_i, which is due to the time duration generator (see third and fourth line).

Tab. 1

TYPE of Processing	Unnormalized	Normalized
$\eta_i(t+\Delta t) = \eta_i(t) + \Delta\mu_i(t)$	30 %	20 %
$\eta_i(t+\Delta t) = \eta_i(t) + \Delta\zeta_i(t)$	40 %	36 %
$\eta_i(t+\Delta t) = \alpha\sum\eta_i(t)\,\Delta s_{ij}(t) + \Delta\zeta_i(t)$	50 %	44 %
$\eta_i(t+\Delta t) = \eta_i(t) + \Delta\zeta_i(t)\,\Delta\mu_i(t)$	60 %	63 %
$\eta_i(t+\Delta t) = \alpha\sum\eta_i(t)\,\Delta s_{ij}(t) + \Delta\zeta_i(t)\,\Delta\mu_i(t)$	70 %	65 %

Our simulation experience shows that the matching scores become saturated for the NP4 with the number of neurons equal to 32. In the case of larger numbers of neurons we assume neurons to be correlated because the matching score for an increasing number of neurons is not increasing. The optimal value of the weight constant "α" is equal to 1/32 which has a very natural interpretation, of the normalization by the number of neurons.

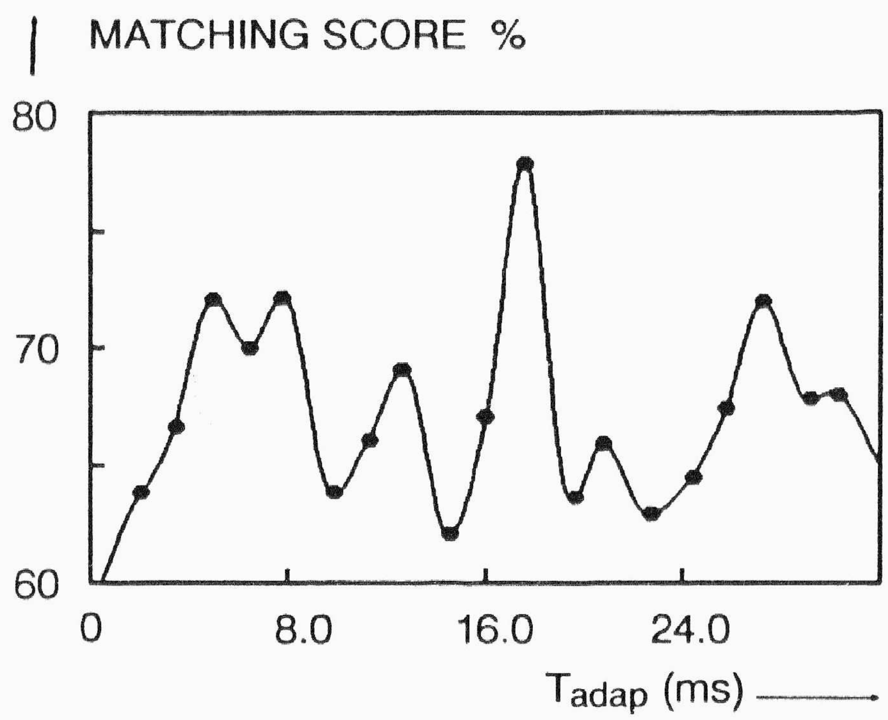


Figure 7

In *fig. 7* we present the adaptation-time dependence of the NP4. We see that the NP4 is functioning for the adaptation time about 1.6 ms, too. The maximum matching score is for the adaptation time equal to 17.6 ms which is in agreement with the experimental facts obtained from a biological measurements, [16]. We think that this adaptation time is optimal only in the statistical sense, what means that during the processing the length of the window is not constant but, for example, a gaussian distributed around this maximum value (17.6 ms). From the simulation results we can conclude that the NP4 does not depend on the retardation time. Our experience with the NP4 tells us that for the generalization task (a transition from small set of words to a larger one) the optimal parameterization of the processing is in choosing of processing with no many fitting parameters (number of neurons, refractation and retardation time we do not consider to be fitting parameters), is obtained with self-adjusted processing — without any parameters, constants, etc. and this is one of the goals of our approach in this paper.

4.2 Modifications of the network dynamics of the NP4

The proposed NP4 model enables some modifications or refinements which can improve it's clustering abilities.

I. First we want to discuss the effect of assuming nontrivial forms of the "synapses" Δs_{ij}. Up to now we have investigated two forms. The first one is the following trivial form

$$\Delta s_{ij}(t) = \alpha\, \delta_{ij},$$

where α is a constant and δ_{ij} is the kronecker symbol. This form corresponds to the situation when each neuron interacts only with itself, the neurons are independent. The results of simulation of this case are given in *Tab. 1* (first, second and fourth row). The second form (the basic one), which is represented in (2.5d) corresponds to the set of mutually interacting neurons. Again the simulation results are given in *tab. 1* (third and fifth row). Now we introduce the so called smoothed form of the synapses by including the sigmoid transformation

$$\Delta s_{ij} \rightarrow \tanh\,(\Delta s_{ij}/T),$$

where T is the smoothing parameter, "temperature". The dependence of such modified NP4 matching scores on the temperature T is in *Tab. 2*.

Tab. 2

T	10^{-6}	10^{-1}	0.25	0.5	1.0	2.0	4.0	10.0	10^2	10^3
match score %	67	68	69	70	70	71	71	68	68	68

We can see that the smoothing parameter T controls the "discussion" i.e. sharpness of the interaction between the neurons. For low temperatures we have lower matching scores (sharp, black and white discussion) and for high temperatures we have also lower matching scores (soft, colorful discussion). Between these limit values exists some optimal temperature T where the matching score is maximal. In our case the obtained optimal value of T is about 3. In this simulation we used 32 neurons, refraction time was 12.8 ms and retardation time about 51.2 ms.

II. Next we investigate the effect of some frustration of the matrix of synapses. Let us assume the following modification of the synapses (half their number are nulled)

$$\Delta s_{ij} \rightarrow \begin{cases} \Delta s_{ij} & \text{for} \quad j < n/4 \quad \text{or} \quad j \geq 3n/4, \\ 0 & \text{elsewhere,} \end{cases}$$

where n is the number of neurons and Δs is one of the assumed forms (trivial, basic or smoothed). The idea of this particular form came from the physiological measurements (see [16] point *H*, [17]) which show us that the acoustic activity is suppressed during the negative portions of the stimulus signal below the spontaneous rate. The negative part of the stimulus signal corresponds in the NP4 model to the left part of the phase plane, see *Fig. 3*. The simulation results show that this type of frustration does not detoriate the resulting matching scores, which are the same as in the previous cases (with the same set of parameters) i.e.

about 71 %. Thus, from the pragmatic point of view we need only $n^2/2$ of connections instead of n^2.

III. Other natural modification of the network are of the functional form of the output neurons. In Sect. 2 we presented the basic form of these neuron outputs as

$$\sigma(x) = \begin{cases} x & \text{for} \quad x \geq 0, \\ 0 & \text{for} \quad x < 0. \end{cases}$$

We used this functional form also in the previous simulation experiments. Now we introduce a sigmoid-like functional form

$$\sigma(x) = \frac{\sigma_0}{1 + e^{-x/\tau}}$$

where τ is the smoothing parameter and σ_0 is the value of the maximum output activity (in our case equal to 256). In *Tab. 3* are presented matching scores of the above frustrated model, with refraction time 12.8 ms and retardation time about 51.2 ms. We see that similarly as to the simulation of smoothed synapses there exists an optimal value of the smoothing parameter τ for which the matching score is maximal. In our case it means improvement of about 5 % above the obtained minimal matching score.

As in the case of synapses we can also consider the frustrated type of the output activity

$$\sigma(x) = \begin{cases} \dfrac{\sigma_0}{1 + e^{-x/\tau}} & \text{for} \quad j < n/4 \quad \text{or} \quad j \geq 3n/4, \\ 0 & \qquad \text{elsewhere.} \end{cases}$$

For this form we obtained about a 70 % matching score for the optimal value of the smoothing parameter. Again it is to be noticed that in this case we use only $n/2$ neurons i.e. $n^2/4$ synapses.

Tab. 3

τ	112	128	144	160	176	192	208	224	240	256
match score %	68	68	70	69	70	72	73	73	72	72

4.3 Influence of noise on the NP4

Finally we present results of the simulations which reflect the noise invariance of the NP4 model. In this investigation we used the basic form of the synapses and output activities (with 32 non-frustrated neurons), refraction time 12. 8ms and retardation time about 51.2 ms. Effect of the noise we simulated in the following way

$$s(i) = z(i) + a(1 - 2\text{rand})z(i) + b(1 - 2\text{rand})\,2048\,,$$

where $z(i)$ is the original speech signal in digital form, $s(i)$ is noise poluted signal, rand is output of the pseudo — uniform random generator from $<0, 1>$, a is the level of the so called "correlated" noise and b is the level of the "uncorrelated" noise, 2048 is the maximal value of the signal for 12-bit analog to digital convertor. The value of parameter a or b equal to 0.2 corresponds to a 20 % level of noise.

We see that the best matching scores are for the null level of the uncorrelated noise and for 40 % correlated noise, *Tab. 4*. Results also demonstrate that excellent noise invariance of the NP4 performance takes places especially for correlated noise.

Tab. 4

$a. = 0$	0.0	0.2	0.4	0.6	0.8
match score %	69	68	74	70	71

$b. = 6$	0.0	0.2	0.4	0.6	0.8
match score %	70	61	51	54	56

4.4 NP5 recogniion results

Now we will present the results illustrating word recognition abilities of the NP5 (NP4 + PNN) model. We chose randomly the set of 100 words form the TTV set (i.e. 20 realizations for every word). These words were not the same as for the NP4 simulation experiment. We used the remaining words of the TTV for the "learning" of the PNN. The results are introduced in *Tab. 5*. We investigated here only learning and classification properties of the PNN which used the NP4 preprocessing with the two best network dynamics of the preceding experiment (see two last rows of *Tab. 1*), only.

It can be seen that the most complex network dynamics of the network-like form introduced in detail in the preceding section give superior results (see right column).

Presented results in the last row illustrate the generalizing abilities of the NP5, in the assumed case of the training pattern number of 400 and test pattern number of 100.

Tab. 5

TYPE of Processing:
$\eta_i(t + \Delta t) = \eta_i(t) + \Delta\xi_i(t)\Delta\mu_i(t)$ [Left column]
$\eta_i(t + \Delta t) = \alpha \sum \eta_i(t)\Delta s_{ij}(t) + \Delta\xi_i(t)\Delta\mu_i(t)$ (Right column]

Classes:	LAMA		LAMu		LAME		LAMI		LAMO		ALL	
Matching scores (%) TRAIN+TEST:	92	96	97	98	91	93	98	99	90	95	94	96
TEST:	82	88	94	96	58	80	94	96	82	88	82	89

It is obvious that the behavior of the PNN model can be adjusted and optimized especially in two ways. The first one discussed here is the choosing of the optimal lowest number of the realizations of every word class and subsequently the proper selection of the particular realizations. This influence is the topic of our current investigations.

The smoothing parameter σ of the PNN processing was found to give acceptable results for a wide range of values. Presented results are for the experimentally chosen optimal value.

Finally, we want to note that obtained results can be considered as preliminary ones.

5. DISCUSSION OF HUMAN PERCEPTION

In this section we want to discuss some topics we consider crucially important for the further elaboration and improvement of the above proposed speech perception and recognition models.

The first part deals with the general lay-out of the biological and especially speech perception.

The other two parts will discuss the differences between the parrot-like perception and human-like perception and subsequently with the possible approaches for the improvement of the models under consideration.

5.1 The general lay-out of perception

Let us assume that we have two systems with some specific and distinct functions. The first system (A) is observed and the second one (B) observes. Because of this we must presume that these systems are separated at some level. This second assumption is necessary to define both systems but we must take into account some kind of interaction between these systems because an isolated or closed system cannot be observed, [3]. Generally speaking,the system B is constructed to observe some features of the system A which presumes some preknowledge about the system A. Without this knowledge one does not know what to observe. An extreme case of this in speech recognition is the situation where we recognize only what we already know. We can define the perception as some kind of interaction which exists between co-evolving or coupled systems, in contrast to the observation (in the proper sense) when the systems in consideration are sufficiently separated, see *fig. 8*. In this paper we constructed the NP4 model just as the first approximation to the human perception understood as above.

The above outlined picture is in good agreement with the two following basic paradigms which play, in our view, a very important role in all considerations dealing with our topic:

PO — We think the perception phenomenon must be

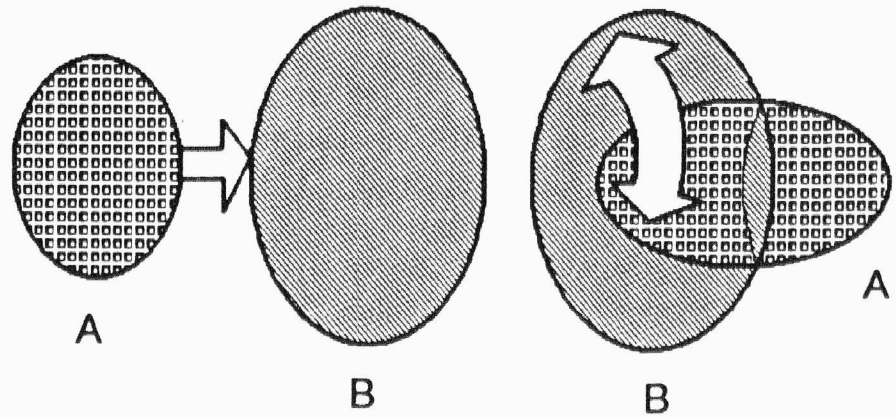


Figure 8

consistent with the three *postulates of Kantor information theory*

1. Conservation of information
2. Communicability of information
3. Finite accessibility of information

which follow from the information uncertainty relation

$$\Delta I \sim \Delta E \, \Delta t / h,$$

where ΔI is an amount of information transfer, Δt is the time duration necessary to transfer an amount of energy ΔE and h is Planck's constant, minimum of action or elementary action, see [3]. We see that transfer of this elementary action leads to the information transfer $\Delta I = 1$.

P1 — *N. Chomsky paradigm:* "Speech is the inborn, congenital biological ability of a human mind."

P2 — *The fundamental paradigm of comparative linguistics:* "All natural languages have the same description capability, the potential to describe the outer, inner world and itself."

Now we briefly discuss these two paradigms. First, we must mention that P1 does not mean that social, cultural and historical background have no influence on the generation of the whole language structure. This paradigm only presumes that at the fundamental level the structure of language is predetermined by a biological one. There are many facts and experimental results which illustrate P1, for example the ability of small children to manipulate symbols or to recognize distinctive features, [7]. The second paradigm deals with the capability of a description and does not relate the actual differences and advantages of the particular language relative to another one. For example, the difference in concepts of time in the English and the Hopi-language, or the geometrical language concepts of the society living in plane-like country versus the concepts of the society living in mountain-like country is not relevant for P2, which reflects a potential for describing anything on the fundamental level.

5.1 The frequency selective NP4

As was shortly mentioned in Sect. 1 the above proposed and investigated NP4 and NP5 models correspond more to the parrot-like perception mechanism. For the purpose of designing the models more consistent with human-like physiology it will be useful to take inspiration from the differences between parrot-like and human-like speech perception.

We know that the existence of a long (35 mm) cochlea for humans is essential for the cochlear frequency analysis processing. But we also know that birds, especially parrots, have some rudimentary cochlea which less than 3 mm. In such a small portion of the parrot cochlea cannot arise travelling waves and no frequency analysis based on these waves is possible. In the parrot ear there is no clear distinction between the inner and the outer hair cells,too. Nevertheless, the frequency range of the parrot ear is about (50—20 000) Hz and the frequency resolution is the same as for the human ear. As we all know, a parrot can produce all human speech sounds in a satisfactory manner and it is also able to imitate pitch intonation with some perfection.

We will now discuss possible extrapolations of the (parrot-like) NP4 model to the human-like model of perception. From our point of view there exist two basic ways how to make this extrapolation. The first one deals with the just mentioned frequency selectivity of the human ear. The second one is principally different to the previous one. This approach deals with the so called inner symmetries of the phonemic system (see Sect. 2). We will discuss these ideas on the conceptual level in the following parts.

We can describe the first way of extrapolation as the explicit frequency selective NP4. In *fig. 6* we postulate the four frequency bands, say; (50—500) Hz, (500—1 500) Hz, (1 500—3 000) Hz and (3 000—5 000) Hz. A larger number of frequency filters is not necessary. For example, if we have used 20 or 100 filters then the outputs from these filters are approximately harmonic with the frequencies of the filters and we practically destroy useful information due to this transformation.

Thus we have 4 outputs from the "basilar membrane" part and the "cochlear nucleus" part. For example, if we use 32 neurons then for every frequency band we will have 8 neurons. It is stressed that these 4-neuron sets are not independent but they communicate, interact among themselves.

The second alternative can be called the implicit frequency selective NP4. We postulate the past-time dependent Fourier transformation of the low-pass filtered signal $s_1(t)$ (see Sect. 2), as

$$F(\omega, t) = \int_{-\infty}^{t} s_1(t')\lambda(t-t')e^{-i\omega t'}dt' = Re[F(\omega, t)] + iIm[F(\omega, t)] , \tag{5.1}$$

where the window function $\lambda(.)$ is there to satisfy convergence of the integral in (5.1). We must mention that (5.1) is not the short-time Fourier transofrmation where the window function has a finite length. We define some smoothing procedure for $F(\omega, t)$

$$G(\omega, t) = \int F(\omega', t)H(\omega, \omega')\,\delta\omega' ,$$

where the real function $H(.)$ is the impulse characteristic in the frequency domain of the smoothing filter. This filter is used to smooth rapidly oscillating components of $F(\omega, t)$. In the discrete time (and frequency) domain $G(\omega, t)$ has about 10 000 harmonics for the sampling frequency of about 10 kHz and the time duration for about 1 second. To compress redundant information in $G(\omega, t)$ we can use some analog to digital convertors with a sampling rate chosen to compress number of frequency components from about 10 000 to about 500. If all system components are physically causal then it can be proved, [8], that $ReG(k, t)$ and $ImG(k, t)$, the smoothed, compressed and discrete versions of (5. 1), are mutually Hilbert transformants. This is sufficient for our requirements of the intensity and noise invariance, mentioned in Sect. 2. Finally, these frequency components will serve as inputs into the module TOP.

In the frame of this approach we have very simply satisfied the requirements of the closing of the trajectory, see Appendix A. The reason for this is simply in the real character of the function $s_1(t)$. Thus the so called Nyquist plot in the frequency domain is symmetric against the real axis and the trajectory of this plot which is given by the real and imaginary part of $G(k, t)$ is the closed trajectory. Further parts of the processing, neural structures, we do not discuss in this paper.

We want also to note that this approach is in some way connected with the phase component relations

$$I(k, t) = (Re^2[G(k, t)] + Im^2[g(k, t)])^{1/2},$$
$$\varphi(k, t) = \mathrm{arctg}(\mathrm{Im}[G(k, t)]/\mathrm{Re}[G(k, t)]) ,$$

where $I(k, t)$ is intensity and $\varphi(k, t)$ is the phase of the given harmonics. Then we see that in this approach will arise very naturally the problem of the phase sensitivity of the ear. However, we will not discuss this issue in this paper.

5.2 The inner symmetries

We believe that the previous extrapolations of the NP4 are not essential for the distinction of the parrot-like perception from human-like perception. We think that the essential difference between parrot-like and human-like perception lies in some structures (and related kinds of processing) of the nervous system. This structure manifests itself in the outer world by the so called inner symmetries of the phonemic system of the human language.

The phonemic symmetries are not something unusual. We use these symmetries in every-day life but we do not realize these facts. In this subsection we will try to describe the symmetries of the phonemic system in conceptual fashion without rigor and precise mathematical machinery. This conceptual approach is suitable for its ability to perform an analysis of the events and principles which are in their nature fuzzy and redundant. Some technicalities we have introduced in Appendix B.

The phonemic symmetries manifest themselves in the properties of similarity, analogy, in the grouping of some phonemes to the various hierarchical structures and in the linking of some phonemes to others according to certain rules. The fundamental manifestation of this symmetry is the so called classification schema of the given language phonemic system. We have presented in fig. 9 the classification schema of the Slovak phonemic system.

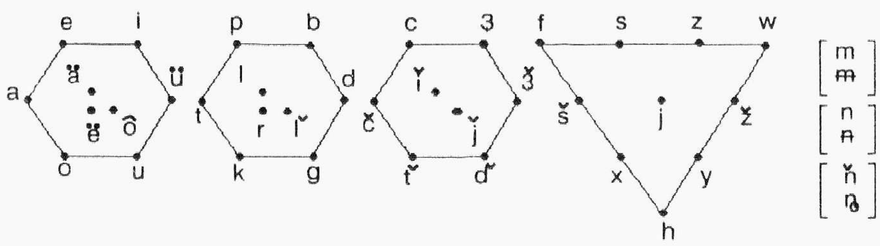


Figure 9

This classification schema can be considered to be a heuristic principle, too. We can see that all phonemes are classified to the 5 multiplets according to certain features we will mention later. The first multiplet is the nonet (singlet + octet) of the vowel-like phonemes. The second one is also the nonet (singlet + octet) of the semi-vowel-like phonemes and the stop-like phonemes. The third one is the octet of the stop-like and the affricative-like phonemes. The fourth multiplet is the decouplet of the fricative-like phonemes. The last one is the set of the nasal-like consonant phonemes.

At this point we must briefly mention what the phoneme "actually" is. In the theory of phonology the phoneme is interpreted as the smallest unit of phonology. This concept arose out of the awareness that the precise phonetic realization of a particular sound of speech is not important as its function within the sound system of a particular language. The phonetic variants of the particular phoneme are called the alophones. The phoneme is understood as a generalized form of a particular sound by disregarding all nonessential (dialectal, ideolectal, redundant) variants of the particular sounds in a process of abstraction. This process can be expressed by the phonemic difference between a pair of words which differ in only one phoneme, for example, pes (a dog) — les (woods). Each language has its own arrangement of phonemes, its own phonemic structure.In practice the linguists use the so called distinctive features to classify phonemes to the particular classes. There exist many systems of distinctive features. For our purposes the appropriate one is the system of N. Chomsky and M. Halle, [4], in which these features have been placed into categories, classes such as

1. Major class features — this class is related to such features as vocalic-nonvocalic, sonorant-nonsonorant, nasal-oral, etc.
2. Cavity class features — this class is related to such features as the point of articulation and shape of the oral cavity.
3. Manner of articulation features — relating to continuant-noncontinuant features
4. Source features — relating to such features as voiced-voiceless, etc.
5. Prosodic features — relating to such features as stress, pitch, etc.

The last class of features we do not consider apriori as inner symmetry, on the contrary they are just speech transformations under which the recognition mechanism is invariant (see Sect. 2).

The classification to the multiplets is based on the major and manner of articulation features. The sub classification in the given multiplet is based on the cavity features — horizontal lines and vertical lines on the source and cavity classes.First,consider non-nasal-like phonemes. In this case we have 4 horizontal lines to which we attribute the following cavity features:

upper line — (bilabial + labiodental + prealveolar) — front middle line — (alveolar + postalveolar + prepalatal) — central

lower line — (palatal + postpalatal + prevelar) — back lowest line — (velar + postvelar + glottal) — glottal

Similarly to the standard classification whose attributes are introduced above in brackets, our classification is a matter of convention. But we think that our classification — front, central, back, glottal — has some very interesting properties. The vertical lines in the first nonet describe the cavity features known in phonology as high, middle and low according to position of tongue. The vertical lines in the remaining multiplets describe the voiced-voiceless relation, for example (*p*—*b*, *s*—*z*). The phonemes in the middle part of a given multiplet have no pairs of phonemes according to the voiced-voiceless feature.

The last set of nasal-like phonemes contains three phonemes and for each phoneme belones an alophone or phoneme which has no semantic support. Again, the classification is due to the cavity features — front, central, back.

A comparison of the classification of phonemes introduced in fig. 8 with the classification of elementary particles — hadrons according to the SU(3) group and leptons, [9], tells us that there may exists some analogy between these two subjects. A mathematical treatment of this analogy from the point of view of perception and neural nets is discussed in Appendix B. This leads to some group theoretical description of our system of phonemes according to the given continuous Lie group SU(N) and its irreducible representations.

We do not think that the all ideas and speculations presented here are perfect in many details but we believe that in some coherent,holistic view these ideas can be useful and, maybe, true.

Finally, one can ask in the words of N. Bohr and W. Pauli: "Are these ideas sufficiently crazy to be true?"

APPENDIX

A. The homotopic group as the group along the trajectories

In this part of the Appendix we introduce an elementary discussion about several notions from topology related to the degree of the mapping between the two varieties. This concept underlies the understanding of the invariant feature extraction mechanism of the NP4.

Let us first define the mapping $f: X \to Y$. The varieties are compact n — dimensional, oriented and the boundaries of X, Y are null. The variety Y is connected. Then an integer number exists and this number is called the degree of the mapping f. In some elementary fashion this number describes how many times the Y covers the X by the mapping f. In our case we have the two real-valued functions $x(t)$, $y(t)$ from R^1. We define the trajectory in the complex plane as

$$z(t) = d(t)\,[x(t) + iy(t)]\,, \qquad (A.1)$$

where parameter d, in general dependent on time, is called the parameter of deformation of the trajectory. For this moment we consider the parameter d to be independent of time. In this case we note the global deformations to be contrary to the local deformations where d is dependent on time. If the trajectory does not intersect the beginning of the coordinate system we can define the phase of the trajectory $\Phi(t)$. It holds

$$\varphi(t_a) = \varphi_a\,, \quad \varphi(t_b) = \varphi_b\,, \quad t_a \leq t \leq t_b\,.$$

The trajectory can be intersected by itself, what means that

$$z(t_2, d) = z(t_3, d) \quad \text{for } t_2 \# t_3\,.$$

The trajectory is called the closed trajectory when it holds that

$$z(t_a, d) = z(t_b, d)$$

Because z is not equal to zero for $t_a \leq t \leq t_b$ then for every time t we can define the phase $\varphi(t)$ as we had already mentioned, but only with the nonuniquiness of $2k\pi$. To avoid this we can fix the initial phase $\varphi(t_a)$. The phase $\varphi(t)$ is a continuous function of time. We can define that the difference of the phases

$$\Delta\varphi = \varphi(t_b) - \varphi(t_a) = 2k\pi\,. \qquad (A.2)$$

We call the integer number k the degree of the trajectory $z(t, d)$. The physical meaning of the degree of the trajectory is clear from the following considerations and from figure A1. Let us first consider the elementary case of the closed trajectory

$$z(t, d) = c + d(\cos t + i\sin t)\,, \text{ where } 0 \leq t \leq 2\pi. \quad (A.3)$$

We see that the trajectory is a circle with its center at the point c and a radius of d.

If $d < c$ then the beginning of the coordinate system is out of this circle and the degree is null.

If $d = c$ then the trajectory intersects this beginning and the degree is not defined.

If $d > c$ *then the degree is equal to one. If the parameter d* is changing then we say that the trajectory (A.3) or (A.1) is deformed. Note that due to the integer nature of the degree, this remains unchanged — invariant under the continuous deformation of trajectory. Of course the deformed trajectory is not allowed to intersect the beginning of the coordinate system or the point where the phase of trajectory cannot be defined.

Following the elementary geometrical ideas we can express the difference of the phases $\Delta\varphi$ defined in (A.2) in the integral form as

$$\frac{\Delta\varphi}{2\pi} = \int_C d(\mathrm{arctg}[y'(t)/x'(t)]) = \int_L A(t)\mathrm{d}l\,, \quad (A.4)$$

where

$$A(\tau) = [y'(t)x''(t) - y''(t)x'(t)]/x'^2(t) + y'^2(t)]^{3/2}$$

$$dl = [x'^2(t) + y'^2(t)]^{1/2} dt$$

and

$$x' = dx/dt\,, \quad y' = dy/dt\,.$$

From these considerations we can conclude that the whole set of the trajectories in a question can be classified to the various classes due to their degree of the trajectory. These classes have different topological properties. In our model of perception we dealt with the quasi topological properties limited to a large degree values because we did not consider exactly the conditions of the closing of the trajectory.

All these concepts we have mentioned can be formulated exactly by the notion as the set of classes of equivalence of the trajectories, the homotopic classes or homotopic invariance [10]. The set of classes of equivalence has the group properties in the case of closed trajectories. This group is called the fundamental or Poincaré group of variety Y in its point y — $\pi(Y, y)$. The formulation of the fundamental theorem which describes our case can be the following:

If $f: X \to Y$ is the homotopic equivalence (or homotopic mapping) then for any point x from X homomorphism f^*: $p(X, x) \to p(Y, f(x))$ is an isomorphism.

(The exact definitions of the concepts introduced above are given in [10], again.) For this reason we have called the degree of the trajectory the homotopic invariant of the mapping in question. We must mention that the degree of the mapping is not the only homotopic invariant of the given mapping. The degree of the mapping is the only homotopic invariant in the case of the mapping of the n-dimensional compact, connected variety in S^n, where S^n is the n-dimensional unit sphere. All this can be formulated also for any n-dimensional topological space. In this case we call it a homotopic group instead of a Poincaré group.

From the point of view of our neural net, the NP4, we can say that the neural net realizes the homotopic mapping $f(x)$. This can be seen from the explicit form of the network and the simulation results which are given in Sect. 5. The mapping f depends on x, or on the state of the neural net and also on the time t. It is a local in time invariant mapping.

B. The gauge group as the group along the trajectories

In the following last part we will present some preliminary ideas and proposals of the direct incorporation of inner symmetries into the human speech perception and recognition model. As it was mentioned inthe above Discussion these symmetries are reflected in the symmetries of the phonemic system and we believe that they reflect one of the most important underlying commonalities between human and parrot perception and recognition.

Our approach was strongly inspired by the formalism of the Quantum gauge calibration field theory, where the symmetry principles are inherent and manifested to a great extent.

Let us consider the 4-dimensional space-time manifold X, this space is generated by the 4-vectors $x = (ct, x)$. Further we will put the constant c, the velocity of light, to be equal to 1. We postulate the mapping

$$f: X \rightarrow Y, \quad Y = f(x),$$

where X is the 4-dimensional vector space and Y is the 4-dimensional vector space of the acoustic field. In other words we postulate that for every point from X esists just one point y from Y. The physical relevant quantity in our case is the 4-vector of the acoustical strength of the acoustical field which acts on the basilar membrane or microphone membrane, [11].

Thus we can postulate the existence of the system of the fields of the patterns in the neuron system, in the brain, for which every point from the space X or Y is represented by the vector field $\psi(x)$. We postulate that this system of fields is gauge invariant and components of $\Psi(x)$ are grouped into multiplets which are transformed under the given irreducible representation of the Lie group G. We expressed the inner spa-

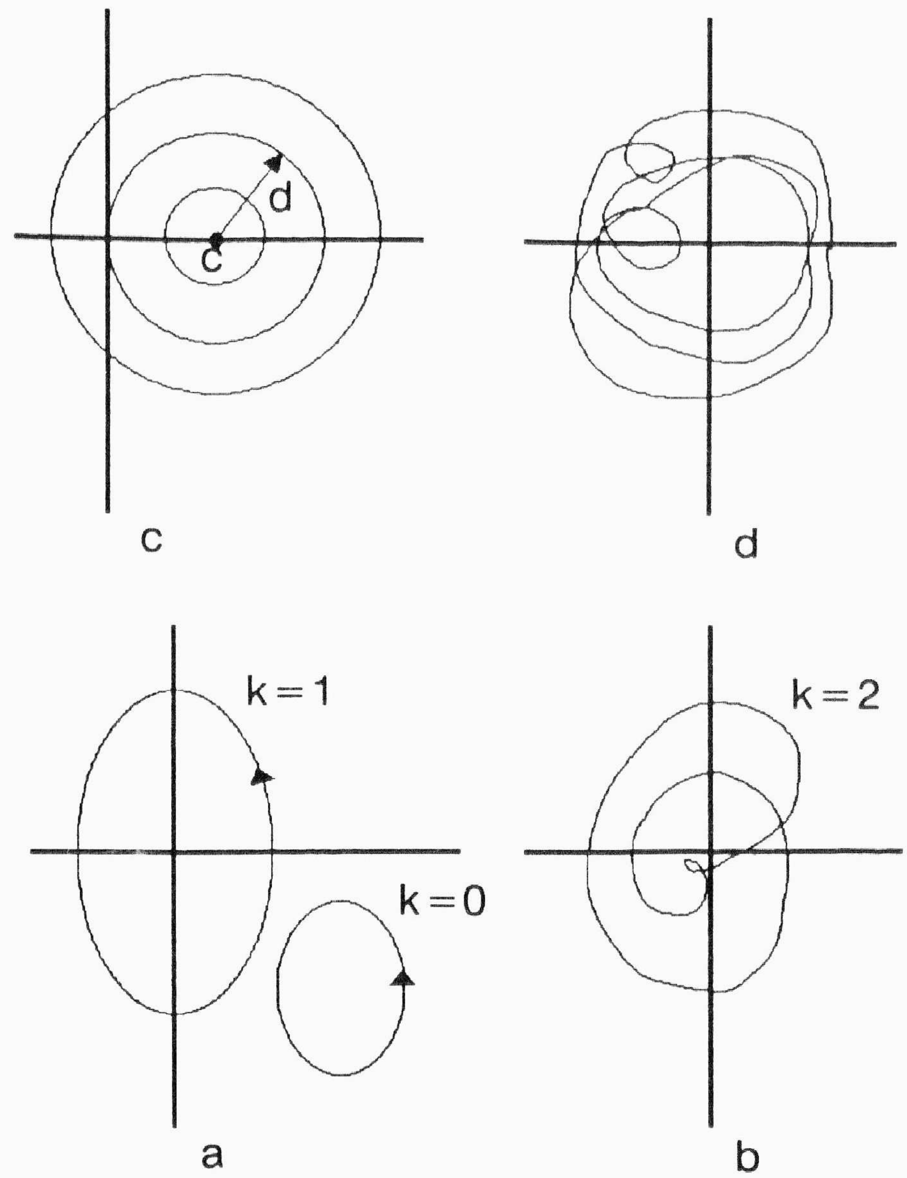


Figure A1

ce symmetries just in this way and we assume that just these symmetries are present in our phonemic system, too.

Now we briefly specify some concepts of the Lie groups. First we introduce the concept of the Lie algebra. Thus, let F be a commutative field. A Lie algebra over F is a vector space L over F equipped with a bilinear multiplication

$$[\,,\,]: L\,x\,L \rightarrow L$$

satisfying

$$[x, x] = 0 \quad \text{for} \quad x \in L$$

and the Jacobian identity

$$[[x, y], z] + [[y, z], x] + [[z, x], y] = 0 \quad \text{for} \quad x, y, z, \in L.$$

In field theory we are limited to N dimensional algebras the basis of which are generated by N so called generators L_a from L, $(a = 1, 2, \ldots, N)$ for which hold the so called commutative relations as bilinear multiplications

$$[L_a, L_b] = L_a L_b - L_b L_a = iC^c_{ab} L_c\,. \qquad \text{(A.5)}$$

We presume in this formula, and in the following ones, the Einstein summation rule. Real numbers C^c_{ab} are called the structure constants and they determine the whole structure of the Lie algebra L. *Now the ele-*

ment of the continuous group G for the given algebra described by (A.5) is defined as

$$U = \exp(-i\omega_a L_a) \qquad (A.6)$$

where $\omega_a(x)$ are some real numbers. The nontrivial expression (A.6) and its group properties follow from the Campbell-Hausdorf theorem. If any of the structure constants is not equal to null then the group G is not abelian. For the case of $N=3$ we have the SU(2) group which is the lowest dimensional non-abelian group. When we describe the generators in the matrix form we talk about a matrix representation of G. In field theory we are limited to the unitary and final-dimensional matrices.

Now we will briefly discuss the gauge transformations. We postulate the existence of the "lagrangian density" L_o of the neuronal-pattern system.The physical-information meaning of this lagrangian we will discuss later. Thus we have

$$L_o = L_o(\psi, \partial^\alpha \psi), \qquad (A.7)$$

where $\partial^\alpha \psi = \partial\psi/\partial x_\alpha$ and the gauge invariance means that

$$L_o(U\psi, \partial^\alpha U\psi) = L_o(\psi, \partial^\alpha \psi), \qquad (A.8)$$

where

$$U(x) = \exp(-i\omega(x))$$

is the element of the Lie group and $\omega(x)$ is the element of the Lie algebra under consideration. In this case we speak about the local gauge invariance. If U does not depend on x we talk about the global gauge invariance. Then

$$\psi(x) \rightarrow U(x)\psi(x),$$

$$\partial^\alpha \psi(x) \rightarrow U(x)(\partial^\alpha \psi(x)) + (\partial^\alpha U(x)\, \psi(x)$$

and for infinitesimal transformations we have

$$\partial\psi(x) = -i\omega(x)\, \psi(x),$$

$$\partial\left[\partial^\alpha \psi(x)\right] = -i\,\omega(x)\, \partial^\alpha \psi(x) - i\left[\partial^\alpha \omega(x)\right] \psi(x).$$

We define the covariant derivative as

$$D^\alpha \psi(x) = \left[\partial^\alpha + igA^\alpha(x)\right] \psi(x), \qquad (A.9)$$

where the $A^\alpha(x)$, the elements of the Lie algebra, are defined as

$$A^\alpha(x) = A^\alpha_a(x)\, L_a. \qquad (A.10)$$

If we require the covariant derivative to be transformed as $\psi(x)$ under the local gauge transformations then the fields $A^\alpha(x)$ must be transformed as

$$\partial A^\alpha(x) = \frac{1}{g}\partial^\alpha \omega(x) - i\left[\omega(x), A^\alpha(x)\right]$$

or

$$\partial A^\alpha(x) = \frac{1}{g}\partial^\alpha \omega(x) + C_{abc}\, \omega_b(x)\, A^\alpha_c(x). \qquad (A.11)$$

In this way we have defined N gauge fields, the so called Yang-Mills fields. Then it can be very easily shown that the substitution for the derivative in (A.8) by the covariant derivative makes the Lagrangian invariant relative to the local gauge transformations

$$L_o\left(U_\psi, D^\alpha U_\psi\right) = L_o(\psi, D_\alpha \psi).$$

The fields $A^\alpha(x)$ are interpreted as dynamical variables in the gauge field theory. For this we introduce the density of the free langrangian of the gauge fields as

$$-\frac{1}{4} F^{\alpha\beta}_a F_{a\alpha\beta} \qquad (A.12)$$

where

$$F^{\alpha\beta}_a = \partial^\alpha A^\beta_a(x) - \partial^\beta A^\alpha_a(x) - g\, C_{abc}\, A^\alpha_b(x)\, A^\beta_c(x)$$

The proportionality constant g is interpreted as the coupling constant, together with C_{abc} it expresses a relative weight of the particular fields. Thus we finally have the local gauge invariant and the Lorentz invariant density of the agrangian

$$L = -\frac{1}{4} F^{\alpha\beta}_a F_{a\alpha\beta} + L_o\left(\psi, D^\alpha_\psi\right). \qquad (A.13)$$

Now we will discuss the variations of the fields Y along a some trajectories in the outer coordinate space X or Y. If we have the gauge fields (A.10), then at every point of X the independent selection of the orientation of the coordinate system in the inner space of the fields of the patterns exists. The gauge transformations of A^α are correlated with the change of the orientation of the local inner system under the transition from point x to the point $x+dx$. The field $\psi(x)$ locally does not differ from $U(x)\,\psi(x)$ and the gauge transformations of the $A^\alpha(x)$ also do not change the properties of the system in qeustion. For the Yang-Mills fields we require the so called parallel translation to be null

$$dx_\alpha D^\alpha \psi(x) = dx_\alpha\left[\partial^\alpha + igA^\alpha(x)\right] \psi(x) = 0 \qquad (A.14)$$

Let us presume that $\psi(x)$ is translated in parallel under the trajectory P which is parameterized by the real parameter τ from $\langle 0, 1\rangle$. Then

$$x = x(\tau), \qquad A^\alpha = A^\alpha(x(\tau)),$$
$$\psi = \psi(x(\tau))$$

and

$$\frac{dx_\alpha}{d_\tau}\,[\partial^\alpha + ig\,A^\alpha(\tau)]\;\psi(x) = 0 \qquad \text{(A.15)}$$

with the solution written as

$$\psi(\tau) = T\left[\exp\left(-ig\int_{o}^{\tau} d\tau'\,(dx_\alpha/\delta\tau')\,A^\alpha(\tau)\right)\right]\psi(0), \qquad \text{(A.15a)}$$

where T is "time-ordering" operator defined as

$$T[a(\tau)\,b(\tau')] = a(\tau)\,b(\tau')\,\Theta(\tau-\tau') \pm$$
$$\pm\, b(\tau')\,a(\tau)\,\Theta(\tau'-\tau).$$

The upper sign holds for boson-like fields and the lower one for fermion-like fields. Then we can define for every trajectory P the matrix operator

$$\Omega(P) = T\exp\left[-ig\int_P dx_\alpha\,A^\alpha(x)\right]. \qquad \text{(A.15b)}$$

The physical meaning of this operator is to perform the gauge transformation of the $\psi(x(0))$ to the $\psi(x(1))$ along the trajectory

$$\psi(x(1)) = \Omega(P)\;\psi(x(0)), \qquad \text{(A.16)}$$

where $x(0)$ is the initial and $x(1)$ the final point of the trajectory P.

We will now prove that for the closed trajectory C the Spur $[\Omega(C)]$ is the gauge invariant quantity. We have for the gauge transformation of an arbitrary ψ

$$\psi'(x) = U(x)\;\psi(x)$$

and from (A.16)

$$\psi'(x(1)) = \Omega'(P)\;\psi'(x(0)),$$

what leads to the

$$U(x(1))\;\psi(x(1) = \Omega'(P)\;U(x(0))\;\psi(x(0),$$

$$U(x(1))\;\Omega(P)\;\psi(x(0)) = \Omega'(P)\;U(x(0))\;\psi(x(0))$$

and finally we have

$$\Omega'(P) = U(x(1))\;\Omega(p)\;U^{-1}(x(0)).$$

It follows from this directly that

$$\text{Spur}\,[\Omega(C)] = \textit{gauge invariant}. \qquad \text{(A.17)}$$

The most important property of $\Psi(C)$ is the following one. If the $\Omega(C)$ is known for an arbitrary closed trajectory C then in $\Omega(C)$ we have concentrated all the physical information about the gauge fields and, what is very important, all of the redundant information is out of this quantity, [12].

Now we can return to the physical-information interpretation of the "Lagrangian density" L given by (A.13). Following the previous considerations and [13] we postulate that the pattern field ψ is "perceived" or evaluated due to the least action principle

$$\textit{variation}\,(L) = 0.$$

This principle is necessary to compress the information rate so as to maximize the matching score of the given pattern by minimizing the pattern field excitations. The pattern prepared in this way will be topologically the most similar to the memorized pattern-prototype. From (A.17) we can conclude that the pattern fields (or neurons) are excited only in the case when the two patterns of the acoustical field cannot be transformable through the given gauge group transformations. We believe that the most straightforward correspondence between the neural nets and the gauge fields can be seen in the lattice-like formalism of the Feynman path integrals, [14].

From the comparison of (A.4) and (A.15b) we can conclude that the "parrot" case (A.4) is analogous to the "human" case, (A.15b). We can interpret this correspondence as the translation from "mechanical-like" processing to "field-like" processing.

References

[1] G. von Bekesy: Experiments in Hearing, McGraw-Hill, 1960
[2] W. S. Rhode, J. Acoust. Soc. Amer., Vol. 49, p. 1218, 1971
[3] H. Atmanspacher, MPE Preprint, Vol. 123, 1988
[4] N. Chomsky, M. Halle: The Sound Patterns of English, Harper & Row, 1968
[5] T. Kohonen: Self-Organization and Associative Memory, Springer-Verlag, 1988
[6] D. F. Specht, Neural Network, Vol. 3, p. 109, 1990
[7] P. A. Kolers, M. Eden: Recognizing Patterns, M.I.T. Press, 1968
[8] H. M. Nussenzveig: Causality and Dispersion Relations, Academic Press, 1972
[9] L. H. Ryder: Quantum Field Theory, Cambridge University Press, 1985
[10] W. S. Massey: Algebraic Topology, Yale University Press, 1971
[11] C. Moller: The Theory of Relativity, Clarendon Press, 1972
[12] E. S. Abers, B. W. Lee, Phys. Reports, Vol. 9, p. 1, 1973
[13] A. Masahiro, Phys. Rev. A, Vol. 37, p. 4415, 1988
[14] T. T. Wu, c. N. Yang, Phys. Rev., Vol. D, 12, p. 3845, 1975
[15] G. W. Bluman, S. Kumei: Symmetries and Differential Equations, Springer-Verlag, 1989
[16] V. Chudý, V. Hapák, L. Chudý, Selected papers of the symposium NEURONET '90, Elsevier Science Publishers, will be published, 1991
[17] G. M. Shepherd, Neurobiology, Oxford University Press, 1983

Neurocomputer Companies

Artificial neural networks extend in last few years from laboratories into commerce and industry. At present there are known several tenths of companies producing and seling the neuro-software and/or -hardware tools and services. In this section of our Journal we shall inform the readers substantially about the adresses of some of these companies.

Ward Systems Group, Inc.
228 West Patrick St.
Frederick MD 21701, USA
Tel.: (301) 662-7950

Association for Intelligent Systems Technology, Inc.
6310 Fly Road
E. Syracuse NY 13057, USA

TSA
1470 Wildrose Way
Mountain View CA 94043, USA
Tel.: 415/965-4561

Intelligence Ware
9800 S. Sepulveda Blvd.
Los Angeles CA 90045-5228, USA
Fax: (213)417-8897

Concordia University , Centre for Mach. Int.

1455 de Maisonave. , Blvd. West
Montreal, Quebec H3G 1M8, Canada

Hecht-Nielsen Neurocomputers
5501 Oberlin Drive
San Diego, CA 92121-1718
Tel.: (619) 546-8877
Fax.: (619) 452-6524

H. P. Central Mailing Dept.
P. O. Box 529
1180 Amstelveen, Netherlands

H. P. Comp.
3200 Hillview Ave.
Palo Alto CA 94304, USA

Scientific Computers Limited
50 Victoria Road
RH15 9LW Burgess Hill , West Sussex, England
Tel.: 011-44(444)65101 Telex:

Soft Center
NovaCast Expert Systems
S-37200 Ronneby, Sweden
Tel.: 011-46(457)718-30

Informagic, S. A.
Barcelona 08011, Spain SP
Tel.: 011-34(3)254-6235

Electronic Associates Sarl
25-27 Rue Ginoux
75 737 Paris Cedex 15, France
Tel.: 011-34(3)254-6235

The Numerical Algorithms Group Ltd.
NAG Central Office Mayfield House
256 Banbury Road
OX2 7DE OXFORD, England, United Kingdom UK
Tel.: +44 865 511245 Telex: 83345 NAG UK G

California Scientific Software
Grass Valley, California CA (800) 284 8112, USA

David Olmsted
411 Briargrove
Sildell LA 70458
USA

NeuralWare
Penn Center West — Bldg IV. , Suite 227,
Pittsburg PA 15276
USA

COMPAREX
Information Systeme GmbH
Gottlieb-Daimler Str. 10,
D-6800 Mannheim 1,
BRD

Floiting Point Systems Computing (FSP)
3601 S. W. , Murray Blvd.
Beaverton, Oregon OR-79005,
USA

Mimetics
6 Rue Lacepede
Paris 75005
France

Software A. G.
Veranstaltungen
Hardtring 100
Darmstadt W 6100
BRD

Software Tailors Company
1295 N Providence Rd. , Suite B103
Media, PA 19063,
USA

A NEURAL NETWORK SYSTEM FOR ACTIVE VISUAL PERCEPTION AND RECOGNITION

I. A. Rybak, A. V. Golovan, V. I. Gusakova, N. A. Shevtsova and L. N. Podladchikova)*

Abstract

A method for parallel-sequential processing of grey-level images and their representation which is invariant to position, rotation, and scale, is developed. The method is based on the idea that an image is memorized and recognized by way of consecutive fixations of moving eyes on the most informative image fragments. The method provides the invariant representation of the image in each fixation point and of spatial relations between the features extracted in neighboring fixations. The applications of the method to recognition of grey-level images are considered.

1. Introduction

It would be difficult to understand and to explain the remarkable features of living recognition systems on the basis of neurophysiological data only without findings of visual psychology and psychophysics. For the same reason, most of the classical neural network paradigms cannot be directly used for analysis and invariant recognition of grey-level visual images. So, on the one hand, it is necessary to develop adequate neural network models of preattentive vision including preprocessing visual information and extraction of primary features of visual images. Progress in this direction has already taken place due to the remarkable research of S. Grossberg and E. Mingolla [4, 5], J. Daugman [3], M. Porat and Y. Zeevi [7], J. Buchmann and Chr. von der Malsburg [1], and some others. On the other hand, it is necessary to develop methods and algorithms for transformation of extracted primary features of grey-level images into invariant features which can be used as input signals for classical neural networks. In this case the neural networks would successfully realize the functions of classifier and **associative** memory of visual images. An example of the successful application of a similar approach was given by S. Troxel, S. Rogers, and M. Kabrisky [8]. They used the transformation of the image into the magnitude of the Fourier transform with log radial and angle axis, $|F(Lnr, \Theta)|$, feature space, on the lower-level and the multilayer perceptron neural network using a back propagation algorithm on the upper-level of the recognizing system. But, it is interesting to find an adequate invariant transform and representation of the image on the basis of data and ideas of vision psychophysiology.

It is widely known that in the process of visual perception and image recognition human eyes move and consequently fixate on the most informative points of the image [9]. In accordance with the concept of Smart Sensing [2] (intelligent sensory perception), they actively accomplish a selective and problem-oriented collection of information from the visible world. The main principles of the Smart Sensing theory [2] are as follows:

(i) The eye is able to get exact information from a small area of the visual field only. The sharpness of perception decreases quickly from the fovea to the periphery of the retina. It provides the local processing in the areas of the fixation points and reduces the information processed in parallel.

(ii) The peripheral vision is of lower resolution but it excites and directs the gaze to shift to the next fixation point.

(iii) The high-level structures control eye movements for collecting information which is necessary for verification of hypotheses formed and reformed in the process of image recognition.

In 1971, D. Noton and L. Stark [6] carried out research devoted to comparing individual trajectories ("scanpaths") of human eye movements in two phases: when an object was being memorized (learning phase) and when it was being recognized (recognition phase). They have shown that these scanpaths are topologically similar and have suggested that an individual trajectory (a specific scanpath) is formed while the object is being viewed. As a result of this process, the object has been memorized and stored as an alternating sequence of sensory and motor memory traces, recording alternately the feature of the object and the eye movement required to reach the next feature. When the object is being recognized (when a hypothesis on the object is being verified), the reproduction of the successive eye movement memories and verification of the successive feature memories take place.

To realize the ideas described above in a concrete model, it is necessary to also develop the following aspects: 1. to realize primary transforms imitating

*) I. A. Rybak, A. V. Golovan, V. I. Gusakova, N. A. Shevtsova, and L. N. Podladchikova
Institute of Neurocybernetics at Rostov State University 194/1 Stachka Avenue, Rostov-on-Don 344104, USSR

a decrease in the resolution of visual field perception from the fovea to the retinal periphery; 2. to choose the set of primary features and to realize their extraction and encoding; 3. to develop algorithms for invariant representation of the image fragment in the fixation point and of spatial relations between the fragments in neighbor fixation points; 4. to develop algorithms of interactions between lower and higher levels of the system in operation modes of object memorizing, search, and recognition.

2. Primary Transform

The primary transform of an initial image $I = \{x_{ij}\}$ in the model developed forms the retinal image $I'(n) = \{x'_{ij}(n)\}$ in each n-th fixation point. The position of this point $(io(n), jo(n))$ and the resolution level $lo(n)$ in the round area rounding the point are considered as the initial parameters for the point. Three concentric circles with the center in the point $(io(\mathrm{n}), jo(n))$ divide the raster into four areas. The radii of the circles are

$$Ro(lo) = 3 \,.\, 2^{lo-1},$$
$$Ri(lo) = 3 \,.\, 2^{lo}, \qquad (2.1)$$
$$Rz(lo) = 3 \,.\, 2^{lo+1}.$$

Within the central round area, the image is represented on the resolution level $l = lo(n)$. Within the first ring area, it is represented with lower resolution (on the next resolution level $l = lo(n) + 1$). Within the second ring, it is represented on the resolution level $l = lo(n) + 2$.

Figure 1. Test image.

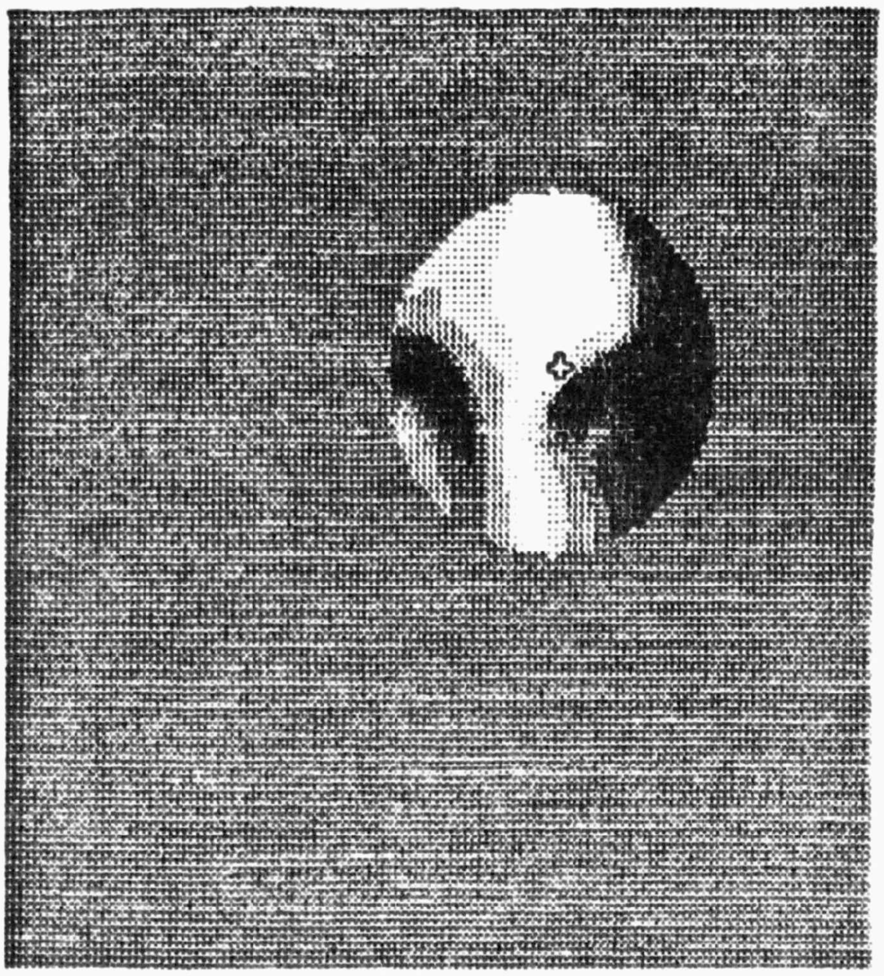


Figure 2. Retinal image in one fixation point.

To represent some part of the image $I = \{x_{ij}\}$ $((i, j) \in D)$ on the resolution level l the recurrent procedure of computation of the Gauss convolution in each point of the part D has been used:

$$\begin{aligned} x_{ij}^{(1)} &= x_{ij}, \\ x_{ij}^{(2)} &= \sum_{p,q} G_{pqij}\, x_{pq}^{(1)}, \qquad (2.2) \\ &\vdots \\ x_{ij}^{(l)} &= \sum_{p,q} G_{pqij}\, x_{pq}^{(l-1)}, \end{aligned}$$

where $G_{pqij} = -\exp\left\{-\upsilon^2\left[(p-i)^2 + (q-j)^2\right]\right\}$, (2.3)

υ is the coefficient $(\upsilon > 0)$ **and** i, j **and** p, q **are the coordinates of image pixels.**

Thus, the retinal image in the n-th fixation point $I'(n) = \{x'_{ij}(n)\}$ is formed from $I = \{x_{ij}\}$ in the following way:

$$x'_{ij}(n) = \begin{cases} x_{ij}^{(lo(n))} & \text{if } \; p_{ij}(n) \le Ro(lo), \\ x_{ij}^{(lo(n)+1)} & \text{if } \; Ro(lo) < p_{ij}(n) \le R_1(lo), \\ x_{ij}^{(lo(n)+2)} & \text{if } \; Ri(lo) < p_{ij}(n) \le R_2(lo), \\ x_{ij}^{(lo(n)+3)} = \bar{x}_{ij} & \text{if } \; p_{ij}(n) > R_2(lo), \end{cases}$$

(2.4)

where $\bar{x}_{ij}$ is the averaged intensity of $I = \{x_{ij}\}$ and

$$p_{ij}(n) = \sqrt{(i - io(n))^2 + (j - jo(n))^2}. \qquad (2.5)$$

Fig. 2 shows the retinal image in one fixation point (marked by the cross sign) for the initial test image (*Fig. 1*). *Fig. 3* shows that a few fixation points seem to be sufficient for image recognition.

3. Extraction of Primary Features

As primary features (image elements) we have considered the oriented edge segments extracted with different resolutions depending on their positions in the retinal image. Although it seems more prospective and adequate to extract primary image elements by the use of the Gabor transform [1, 3, 7], for the present we have used a slightly modified algorithm of S. Grossberg and his colleagues [5]. Orientation tuning of a neuron is determined by its receptive field which is formed as the difference of two Gauss convolutions with spatially shifted centers.

The magnitude of the input signal to a neuron (i, j) tuned to the edge segment orientation α is calculated in the following way:

$$Y_{ij\alpha} = \sum_{p,q} x'_{pq} \, (G'_{pqij\alpha} \times G''_{pqij\alpha}), \tag{3.1}$$

where

$$G'_{pqij\alpha} = -\exp\left\{-\gamma^2\left[(p-i-m_\alpha)^2 + (q-j-m_\alpha)^2\right]\right\}, \tag{3.2}$$

$$G''_{pqij\alpha} = -\exp\left\{-\gamma^2\left[(p-i+m_\alpha)^2 + (q-j+m_\alpha)^2\right]\right\}.$$

The step of orientation tuning of a neuron was 22.5° and it was taken to be the unit of angle measure

Figure 3. United result of primary transforms of the test image in 11 fixation points.

($\alpha = 0, 1, 2, \ldots, 15$). The parameters m_α and n_α depend on the neuron orientation tuning α:

$$m_\alpha = d(l)\cos(2\pi\alpha/16), \quad n_\alpha = d(l)\sin(2\pi\alpha/16), \tag{3.3}$$

where $d(l)$ defines the Gauss convolution center distances from the center of the receptive field (i, j) and depends on the resolution level l at the point (i, j) of the retinal image $I' = \{x'_{ij}(n)\}$:

$$d(l) = \max(2^{l-2}, 1) \tag{3.4}$$

Sixteen neurons corresponding to each point i, j, but tuned to different edge orientations (different directions of brightness gradients) interacted competitively owing to the strong reciprocal inhibiting connections. The interactions between the neurons are described as follows:

$$\begin{cases} \tau \dfrac{d}{dt} U_{ij\alpha} = -U_{ij\alpha} + Y_{ij\alpha} - B \displaystyle\sum_{k=0}^{15} Z_{ijk} - h, \\ Z_{ij\alpha} = f[U_{ij\alpha}], \ \alpha = 0, 1, 2, \ldots, 15 \end{cases} \tag{3.5}$$

where $U_{ij\alpha}$, $Z_{ij\alpha}$ and $Y_{ij\alpha}$ designate respectively the membrane potential, output and input signals of the neuron (i, j) tuned to the orientation α; B is the coefficient characterizing the reciprocal inhibitory interactions ($B > 1$); h is the threshold, and τ is the time constant. f[U] is the nonlinear function

$$f[U] = \begin{cases} U \text{ if } U \geq 0, \\ 0 \text{ if } U < 0. \end{cases} \tag{3.6}$$

The solution of the system passes to the state of equilibrium in which either all $Z_{ij\alpha} = 0$ (if all $Y_{ij\alpha} < h$) or only one $Z_{ij\alpha} = Y_{ij\alpha} - h > 0$ when $\alpha = \psi_s$ (for which $Y_{ij\alpha}$ is maximum), and the others $Z_{ij\alpha} = 0$ if $\alpha \neq \psi_s$. In the first case, it is considered that there is no oriented edge segments in the point (i, j). In the second case, it is considered that there is the edge segment in the point with the orientation $\alpha = \psi_s$ and with the corresponding contrast value as $Z_{ij\alpha}$.

In each fixation, the oriented edge segments are extracted in the fixation point $(io(n), jo(n))$ (the basic edge segment) and in 48 context points lying on intersections of 16 radiating lines differing 22.5° and of three concentric circles with exponentially increasing radii 2^{lo}, 2^{lo+1}, and 2^{lo+2} (*Fig. 4*). The oriented edge segments corresponding to the first (the smallest) circle are extracted with the same resolution as the basic one. The resolutions $lo(n)$ which the other edge segments were extracted with were determined by their position in the $I'(n) = \{x'_{ij}(n)\}$. The basic edge segment and context edge segments for one fixation point (the same as in *Fig. 2*) are shown in *Fig. 5* as the doubled white and black segments whose lengths are greater the lower the resolution.

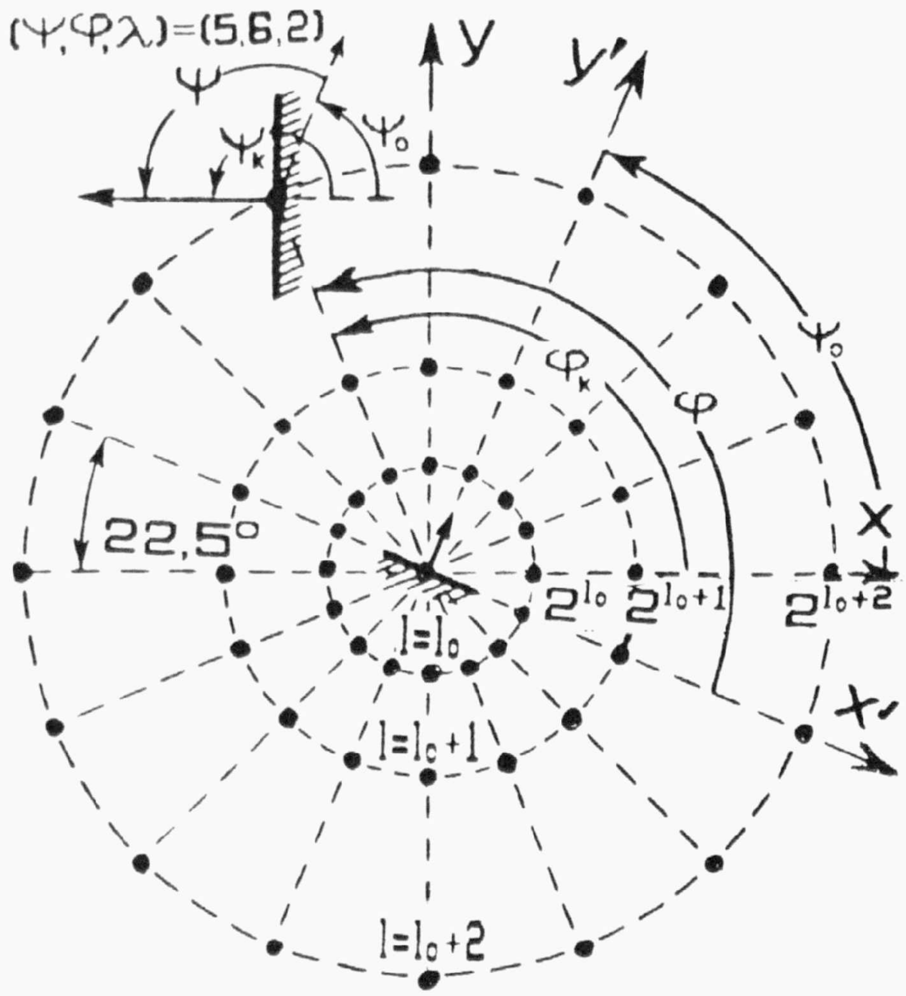


Figure 4. Positions of the basic edge segment and context edge segments in fixation point.

4. Invariant Representation of the Image

Each context segment can be invariantly encoded in a given fixation point by the relative orientation ψ, the relative angle position φ, and the relative decrease of the resolution level λ (see in *Fig. 4*). These parameters for each context segment are calculated in the following way

$$
\begin{aligned}
\psi &= \mathrm{mod}_{16}(\psi_k - \psi_o + 16), \\
\varphi &= \mathrm{mod}_{16}(\varphi_k - \psi_o + 4), \\
\lambda &= l - lo, \qquad\qquad (4.1) \\
\varphi, \psi &\in \{0, 1, 2, \ldots, 15\}, \\
\lambda &\in \{0, 1, 2\},
\end{aligned}
$$

where ψ_k and ψ_o are the orientations of the basic segment and the corresponding context one. φ_k is the angle position of the context segment in the coordinate system XOY, and l is the resolution level on which the context segment is extracted.

In this case, the image can be invariantly represented in each fixation point by the points on the surfaces of three tori each of which is formed by cyclic changing ψ and φ and corresponds to the definite value λ. This representation is shown in *Fig. 6* (for the same fixation point as in *Fig. 5*) as the black points in three coordinate systems on the evolvents of three tori (the abscissa and ordinate axes are the ψ and φ axes, respectively. Such representation of a grey-level imagage (or its fragment) in each fixation point is invariant with respect to position, rotation, and size. Besides, it is appropriate to the application of a classical neural network classifier for memorizing, storing and recognition of image fragments.

It is most natural and suitable that the next fixation point be chosen from the set of context points. In our model, in the memorizing mode, the choice of each next fixation point could be accomplished by a supervisor or automatically. In the latter case the choice is defined by the contrast values of the segments. The invariant encoding of image fragments in fixation points is a necessary but insufficient condition for invariant representation of the whole image. In addition, it is necessary to encode invariant spatial relations between neighboring fragments of a scanpath. In the model, it is provided by encoding the position of the basic edge segment in the next $(n + 1)$-th fixation point in the coordinate system (X'OY') joint with the basic edge segment in the previous n-th fixation point (by parameters $\Delta\psi_o(n + 1)$ and $\Delta\varphi_o(+ 1)$) and by encoding the relative change of resolution levels when the "gaze" is shifted from one fixation point $(\Delta lo(n + 1))$ to another (the position of the next fixation point is marked in *Fig. 6* by a circle). The parameters $\Delta\psi_o(n + 1)$, $\Delta\varphi_o(n + 1)$, and $\Delta lo(n + 1)$ are:

$$
\begin{aligned}
\Delta\psi(n + 1) &= \mathrm{mod}_{16}(\psi_o(n + 1) - \psi(n) + 16), \\
\Delta\varphi_o(n + 1) &= \varphi_k^*(n), \qquad\qquad (4.2) \\
\Delta lo(n + 1) &= lo(n + 1) - lo(n),
\end{aligned}
$$

where $\varphi_k^*(n)$ is the relative angle position of the context point (in the coordinate system joint with the n-th

Figure 5. The extracted basic and context segments in one fixation point.

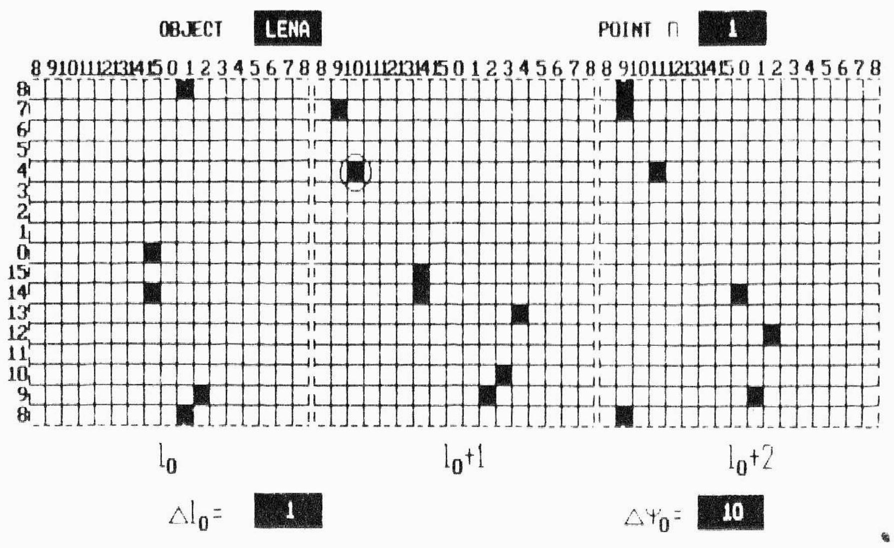


Figure 6. Invariant representation of one fragment (the same as in Fig. 5) as the black points in three coordinate systems on the evolvents of three tori.

fixation point) which is chosen as the next fixation point.

5. Object Memorizing, Search, and Recognition

The functional schema of the model of a visual neurocomputer is shown in *Fig. 7*. The model is to function in the following manner. In the mode of object memorizing, the image is processed in consecutively chosen points of fixation. In *Fig. 8a, b* the intermediate and final stages of sequential viewing of the object in the mode of memorizing are shown. In each fixation point, the set of oriented edge segments (the basic and several context ones) is extracted from the image fragment Then, the set is transformed into the form of three invariant patterns (as in *Fig. 6*). These patterns are memorized in the Neural Network playing the role of the Associative Memory for Fragment Storing. As a result of the memorizing mode, the fragments have been memorized in the Neural Network, and invariant relations between neighboring fragments have been memorized in the Motor Memory. In the mode of object search, the raster is scanned until a fragment similar to some memorized fragment of some object is found in some fixation point. When such a fragment is found, a hypothesis on the object is generated and the system turns to recognition. In the mode of recognition, consecutive fixations (controlled from the Motor Memory) and a consecutive verification of similarity of fragments (processed in the fixation points and represented in the invariant form) with the fragments stored in the Recognizing Neural Network take place. (A scanpath of viewing in the recognition mode ought to consequently reproduce the scanpath of viewing in the memorized mode). If a series of coincidences occurs, the decision is made that the object has been recognized. If it does not, the system turns to the mode of object search.

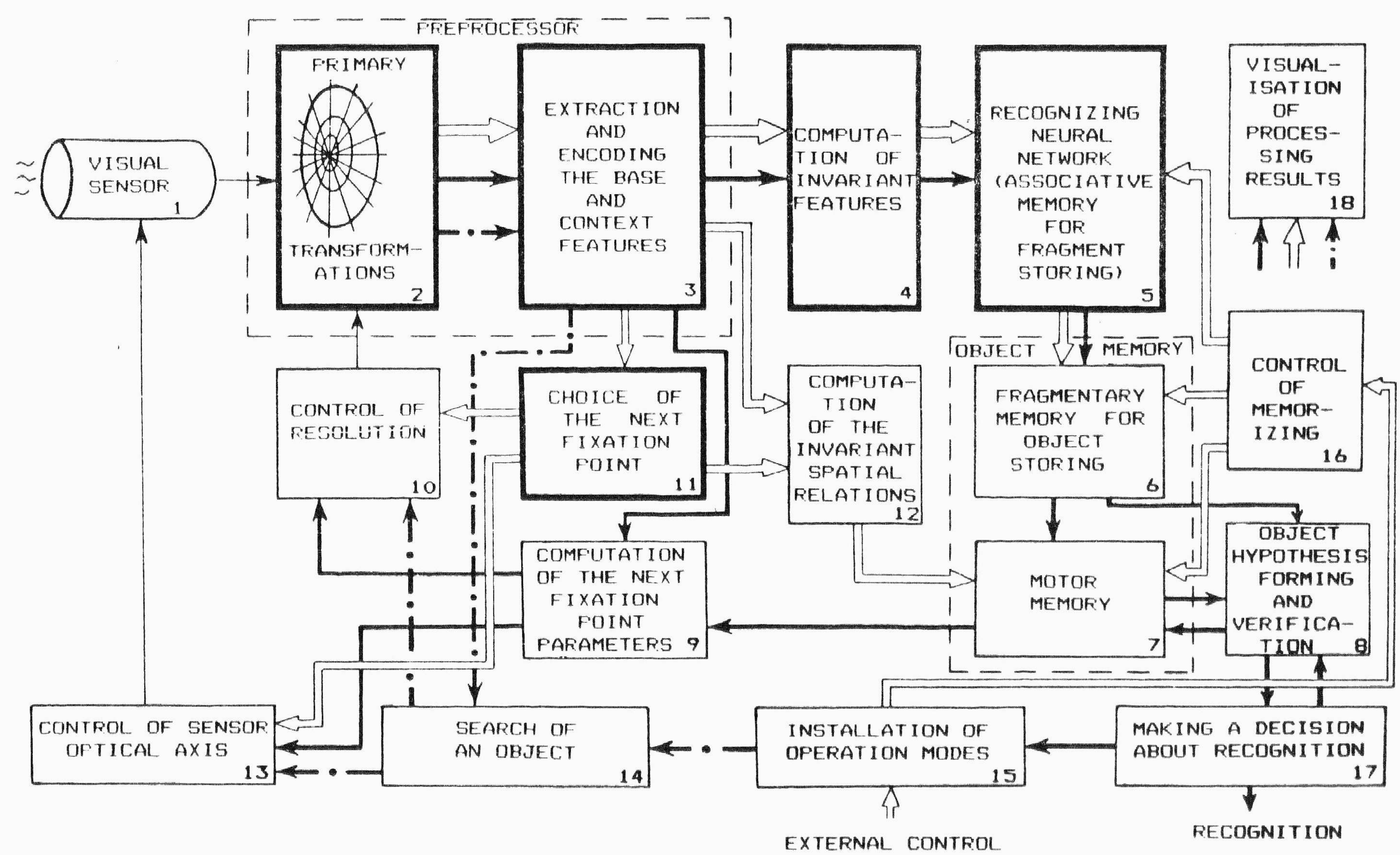


Figure 7. Functional scheme of the visual neurocomputer model. The interactions between functional blocks are depicted by arrows. The white, dot-dash, and black arrows relate to the modes of object momorizing, of object search, and of object recognition, respectively.

Figure 8. Intermediate (a) and final (b) stages of sequential viewing of the image.

6. References

[1] J. Buhmann, J. Lades, Chr. von der Malsburg, "Size and distortion invariant object recognition by hierarchical graph matching", *Proc. of the Int. Joint. Conf. on Neural Networks (IJCNN-90), San-Diego*, vol. II, pp. II—411—II—416, 1990.

[2] P. J. Burt, "Smart sensing within a pyramid vision machine", *Proc. of the IEEE*, vol. 76, pp. 1006—1015, 1988.

[3] J. Daugman, "Complete discrete 2-D Gabor transform by neural networks for image analysis and compression", *IEEE Trans. on Acoustic, Speech, and Signal Processing*, vol. 36, pp. 1169—1179, 1988.

[4] S. Grossberg, E. Mingolla, "A neural theory preattentive visual information processing: emergent segmentation, cooperative-competitive computation, and parallel memory storage", *Proc. of Int. Symp. "Struct. and Dyn. Nucl. Acids, Proteins, and Membranes, Riva del Garda, Aug. 31—Sept. 5, 1986"*, pp. 355—401, 1986.

[5] S. Grossberg, E. Mingolla, D. Todorovic, "A neural network architecture for preattentive vision", *IEEE Trans. on Biomedical Engineering*, vol. 36, pp. 65—84, 1989.

[6] D. Noton, L. Stark, "Scanpaths in eye movements during pattern recognition", *Science*, vol. 171, pp. 72—75, 1971.

[7] M. Porat, Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, pp.452—468, 1988.

[8] S. E. Traxel, S. K. Rogers, M. Kabrisky, "The use of neural networks in PSRI target recognition", *Proc. of the IEEE Int. Conf. on Neural Networks (ICNN-88), San Diego*, vol. I, pp. I—593—I—600, 1988.

[9] A. L. Yarbus, *Eye movements and vision*, Plenum Press, New York, 1967.

Literature Survey

Rasmussen S., Karampurwala H., Vaidyanath R., Jensen K. S., Hameroff S.: Computational Connectionism within Neurons: A Model of Cytoskeletal Automata Subserving Neural Networks, Physica Vol. D 42, 1990 pp. 428-449
Abstract: Numerous models of information processing within the cytoskeleton (in particular, microtubules) have been proposed. We have utilized cellular automata as a means to model and demonstrate the potential for information processing in cytoskeletal microtubules. In this paper, we extend previous work and simulate associative learning in a cytoskeletal network as well as assembly and disassembly of microtubules. We also discuss possible relevance and implications of cytoskeletal information processing to cognition.

Remus W., Hill T.: Neural Network Models of Managerial Judgment, Technical report: NN2.wp5/B9 Honolulu, Hawaii, University of Hawaii, 1989, 25 p.
Key words: decision making; neural networks; production scheduling.
Abstract: The study makes use of data from subjects making the production scheduling decision. The subject's behavior is modeled with linear decision rules and several neural network models. The data reveal that neural networks perform as well as linear decision rules but not better.

Saund E.: Distributed Symbolic Representation of Visual Shape, Neural Computation Vol. 2, 1990 No. 2 pp. 138-151
Abstract: This communication shows that the concept also offers motivation in devising representations for visual shape within a symbolic computing paradigm. In a representation for binary (silhouette) shapes, and in analogy with conventional distributed connectionist networks, descriptive power is gained when microfeatures are available naming important spatial relationships in images.

Seelen W. von, Mallot H. A.: Parallelism and Reduncancy in Neural Networks In:Neural Computers (Proc. of the NATO Advanced Research Workshop on Neural Computers), Neuss (Dusseldorf), FRG, Held: September 28-October 2, 1987 Berlin-Heidelberg, Springer-Verlag (Eds: Eckmiller R., Malsburg Ch.), 1989 pp. 51-60
Abstract: Biological systems take advantage of parallelism in many other respects including natural implementations, minimization of the number of computation steps, exploitation of signal redundancy, and a balanced distribution of processing tasks between all subsystems. As a result, reliability and accuracy of computation become exchangeable. We present examples for these principles of biological information processing and discuss how parallelism is used for their implementation.

Shadmehr R. ,D' Argenio D.: A Neural Network for Nonlinear Bayesian Estimation in Drug Therapy, Neural Computation Vol. 2, 1990 No. 2 pp. 216-225
Abstract: The estimation performance of a backpropagation trained network is compared to that of the maximum likelihood estimator as well as the maximum a posteriori probability estimator. In the example considered, the estimator prediction errors (model parameters and outputs) obtained from the trained neural network were similar to those obtained using the nonlinear Bayesian estimator.

Simic P. D.: Statistical Mechanics as the Underlying Theory of „Elastic" and „Neural" Optimizations Technical report: CALT-68-1556 & C3P-787 Pasadena, California Institute, 1989, 23 p.
Abstract: We derive a new algorithm of the elastic-net-type based on statistical mechanics.It has some of the „positive" ingredients of the elastic-net method, yet, it doesn't have an intrinsic problem (discussed in this paper) of the original algorithm.

Simpson P. K.: Higher-Ordered and Intraconnected Bidirectional Associative Memories
IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, 1990, No. 3, pp. 637—653
Abstract: Autocorrelation associative memories (autocorrelators) are feedback neural network architectures that store bipolar patterns. Heterocorrelation associative memories (heterocorrelators) are feedback neural network architectures that store bipolar pattern pairs. The relationship between autocorrelators and heterocorrelators is examined. A review of the encoding algorithms, recall operations, stability proofs, and capacity arguments of first and higher-order autocorrelators, commonly referred to as Hopfield associative memories (BAM's) is presented. Higher-ordered BAM's and first— and higher— ordered intraconnected BAM's are then introduced. The encoding, recall, and stability procedures of each are discussed. Finally, simul. results that compare the storage capacity and storage efficiency of three heterocorrelators, the heteroassociative BAM, the BAM, and the intraconnected BAM, are presented.

Surkan A. J., Wendel F. C., Lee S. M.: Neural Network Pattern Learning for Classifying Administrators from Examples, IEEE, 1990 pp. 335-362
Abstract: A simulated neural network demonstrates the ability of trainable connection-base problem solvers to build and identify an effective prediction model.

Sutton R. S.: First results with DYNA, an Integrated Architecture for Learning, Planning and Reacting In:Spring Symposium on Planning, Stanford, February 20, 1990, 1990 pp. 1-5
Abstract: In this paper I briefly introduce DYNA, a simple architecture integrating and permitting tradeoffs among these approaches. Results are presented for a simple Dyna system that learns from trial and error while it learns a world model and uses the model to plan reactions that result in optimal action sequences.

Tang D. S., Menon V.: Temporal Differentiation and Violation of Time-Reversal Invariance in Neurocomputation of Visual Information, Neural Computation Vol. 2, 1990 No. 2 pp. 162-172
Abstract: Information-theoretic techniques have been employed to study the time-dependent connection strength of a three-layer feedforward neural network. The analysis shows (1) there is a natural emergence of time-dependent receptive field that performs temporal differentiation and (2) the result is shown to be a consequence of a mechanism based on violation of the time-reversal invariance in the visual information processing system. Both analytic and numerical studies are presented.

Thacker N. A., Mayhew J. E. W.: Designing a Layered Network for Context Sensitive Pattern Classification, Neural Networks Vol. 3, 1990 No. 3 pp. 291-299

Key words: neural nets; pattern classification; information preservation; flexible architectures; learning.

Abstract: The new network has been designed to be robust under noise while still maintaining enough flexibility to learn new patterns. This is achieved by a combination of a novel new-node generation algorithm and a simple resonance mechanism. The network has a fixed number of layers which are used to classify accumulated classifications from previous layers. The number of nodes in the network is flexible and a complete classification network can be grown from just a few seed nodes during the course of training. Connectivity is also flexible, connections are generated and maintained according to the demands of the training data. This network is not meant to be a complete solution to the problem of context-sensitive classification but a step towards making such networks possible. Its use is demonstrated in the recognition of planar objects given edge vectors.

Traub R., Miles R., Wong R. K. S.: Brain Waves, Communications of ACM Vol. 35, 1990 No. 4 pp. 1-387

Wang J. H., Krile T. F., Walkup J. F.: Determination of Hopfield Associative Memory Characteristics Using A Single Parameter, Neural Networks Vol. 3, 1990 No. 3 pp. 319-331

Key words: Hopfield associative memory; signal-to-noise ratio; attraction radius; convergence probability; direct convergence; indirect convergence; storage capacity.

Abstract: A new statistical method is proposed for exploring the characteristics of the Hopfield associative memory (HAM). The existence of an average signal-to-noise ratio parameter (which we call C) has been successfully applied to derive equations that are capable of concisely estimating the storage capacity of (a) direct convergence nets in which the initial vector is required to precisely converge to the memorized vector in one iteration and (b)indirect convergence nets in which a specified error epsilon is allowed after multiple iterations.The close tie between the memory capacity and the required convergence probability of the HAM is described.The significance of the 1-to-1 relationship between the indirect convergence probability and the parameter epsilon/eta(etaprobabitity of a neuron state being an incorrect bit)is shown.The importance of the parameter ETA in determining the capacity of the direct convergence nets is also discussed.

Xu X., Tsai W. T.: Constructing Associative Memories Using Neural Networks, Neural Networks Vol. 3, 1990 No. 3 pp. 301-309

Key words: neural network model; associative memory; memory capacity.

Abstract: This paper proposes using a generalized Hopfield's model, also known as the McCulloch-Pitts model, as associative memories. The system always converges to the nearest stored words. This is not true for Hopfied's model. We also determine the information capacity of this generalized model. Our result is better than those in the literature.

Yamanaka K., Koren Y.: A Probabilistic Model of Neural Networks with Static Attractors., IEEE Transactions on Systems, Man and Cybernetics Vol. 20, 1990 No. 4 pp. 921-922

Abstract: A probabilistic version of the binary Hopfield networks is proposed.Operation of the network is completely in the sense that evolution of each unit is governed only by its inherent probabilistic law.It is shown that the global state is attracted by one of the equilibria with probability one. Yan H.,Gore J.C.: Weight Adjustment Rule of Neural Networks for Computing Discrete 2-D Gabor Transforms., IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol.38, 1990 No.9 pp. 1654-1656

Abstract: Daugman has recently proposed a neural network model for computing the discrete 2-D Gabor transform. We prove here that the weight adjustment rule used in the neural network is equivalent to the use of Jacobi iteration for solving simultaneous linear equations, and we propose more efficient algorithms for solving the problem.

Zhang D., Jullien G. A., Miller. C.: A Neural-Like Network Approach to Finite Ring Computations, IEEE Transactions on Circuits and Systems Vol. 37, 1990 No. 8 pp. 1048-1052

Key words: neural networks; finite rings; residue number systems.

Abstract: This paper discusses computation over finite rings using networks modelled after the general neural network approach. In this case the neurons are arithmetic elements that have modulo operator characteristics, rather than the usual nonlinear, saturating characteristics of learning and associative memory neural network applications.

A VIEW ON NEURAL NETWORKS PARADIGMS DEVELOPMENT

(Part 4)

J. Hořejš)*

Here we continue in the tutorial paper concerning the neural network paradigm, which first part was published in the Neural Network World, No. 1, 1991.

d) Dimension reduction.

There is still another use of training a net *m-k-n* to the identity mapping. In the net 6-2-1, taught by BP to recognize symmetry (see *Fig. 6*), 17 weights were adapted (2×6+2 drawn in the Figure + 3 thresholds which are in BP treated as weights from fictive neurons). Thus, the weight space was 17-dimensional and it would be of course interesting to have an idea, how the trajectory of 17-dimensional vectors **w**0 – >**w**1 – > **w**2 – > ... looks like during the process of adaptation. Using methods of descriptive geometry in 17-dimensional space is of course hardly thinkable. To get at least some visualization, the nets 17-2-17 and 17-3-17 were constructed by Pelikan* and the weights from the above 6-2-1 problem, as they have arisen during adaptation of that (symmetry) problem were used as the training set for our 17-*-17 nets, which were taught the identity mapping; i.e. *T* = {[**w**0, **w**0], [**w**1, **w**1], [**w**2, **w**2], ...} After this training successfully ended, we were able at least to draw 2- and 3-dimensional „neural" projections of the 17-dimensional trajectory. You can see them on *Fig. 24ab* and observe moments, where changes (both in magnitude and in direction) were interesting.

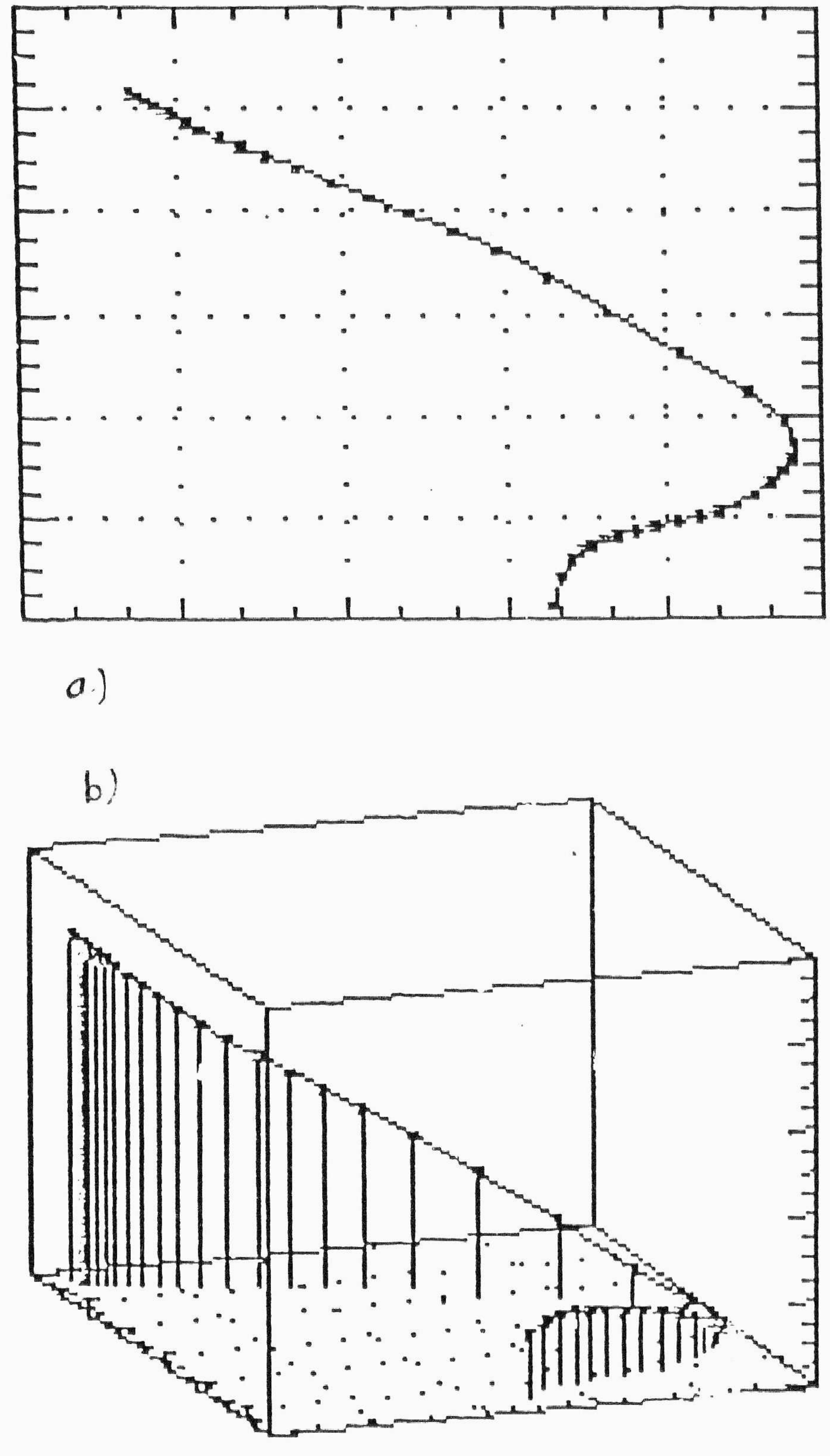


Fig. 24a, b

e) "Instant expert systems"

A simple rule oriented expert system, say a medical diagnostic system, can be expressed in a form of layered graph as in Fig.25.

At the bottom there is a layer of symptoms (*S*1, . . . ,*Sm*), which can be established by a doctor and/or patient (subjective feelings, laboratory findings etc). In the next layer there are probable diagnosis (*D*1, . . . ,*Dk*) and in the output layer, recommended therapy (*T*1, . . . ,*Tn*) is indicated. Medical experts then evaluate numerically „weights" (measures of certainty or credibility) by which the system can conclude that from established facts other ones follow. From *Fig. 25* we see e.g. that presence of symptoms *S*1, *S*2, *S*3 leads to a conclusion that diagnosis *D*1 is very probable (symptom *S*1 contributing to the hypothesis by positive influence 3, *S*2 by 1). Yet from the two therapies *T*1 and *T*2 which may cure the disease (with the same expectations if diagnosis *D*1 alone is taken into account, because of expert evaluation 2 in both cases), the symptom *S*3 (indicating e.g. allergy on some medication required by *T*1) prefers *T*2.

As seen, the graph does not form a CMN; first, experts seem not to have enough evidence on the direct influence of *S*3 on the diagnosis *D*1 per se (and other connections are missing too), second there are some conclusions which by-pass the diagnosis level leading directly to the therapy level (dashed lines). Moreover various experts often differ in their influences estimations and in complex cases expert systems are prone to other misunderstandings and problems as well.

NNs can principally offer another solution. It is possible to recalculate the BP algorithm so that it covers some by-passes and sparse weight vectors. Also,

*) Prof. Dr. Jiří Hořejš, Department of Computer Science, Charles University, 118 00 Prague 1, Malostranské nám. 25, Czechoslovakia

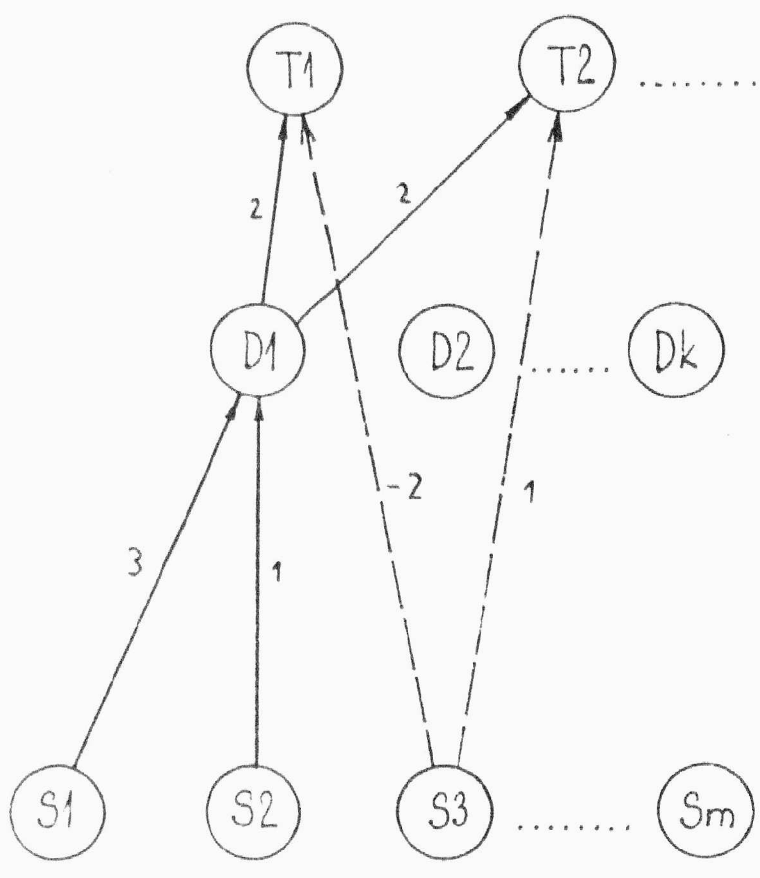


Fig. 25 expert system

it seems that many times it leads to a somewhat better solution introducing a true completely multilayered net with several hidden layers not reflecting on the fact that unlike the system from *Fig. 25* we will not be always able to say which knowledge a given neuron and/or weight in hidden layers represents in medical (or the task dependent) terminology.

The problem of precise *knowledge representation* in NNs is well understood only in some parts of some nets; see g) below. What is important is that we always do know what the inputs and outputs of single neurons and weights of connections attached to them in not hidden layers mean. Thus if we are concerned only with the relationship between symptoms and therapy and can perhaps miss a statement of diagnosis, we can use the above BP techniques and form a huge training set from a hospital database and let the BP to extract the necessary knowledge. Also possible is to extend the output vectors and include there the diagnosis components.

While these approaches are theoretically all right, in practice we can of course encounter many troubles. For simple expert systems we found that their efficiency as compared with some diagnostic expert systems is about the same. However the problem of *explanatory function* of NNs, explaining why and how the final verdict has been concluded (which is usual and relatively simple to implement in rule based expert systems) are more difficult if we have no good names for the hidden layer neurons. Yet some explanations may be sometimes offered, as seen from the following example.

Sima* developed a general expert-like system EXPSYS on the basis of multilayered net, admitting bypasses and including a technique for estimating „credibility" of implications between input nodes information and output ones, and dealing successfully with incomplete information, which permits the expert simply to admit that he/she does not know. The system is expert-oriented in that way they coding and decoding input/output information is automatically translated from the expert language to the language of NNs [every expert statement is mapped on appropriate number of neurons and back]. The training set is expressed in terms of the expert language and implementation techniques are for him/her hidden. Moreover, EXPSYS enables to provide a sort of explanatory facility, indicating those components of input expert description which give most significant reasons for the given answers. This is a consequence of the fact, that Sima uses the whole interval $[-1,1]$ (with the nonlinear transfer admitting these values), where, roughly speaking, YES is represented by 1, NO by -1, while DON'T KNOW by the whole interval $[-1,1]$. (In fact the user can create any number of choices by introducing enumeration type). A sort of interval arithemetic then propagates through the net, distinguishing excitation contributions from inhibitory ones as well as their ratio.

The power of the system has been demonstrated on a 16-25-20-8 net for diagnosis of protection warnings in atomic energy plants (for the cooling system of reactor VVER 410 V-213 to be specific). Starting from 80 original members of T, he subsequently added (after consultation with an expert) another 32; afterwards the net reached global error minimum near 0 (6 hours on a VAX) and good generalization ability as was proved on a test set Q of about 300 members, where more than 80% of answers were accepted (with a feeling of surprise) by an expert, disappointed by previous experience with classical expert systems.

There are two well-known and instructive „historical" examples of NNs applications, which should be not missed here.

f) NETTALK

As one of the first convincing application of BP, Sejnowski developed a system for transforming an English written text into a sequence of pronounced sounds — he taught the net to talk. The translation was context dependent, because the same letters are often pronounced in a different way depending on a large piece of text (and thus the meaning of the words). A rather large input text was chosen and a window moving over it always read 7 consecutive letters assigning to the letter in central (4^{th}) position an appropriate code for its spoken counterpart. For each letter in the input, the code 1 out-of 29 was used, giving thus the total number of neurons in the input layer 203. Each (and the only) phoneme in the output layer was coded by 1 out-of 26 neurons. The hidden layer consisted of 80 neurons. The net 283-80-26 which has just found (or was taught) that the proper transform for „*c*" in a given context is [german] „*k*", is shown in *Fig. 26*.

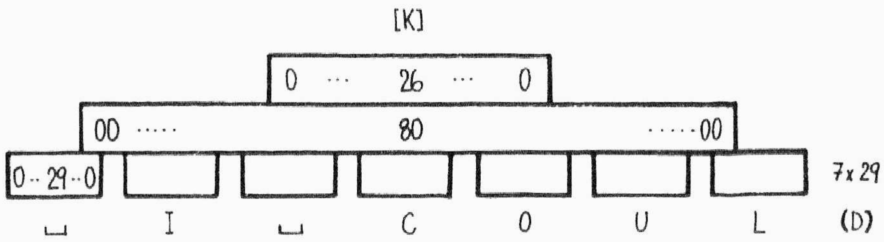


Fig. 26

It should be noted that many attempts are now followed to pursue the inverse task: to transform the spoken language into its written form (e.g. the Kohonen's project of phonetic typewriter). These and similar complex tasks use also NNs, but in connection with other techniques. This is a general characteristic of todays research & development: in many problems, NNs are important but not the only tools used.

g) Representation in genealogy trees.

Hinton presented an example of two families (an English and an Italian one) and relationship between their members ("is a son", „is an aunt", „is a wife" etc) [in form of two genealogy trees]. A multilayered net, which adapts its weights whenever a pair „person 1 — relationship" is submitted on the input, while the teacher completes the triple by a „person 2" (being in a stated relationship with person 1) on the output. See *Fig. 27.*

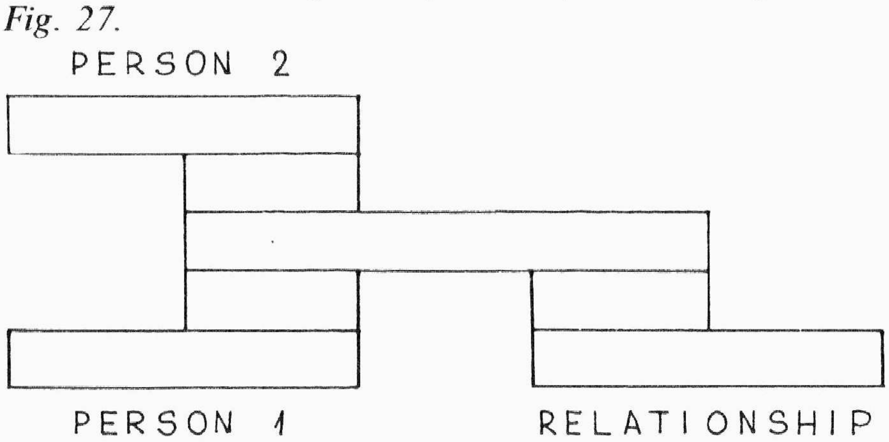


Fig. 27

The training set T consisted of 100 out of 104 possible triples and the remaining cases were correctly generalized (Q having 4 members of form „person 1 — relationship"). The net even found that a particular person has two aunts (2 out of 24 neurons of „person 2" layer became active). Because (names of) persons are coded again by 1 out-of 24 (number of persons in the game) and similarly for the relationship, the generalization abilities can hardly be of a numerical interpolation nature.

So the net really had to „understand" the inner rules which it was taught. In fact, Hinton has been able to discover how some knowledge of the net is internally represented. It was found that some of the neurons developed an ability to represent some general features of the relationship involved, although there were no explicit information in this direction delivered to it. It turned out that e.g. the first neuron in the immediately upper level encodes nationality (having converged to the positive values for Englishmen and negative ones for Italians), next three neurons encode generation level etc. Thus besides the known and formerly illustrated concepts of „knowledge extraction", we have here moreover a clear demonstration of internal „knowledge representation".

Because of the dominant role of the BP, we will include in the further section some additional (slightly more advanced) exposition; it is aimed at those readers who found previous few chapters exciting enough to experiment with the program of sect. 7A, encountered problems in applications on more sophisticated tasks and are even keen to pass from metaphors to real work, suggesting further techniques, explanations, improvements and programming. The section is therefore exceptional also by introducing some references immediately in the text.

(Continuation)

BOOKS ALERT

The following books can be interesting for the readers of our Journal

Information in the Brain. A Molecular Perspective. Ira B. BLACK. Cambridge, MIT Press 1991. 240 pp. 47 illus. $ 30.00.

From Animals to Animals. Proceedings of the First International Conference on Simulation of Adaptive Behavior. Edit. Jean-Arcady MEYER and Steward W. WILSON. Cambridge, The MIT Press 1991. 500 pp. $ 55.00. ISBN 1-800-356-0343.

Intelligent Databases. Object Oriented and Deductive Hypermedia Technologies. K. PARSAYE, M. CHIGNELL, S. KHOSHAFIAN and H. WONG. London, John Wiley & Sons 1989. 496 pp. $ 33.15.

Presented here is a model for intelligent databases based on five information technologies; databases, object-oriented programming, expert systems, hypertext and text management. This self-contained treatment actually shows how to construct an intelligent database on a microcomputer by building a shell much like the shells used to construct expert systems. A realist intelligent database example is used throughout the book to illustrate ideas being discussed.

ISDN Explained. Worldwide Network and Applications Technology. John M. GRIFFITHS. London, John Wiley & Sons 1990. 190 pp. $ 35.10. ISBN 0-471-92675-2.

The advent of ISDN has prompted much discussion in the telecommunications community and created the need for a single overview explaining the various facets of the ISDN operation. This book aims to provide engineers and communications managers with a technical outline of ISDN and an introduction to the detailed Standards, Recommendations and Specifications.

Local Area Networks. The Next Generation. *Second Edition.* Thomas W. MADRON. London, John Wiley & Sons 1990. 320 pp. $ 30.70. ISBN 0-471-52250-3.

Thoroughly revised and updated, this practical book will help users to succeed in setting up and maintaining those important micro-to-mainframe links, as well as to solve a host of other networking problems.

Methods in Neuronal Modeling. From Synapses to Networks. Edit. Chistof KOCH and Idan SEGEV. Cambridge, The MIT Press 1991. 538 pp. $ 22.95.

Neural and Automata Networks. Dynamical Behavior and Applications. E. GOLES and S. MARTINEZ. — Dordrecht, Kluwer Academic Publishers Group 1990. 264 pp. GBL 49.00 ISBN 0-7923-0632-5.

This volume provides a broad mathematical framework for the study of the dynamics of automata and neural networks. The main theoretical tools developed are Lyapunov functionals which enable the description of limit orbits and the determination of bounds for transient lengths. Applications to models in statistical physics are presented.

Neural Networks. EURASIP Workshop 1990. Sesimbra, Portugal, February 15-17, 1990 Proceedings. Edit. L. B. ALMEIDA, C. J. WELLEKENS. Berlin, Springer-Verlag 1990. IX, 276 pp. (Lecture Notes in Computer Science, Vol.142) DM42,— ISBN 3-540-522255-7.

The EURASIP workshop contribution collected in this volume have an interdisciplinary character. The authors include psychologists, biologists, engineers and mathematicians as well as computer scientists.

Pattern Recognition by Self-Organizing Neural Networks. Edit. Gail A. CARPENTER and Stephen GROSSBERG. Cambridge, The MIT Press 1991. 640 pp. $ 47.50.

Books Alert

The following books can be interesting for the readers of our Journal.

Artifical Intelligence, Simulation and Modeling. Lawrence E.Widman, Kenneth A.Loparo and Norman R. Nielsen. Chichester, John Wiley & Sons 1989, 576 pp. $39.80/$60.45, ISBN 0471 60599 9.

This state-of-the-art presentation reviews all the important concepts involved with using artifical models. The authors who have contributed articles for this volume are researchers and practitioners at the forefront of this rapidly developing technology.

Artifical Neural Systems. Foundations, Paradigms, Applications and Implementations. Patrik K. Simpson. New Jersey, Pergamon Press INC. 1990, 224 pp. $19.50/$39.50, ISBN 0-09-037894-3, ISBN 0-08-037895-1.

For the first time in one concise volume, a broad range of Artifical Neural Systems models are clearly described within a common framework. The author includes a comparison between real neural systems, artifical neural systems and the conventional computer, a presentation of the foundational concepts and constructs used to analyze and characterize ANS's, and a history of the field since the early 1940's. The remainder of the book analyzes 28 ANS paradigms, their applications and implementations. An extensive bibliography is also provided.

Cognizers Neural Networks and Machines That Think. R. Colin Johnson and Chappell Brown. Chichester, John Wiley and Sons 1988, 272 pp. $27.95, ISBN 0471 611611.

This book examines the fascinating history, hard science and research behind the computers that process information in much the same way humans do. It charts the rapid developments and exciting discoveries years.

Communications and Networks. A Handbook for the First Time User. P. Croucher. Chichester, John Wiley and Sons 1989, 192 pp, $24.75, ISBN 1-85058-136-3.

A comprehensive guide to transferring files between computers and networking hardware in order to minimise costs.

Complex Systems and Cognitive Processes. R. Serra and G. Zanarini. Berlin, Springer Verlag 199o, 205 pp. 71 figs. Hardcover DM 78,—. ISBN 3-540-51393-0.

After a general introduction to the theory of complex systems, the book gives a thorough treatment of neural networks, which are the most successful and the most thoroughly studied dynamical cognitive systems.

Computers for Artifical Intelligence Processing. Edit. Benjamin W.Wah and C.V.Ramamoorthy. Chichester, John Wiley & Sons 1990, 604 pp. $71.95, ISBN 0471 84811 5.

Artifical Intelligence algorithms and programs are rapidly outgrowing the computation powers of most present-day, conventional computers. The support of efficient symbolic processing requires the development of new architectural features, languages, algorithms and representation schemes.

Encyclopedia of Artifical Intelligence. Edit. Stuart C. Shapiro. Chichester, John Wiley & Sons 1990, 1244 pp. $95.00, ISBN 0471 52079 9.

„This is a truly fantastic Encyclopedia, more than one and a half million words and what can only be described as a tour de force . . . the definitions and explanations given are admirable . . . a book which no one in the field of Artifical Intelligence can afford to be without."
INTERNATIONAL JOURNAL OF SYSTEMS RESEARCH AND INFORMATION SCIENCE
„The Encyclopedia is a wonder of clarity and scope: surprisingly easy to read . . . The clarity is an especially pleasant surprise, considering the articles were all written by AI experts . . . It's a treasure house of easily accessible knowledge." LANGUAGE TECHNOLOGY

The Chaotic Dynamics of Nonlinear Systems. S. N. Rasband. Chichester, John Wiley & Sons 1990. 240 pp. $48.95, ISBN 0-471-634182.

This is the first non-mathematical textbook with exercises written on the subject of chaos. Ideas are presented in their simplest form and developed to sufficient depth so that actual computations can be made by the reader. The major paradigms in the transition to chaos exhibited by dynamic systems are covered in a coherent and integrated fashion.

Instructions to authors

1. Manuscript
Two copies of the manuscript should be submitted to the Editor-in-Chief

2. Copyright
Original papers (not published or not simultaneously submitted to another journal) will be reviewed. Copyright for published papers will be vested in the publisher.

3. Language
Manuscripts must be submitted in English

4. Text
Text (articles, notes, questions or replies) double space on one side of the sheet only, with a margin of at least 5 cm, (2") on the left. Any sheet must contain part or all of one article only. Good office duplication copies are acceptable. Titles of chapters and paragraphs should appear clearly distinguished from the text.
Author produced (camera ready) copy is acceptable if typed on special sheets which are available from the Editor, and adherence to the Typing instructions (also available from the Editor) has been taken care of, is emphasized that camera ready text should be typed *single* space (i. e. with *no* space between the lines).
Complete text records on 5 1/4" floppy discs are also acceptable, if typed according to the instructions available from the Editor.

5. Equations
Mathematical equations inserted in the text must be clearly formulated in such a manner that there can be no possible doubt about meaning of the symbols employed.

6. Figures
The figures, if any, must be clearly numbered and their position in the text marked. They will be drawn in Indian ink on white paper or tracing paper, bearing in mind that they will be reduced to a width of either 7,5 or 15 (3 or 6") for printing. After scaling down, the normal lines ought to have a minimum thickness of 0,1 mm and maximum of 0,3 mm while lines for which emphasis is wanted can reach a maximum thickness of 0,5 mm. Labelling of the figures must be easy legible after reduction. It will be as far as possible placed across the width of the diagram from left to right. The height of the characters after scaling down must not be less than 1 mm. Photographs for insertion in the text will be well defined and printed on glossy white paper, and will be scaled down for printing to a width of 7,5 to 15 cm (3 to 6"). All markings on photographs are covered by the same recommendations as for figures. It is recommended that authors of communications accompany each figure or photograph with a descriptive title giving sufficent information on the content of the picture.

7. Tables
Tables of characteristics or values inserted in the text or appended to the article must be prepared in a clear manner, preferably as Camera Ready text. Should a table need several pages these must be kept together by sticking or other appropriate means in such a way as to emphasize the unity of the table.

8. Summaries
A summary of 10 to 20 typed lines written by the author in the English will precede and introduce each article.

9. Required information
Provide title, authors, affiliation, data of dispatch and a 100 to 250 word abstract on a separate sheet. Provide a separate sheet with exact mailing address for correspondence.

10. Reference
References must be listed alphabetically be the surname of the first author. List author(s) (with surname first), title, journal name, volume, year, pages for journal references, and authors(s), title, city, publisher, and year for the book references. Examples for article and book respectively:

Dawes, Robyn M. and Corrigan, Bernard: Linear models in decision making. Psychological Bulletin, **81** (1974), 95-106

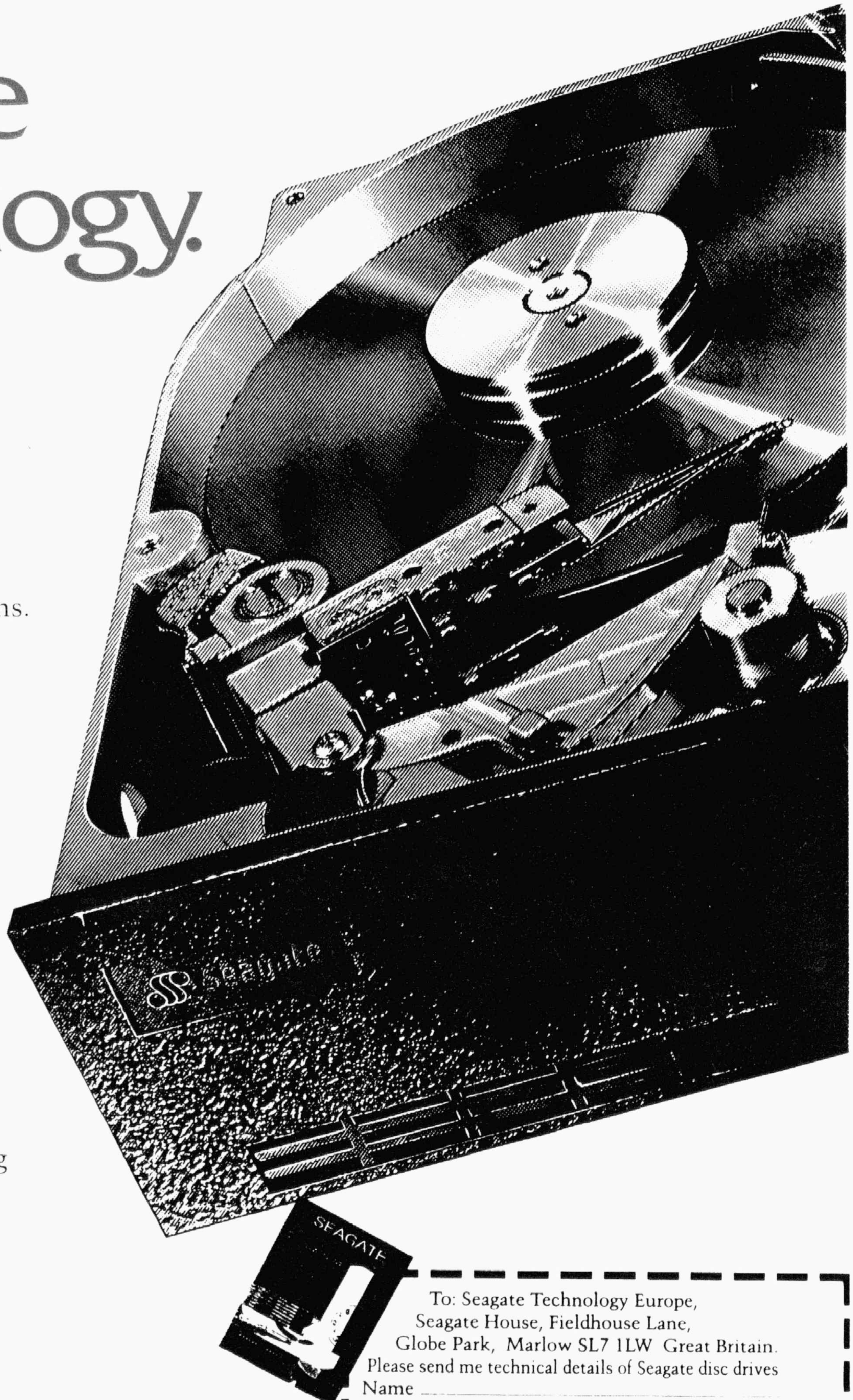
Brown, Robert G.: Statistical Forecasting for Inventory Control, New York: McGraw-Hill, 1959.

All references should be indicated in the manuscript by the author's surname followed by the year of publication (e. g., Brown, 1959).

11. Reprints
Each author will receive 25 free reprints of his article.

PD 3818

This is Seagate Technology.



Seagate's line of hard disc drives is packed with high technology. And every one is built to the highest quality and reliability standards in the industry.

And now, Seagate drives are available locally for all your Personal Computer applications.

Only Seagate can offer you full technical support, and a one-year warranty through our authorised representatives in your country.

Complete technical and interface details are included in the Seagate product brochures, which are free of charge to professional PC buyers and users. Simply use the coupon below to request your copies.

You'll soon see why Seagate has become the world's leading independent manufacturer of disc drives.

To: Seagate Technology Europe,
Seagate House, Fieldhouse Lane,
Globe Park, Marlow SL7 1LW Great Britain.
Please send me technical details of Seagate disc drives
Name
Job Title
Organisation
Address

Country
Type of business
Number of employees Number of PCs
☐ I use a PC ☐ I authorise the purchase of PCs
☐ I am a technical support manager

Seagate

Seagate Technology Europe
Seagate House, Fieldhouse Lane, Globe Park, Marlow SL7 1LW Great Britain
Tel: 0628 890366 Fax: 0628 890660 Telex: 846218 SEAGAT G