An application of neural networks in chemistry

V. KVASNIČKA

Department of Mathematics, Faculty of Chemical Technology, Slovak Technical University, CS-812 37 Bratislava

Received 7 March 1990

Dedicated to Professor L. Valko, DrSc., in honour of his 60th birthday

Basic definitions of the N-layer neural network are given. Partial derivatives of the objective functions with respect to the weight and threshold coefficients are derived. These derivatives are valuable for an adaptation process of the considered neural network, the resulting optimal weight and threshold coefficients are used in the forthcoming active process. The neural network may be applied as a classifier of objects. The theory is illustrated by an application of three-layer neural network to the classification of ^{13}C NMR chemical shifts of acyclic alkanes.

Данны основные определения N-слойной нейронной сети. Были выведены частичные производные объективных функций касающиеся весовых и пороговых коэффициентов. Эти производные имеют значения при процессе адаптации рассматриваемой нейронной сети. Полученные оптимальные весовые и пороговые коэффициенты применились в следующем активном процессе. Нейронная сеть может быть применена в виде классификатора объектов. Теория объясняется применением трех-слойной нейронной сети при классификации химических сдвигов ^{13}C ЯМР ациклических алканов.

I. Introduction

Neural networks [1—3] are computer or algorithmic systems derived from a simplified concept of brain in which a number of nodes, called the neurons, are interconnected in a netlike structure. A network is constructed with three or more layers of neurons: input neurons, output neurons, and often one or more layers of intermediate elements, called the hidden neurons (Fig. 1).

Each neuron receives input signals *via* one-way connections from preceding neurons, and each input is weighted by a variable weight parameter. If the sum of weighted inputs to a neuron exceeds a certain threshold coefficient, the neuron will send a signal to another neuron located in the juxtaposed higher layer.

Unlike conventional computers, neural networks are parallel in their structure and in the way they process information. A structure of network, in particular, the number of layers and the distribution of neurons among layers, is adjusted so that it is appropriate for the problem under study. The network is then put through a training (adaptation) process in which the weight and threshold coefficients are modified recursively by a learning algorithm, based on a "training set" of known data, until the weights and thresholds converge to fixed values. Assuming that the adaptation process was finished successfully, the formed network can be used to solve new problems, in a so-called active process. Unlike the standard computer (von Neumann's concept of the computer), knowledge is represented in neural network in a parallel fashion in the form of weight and threshold coefficients distributed throughout the system.

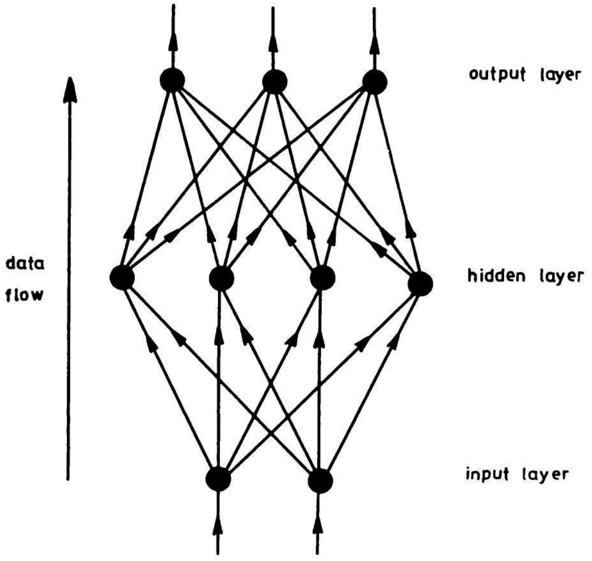

Fig. 1. Typical neural network composed of three layers.

Advantages of neural networks relative to conventional algorithms include: *1.* their *self-learning feature* and *2.* the *ability to generalize*. However, there is also a number of disadvantages. In particular, neural networks *1.* are *poor at mathematics, 2.* they sometimes *fail to give correct answer,* and *3.* they cannot *explain their predictions.*

Neural networks do give wrong answers especially when they have been "trained" in a configuration with inappropriate weight and threshold coefficients from which they cannot escape, or more frequently, when they have been set-up incorrectly. There may be a wrong assignment between the number of neurons in the network and the number of identifiable patterns in the data set.

Another possible reason for failure is an insufficient number of trials in the training process.

Neural networks are not a new "technology" [4]. The crucial point in this field is the concept of perceptron [5], widely used for many learning and pattern recognition algorithms. This concept was the subject of serious criticism [6], directed especially against its inability to solve certain more complex types of problems. The field of neural networks has seen a revival [7] in the 1980's after it was realized that this criticism was only relevant to very simple type of neural networks — perceptrons.

Applications of neural networks to chemistry have just to emerge [8]. The first ones [9, 10] are touching the problem of prediction of three-dimensional protein structure from data on amino acids. Recently, *Chemical and Engineering News* [8] reported a short review communication on a few initial applications of neural networks in organic chemistry; in particular, the distribution of products of nitration in a series of monosubstituted benzenes and the prediction of adverse drug effects have been described.

II. *N*-layer neural networks

We shall postulate [3] that a neural network consists of neurons that are vertically structured in N layers (where $N \geq 3$) (Fig. 2). The bottom (top) layer is called the *input* (*output*) *layer*. The layers which are ranged between the input and output layers are called *hidden layers*. Two neurons indexed by i and j, are connected by an oriented edge (i, j), outgoing from the neuron i and incoming to the neuron j, if and only if (iff) the vertices i and j belong to two distinct juxtaposed layers. Hence, a neuron from the layer p and a neuron from the layer q are connected by an oriented edge iff $q = p + 1$. Each edge (i, j), where i (j) belongs to the layer p $(p + 1)$ is evaluated by the *weight coefficient* $w_{ji}^{(p)}$. Each vertex i from the layer p (where $2 \leq p \leq N$) is evaluated by the *threshold coefficient* $\vartheta_i^{(p)}$. It means that the N-layer neural network may be considered as a special oriented graph the vertices of which are distributed over different layers; the edges exist only between vertices (neurons) that are from juxtaposed layers, and finally, its edges and vertices (except of vertices from the lowest input layer) are evaluated by weight and threshold coefficients, respectively.

The neurons from the same layer (indexed by p, where $1 \leq p \leq N$) are additionally evaluated by the so-called *activities* $x_1^{(p)}, x_2^{(p)}, \ldots, x_{n_p}^{(p)}$, where n_p is the number of neurons placed in the layer p. These activities constitute the *state vector* $\boldsymbol{x}^{(p)} = (x_1^{(p)}, x_2^{(p)}, \ldots, x_{n_p}^{(p)})$. The state vector $\boldsymbol{x}^{(1)}$ assigned to the input layer is called the *input state vector*, and similarly, the state vector $\boldsymbol{x}^{(N)}$ assigned to the output layer is called the *output state vector*.

The state vectors $\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)}, \ldots, \boldsymbol{x}^{(N)}$ for a given neural network and a given input state vector $\boldsymbol{x}^{(1)}$ are calculated recurrently as follows

$$x_i^{(t)} = f\left(\sum_j w_{ij}^{(t-1)} x_j^{(t-1)} + \vartheta_i^{(t)}\right) \tag{1}$$

for $t = 2, 3, \ldots, N$ and $i = 1, 2, \ldots, n_t$. The *transfer function* $f(\xi)$ is a positive and monotonically increasing function which fulfills asymptotic conditions $f(\xi) \to 1$ for $\xi \to \infty$ and $f(\xi) \to 0$ for $\xi \to -\infty$. For instance, these requirements are simply met if the transfer function $f(\xi)$ is given as

$$f(\xi) = \frac{1}{1 + \mathrm{e}^{-\xi}} \tag{2}$$

its first derivative being determined by $f'(\xi) = f(\xi)\,[1 - f(\xi)]$. Applying successively the formula (*1*) we calculate from an input state vector $\boldsymbol{\alpha} = \boldsymbol{x}^{(1)}$ the state vector $\boldsymbol{x}^{(2)}$, then from $\boldsymbol{x}^{(2)}$ we calculate $\boldsymbol{x}^{(3)}$, and so on; finally we arrive at the

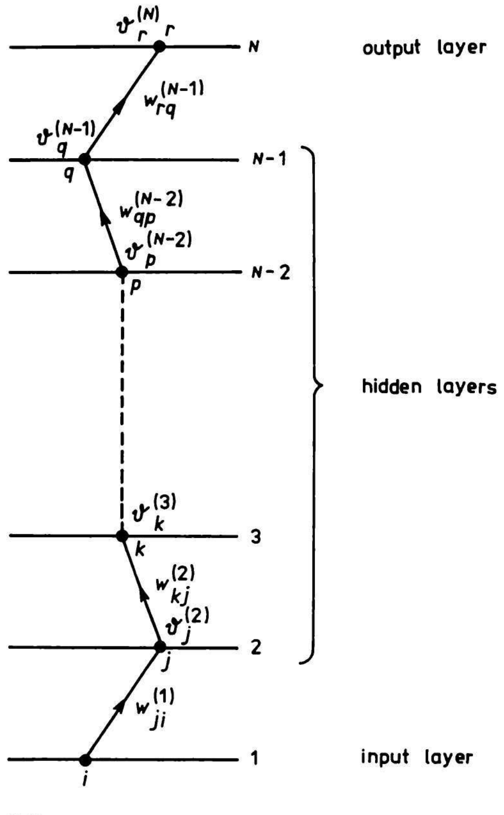

Fig. 2. General scheme of N-layer neural network. A neuron indexed by p and belonging to the i-th layer (for $2 \le i \le N$) is evaluated by the threshold coefficient $\vartheta_p^{(i)}$. Similarly, an edge which connects two neurons p and q belonging to a pair of juxtaposed layers indexed by i and $i-1$ (where $2 \le i \le N$), respectively, is evaluated by the weight coefficient $w_{qp}^{(i)}$.

output state vector $\boldsymbol{x}^{(N)}$ of the output layer. For fixed weight and threshold coefficients $w_{ij}^{(t)}$ and $\vartheta_i^{(t)}$ such a successive calculation of the output state vector $\boldsymbol{x}^{(N)}$ from the input state vector $\boldsymbol{x}^{(1)}$ is called the *active process* of neural network.

In general, a neural network with fixed structure and weight and threshold coefficients may formally be considered as a mapping F which assigns to an input state vector $\boldsymbol{\alpha} = \boldsymbol{x}^{(1)}$ an output state vector $\boldsymbol{x}^{(N)}$

$$\boldsymbol{x}^{(N)} = F(\boldsymbol{x}^{(1)}) \tag{3}$$

Its explicit analytical form may be simply constructed by successive making use of (*1*), where the transfer function f is specified by (*2*).

Now we focus our attention to the so-called *adaptation* (*learning*) *process* of a neural network. Thus, for a given pair $\boldsymbol{\alpha}/\boldsymbol{\beta}$ of input and output state vectors, we try to find such weight and threshold coefficients that a neural network response $\boldsymbol{x}^{(N)}$ to the given input state vector $\boldsymbol{\alpha}$ would be "closely related" to the prescribed output state vector $\boldsymbol{\beta}$. There exist many different approaches how to realize the above-mentioned vague "close relationship". One of them is to minimize the following *objective function*

$$E = \frac{1}{2}(\boldsymbol{x}^{(N)} - \boldsymbol{\beta})^2 = \frac{1}{2}\sum_k (x_k^{(N)} - \beta_k)^2 \tag{4}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_{n_N})$. It means that a goal of our adaptation process is finding such weight and threshold coefficients that minimize the objective function (*4*). The so-called *backpropagation strategy* [3] of the adaptation process of an N-layer neural network consists in the successive calculation of partial derivatives $\partial E/\partial w_{ji}^{(t)}$ and $\partial E/\partial \vartheta_j^{(t)}$ in going from the top output layer to the bottom input layer. In order to evaluate these partial derivatives we have to know the partial derivatives $\partial x_k^{(N)}/\partial w_{ji}^{(t)}$ and $\partial x_k^{(N)}/\partial \vartheta_j^{(t)}$. After simple but tedious manipulations we get

$$\frac{\partial x_k^{(N)}}{\partial \vartheta_j^{(N)}} = \delta_{jk}\, x_k^{(N)}\,(1 - x_k^{(N)}) \tag{5a}$$

$$\frac{\partial x_k^{(N)}}{\partial \vartheta_j^{(t)}} = \left(\sum_l \frac{\partial x_k^{(N)}}{\partial w_{lj}^{(t)}}\, w_{lj}^{(t)}\right)(1 - x_j^{(t)}) \tag{5b}$$

for $t = N-1, N-2, ..., 2$, and

$$\frac{\partial x_k^{(N)}}{\partial w_{ji}^{(t)}} = \frac{\partial x_k^{(N)}}{\partial \vartheta_j^{(t+1)}}\, x_i^{(t)} \tag{6}$$

for $t = N - 1, N - 2, \ldots, 1$. The symbol δ_{jk} corresponds to Kronecker's delta, $\delta_{jk} = 1$ for $j = k$ and $\delta_{jk} = 0$ for $j \neq k$. The partial derivatives of the objective function E determined by the relation (*4*) are

$$\frac{\partial E}{\partial w_{ji}^{(t)}} = \sum_k (x_k^{(N)} - \beta_k) \frac{\partial x_k^{(N)}}{\partial w_{ji}^{(t)}} \tag{7a}$$

$$\frac{\partial E}{\partial \vartheta_j^{(t)}} = \sum_k (x_k^{(N)} - \beta_k) \frac{\partial x_k^{(N)}}{\partial \vartheta_j^{(t)}} \tag{7b}$$

Introducing (*5a, 5b*) and (*6*) into (*7a, 7b*) we get the final formulae for the first partial derivatives of the objective function E

$$\frac{\partial E}{\partial \vartheta_j^{(N)}} = (x_j^{(N)} - \beta_j)\, x_j^{(N)} (1 - x_j^{(N)}) \tag{8a}$$

$$\frac{\partial E}{\partial \vartheta_j^{(t)}} = \left(\sum_l \frac{\partial E}{\partial w_{lj}^{(t)}} w_{lj}^{(t)} \right) (1 - x_j^{(t)}) \tag{8b}$$

for $t = N - 1, N - 2, \ldots, 2$, and

$$\frac{\partial E}{\partial w_{ji}^{(t)}} = \frac{\partial E}{\partial \vartheta_j^{(t+1)}} x_i^{(t)} \tag{9}$$

for $t = N - 1, N - 2, \ldots, 1$. We can see that applying these formulae we are able to calculate successively the partial derivatives $\partial E/\partial w_{ji}^{(t)}$ and $\partial E/\partial \vartheta_j^{(t)}$ going step-by-step from the top to the bottom of the neural network.

The partial derivatives (*8, 9*) are useful in that they minimize the objective function E by making use of a version of the well-known gradient method [11]. In our actual applications variable metric method [11, 12] is performed satisfactorily when the zero-step initial values of weight and threshold coefficients were randomly generated from the interval $(-1, 1)$.

The above adaptation process of neural network may be simply generalized also for more than one pair of vectors $\boldsymbol{\alpha}/\boldsymbol{\beta}$, *i.e.* for p pairs of vectors $\boldsymbol{\alpha}_1/\boldsymbol{\beta}_1$, $\boldsymbol{\alpha}_2/\boldsymbol{\beta}_2$, ..., $\boldsymbol{\alpha}_p/\boldsymbol{\beta}_p$. Then the objective function is determined by

$$E = \sum_{i=1}^{p} E^{(i)} \tag{10a}$$

$$E^{(i)} = \frac{1}{2}(\boldsymbol{x}_i^{(N)} - \boldsymbol{\beta}_i)^2 \tag{10b}$$

where $\boldsymbol{x}_i^{(N)}$ is the output state vector of the neural network corresponding to the preselected i-th input state vector $\boldsymbol{\alpha}_i$, and $\boldsymbol{\beta}_i$ is the required response of the neural network assigned to $\boldsymbol{\alpha}_i$. The partial derivatives of E are then equal to the sum of partial derivatives of $E^{(i)}$ evaluated by (*8, 9*).

For $N = 2$ the concept of neural network is reduced to a concept closely related to the so-called *perceptron* [5, 6]. The formulae (*8*) and (*9*) remain valid even in this rudimentary case of neural networks, and we get

$$\frac{\partial E}{\partial \vartheta_j} = (x_j^{(2)} - \beta_j)\, x_j^{(2)}\, (1 - x_j^{(2)}) \tag{11a}$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial \vartheta_j}\, x_i^{(1)} \tag{11b}$$

where the upper indices of threshold and weight coefficients were omitted as unimportant. Assuming that the output layer contains only one neuron and that the output state vector $\boldsymbol{x}^{(2)} = (x_1^{(2)})$ is composed of a dichotomic entry equal either to one or to zero (when the transfer function $f(\xi)$ is strongly nonlinear, *i.e.* $f(\xi) = 1$ for $\xi > 0$ and $f(\xi) = 0$ for $\xi < 0$), then it is easy to show that this two-layer neural network provides the so-called linear learning machine (or pattern recognition method), a dichotomic (two classes) classifier often used in chemistry [13].

In order to get a deeper insight in an adapted neural network (*i.e.* the weight and threshold coefficients were already chosen) we introduce the so-called *sensitivities* of the neural network. Let us assume that a given input state vector $\boldsymbol{x}^{(1)}$ is changed as $\boldsymbol{x}^{(1)} \rightarrow \boldsymbol{x}^{(1)} + \Delta\boldsymbol{x}^{(1)}$, where the "perturbation" $\Delta\boldsymbol{x}^{(1)}$ is composed of entries that are of the first-order smallness with respect to entries of $\boldsymbol{x}^{(1)}$. Then for each single layer the corresponding state vectors $\boldsymbol{x}^{(t)}$ (for $2 \leq t \leq N$) are changed as $\boldsymbol{x}^{(t)} \rightarrow \boldsymbol{x}^{(t)} + \Delta\boldsymbol{x}^{(t)}$. Entries of $\Delta\boldsymbol{x}^{(t)}$ correspond to a response of the neural network to the "perturbation" $\Delta\boldsymbol{x}^{(1)}$; when going successively from the second layer to higher layers we may trace a propagation of the input "perturbation" $\Delta\boldsymbol{x}^{(1)}$ throughout the whole neural network. The responses $\Delta\boldsymbol{x}^{(t)}$, for $2 \leq t \leq N$, are well approximated (up to the first order) by

$$\Delta\boldsymbol{x}_i^{(t)} \approx \sum_j \frac{\partial x_i^{(t)}}{\partial x_j^{(1)}}\, \Delta x_j^{(1)} \tag{12}$$

where the partial derivatives $S_{ij}^{(t)} = \partial x_i^{(t)}/\partial x_j^{(1)}$ will be called the *t-layer sensitivities* of neural network. These entities are simply calculated from the expressions (*1*, *2*)

$$S_{ij}^{(t)} = x_i^{(t)} (1 - x_i^{(t)}) \sum_k w_{ik}^{(t-1)} S_{kj}^{(t-1)} \tag{13}$$

where $t = 2, 3, ..., N$ and $S_{kj}^{(1)} = \delta_{kj}$. The sensitivities are intuitively interpreted as follows: if entries $S_{ij}^{(t)}$, for a fixed $2 \le t \le N$, have "relatively" very small numerical values, then the state vector $\boldsymbol{x}^{(t)}$ is "relatively" insensitive to the "perturbation" $\Delta \boldsymbol{x}^{(1)}$; increasing values of $S_{ij}^{(t)}$ indicate a greater sensitivity of neural network to $\Delta \boldsymbol{x}^{(1)}$, *i.e.* a small change in an entry of the input state vector $\boldsymbol{x}^{(1)}$ may cause a considerable change in some entries of the state vector $\boldsymbol{x}^{(t)}$.

III. *N*-layer neural network as classifier

Let us consider a universe $\mathscr{U} = \{o_1, o_2, ...\}$ composed of objects $o_1, o_2, ...$. Two mappings Φ and Ψ assign to each object $o \in \mathscr{U}$ the so-called *descriptor vector* $\boldsymbol{d}$ and the *category vector* $\boldsymbol{c}$, respectively, *i.e.* $\boldsymbol{d} = \Phi(o)$ and $\boldsymbol{c} = \Psi(o)$. The descriptor vector $\boldsymbol{d} = (d_1, d_2, ...)$ assigned to $o \in \mathscr{U}$ is composed of real entries that are corresponding to preselected "descriptors". The latter characterize the "structural" parameters of objects from $\mathscr{U}$. Analogously, the category vector $\boldsymbol{c} = (c_1, c_2, ...)$ of $o \in \mathscr{U}$ is composed of real entries corresponding to chosen properties — categories of the given object. The classification process of objects from $\mathscr{U}$ is based on the construction of a function, which assigns a category vector $\boldsymbol{c}$ to the descriptor vector $\boldsymbol{d}$, *i.e.*

$$\boldsymbol{c} = G(\boldsymbol{d}) \tag{14}$$

for each $o \in \mathscr{U}$, where $\boldsymbol{d} = \Phi(o)$ and $\boldsymbol{c} = \Psi(o)$.

A neural network, used as a classifier of objects from the universe $\mathscr{U}$, consists of an identification of the descriptor/category vector with the input/output state vector. It means that for a given neural network a functional form of the function G from (*14*) is known, the only "variable" parameters specifying its actual analytical form are weight and threshold coefficients. In order to select these coefficients we divide the universe $\mathscr{U}$ in two disjoint sets $\mathscr{T}$ (called the *training set*) and its complement $\mathscr{U} \backslash \mathscr{T}$. The function G is "approximated" by the function F from (*3*). The weight and threshold coefficients of neural network are determined in such a way that the output state vector $\boldsymbol{x}^{(N)} = F(\boldsymbol{x}^{(1)})$ is "closely related" to $\boldsymbol{c} = G(\boldsymbol{d})$, where $\boldsymbol{d} = \boldsymbol{x}^{(1)} = \Psi(o)$, for each $o \in \mathscr{T}$. In our approach the above vague term "closely related" is realized by the minimization of the objective function (*10*). Furthermore, assuming that this adaptation process was successfully carried out, the same function F could be used also for the classification of an object outside of the training set, *i.e.* from the complement $\mathscr{U}\backslash\mathscr{T}$. This "extrapolation" of the function F, initially constructed in the course of the adaptation process of neural network for objects from the training set $\mathscr{T}$, represents a principal step in any application of neural network acting as a classifier of objects from the universe $\mathscr{U}$. Intuitively speaking, the training set $\mathscr{T}$ should be selected from the universe $\mathscr{U}$ very carefully, its cardinality (*i.e.* number of objects) is usually much smaller than the cardinality of the universe $\mathscr{U}$, and moreover, the objects belonging to $\mathscr{T}$ must be sufficiently representative of all objects from $\mathscr{U}$.

IV. Application

^{13}C NMR chemical shifts of acyclic alkanes

The applicability and effectiveness of neural-network approach will be illustrated by three-layer network (composed of nine input neurons, five hidden neurons, and one output neuron) used as a classifier of ^{13}C NMR chemical shifts [14] of acyclic alkanes. An alkane molecule with a preselected carbon atom (corresponding to that one to which the chemical shift is related) may be graph-theoretically represented as a rooted tree [15] (Figs. 3—5). This rooted tree will be described by the following nine-dimensional descriptor vector $\boldsymbol{d} = (d_1, d_2, ..., d_9)$ composed of nonnegative integers, d_i is the number of those distinct paths of the length i that are beginning at the root ($1 \leq i \leq 5$), and d_j number of adjacent $(j - 5)$-ary carbon atoms with the root ($6 \leq j \leq 9$). Examples of descriptors are given in Tables 1 and 2. Following *Grant* and *Paul* [16], the chemical shifts δ are empirically calculated as follows

$$\delta = -2.3 + \sum_{i=1}^{5} d_i\varepsilon_i + \sum_{j=6}^{9} d_j S(d_1, j-5) \qquad (15)$$

where d_i and d_j are entries of the descriptor vector $\boldsymbol{d}$, ε_i is an increment of the carbon atom of the distance i from the considered (root) carbon atom, and $S(p, q)$ is a steric increment describing an influence of the q-ary adjacent atom on the considered (root) p-ary carbon atom. Actual numerical values of these increments are given in literature [14, 16]. The empirical values of chemical shifts evaluated by (*15*) of some alkanes are listed in Tables 3 and 2.

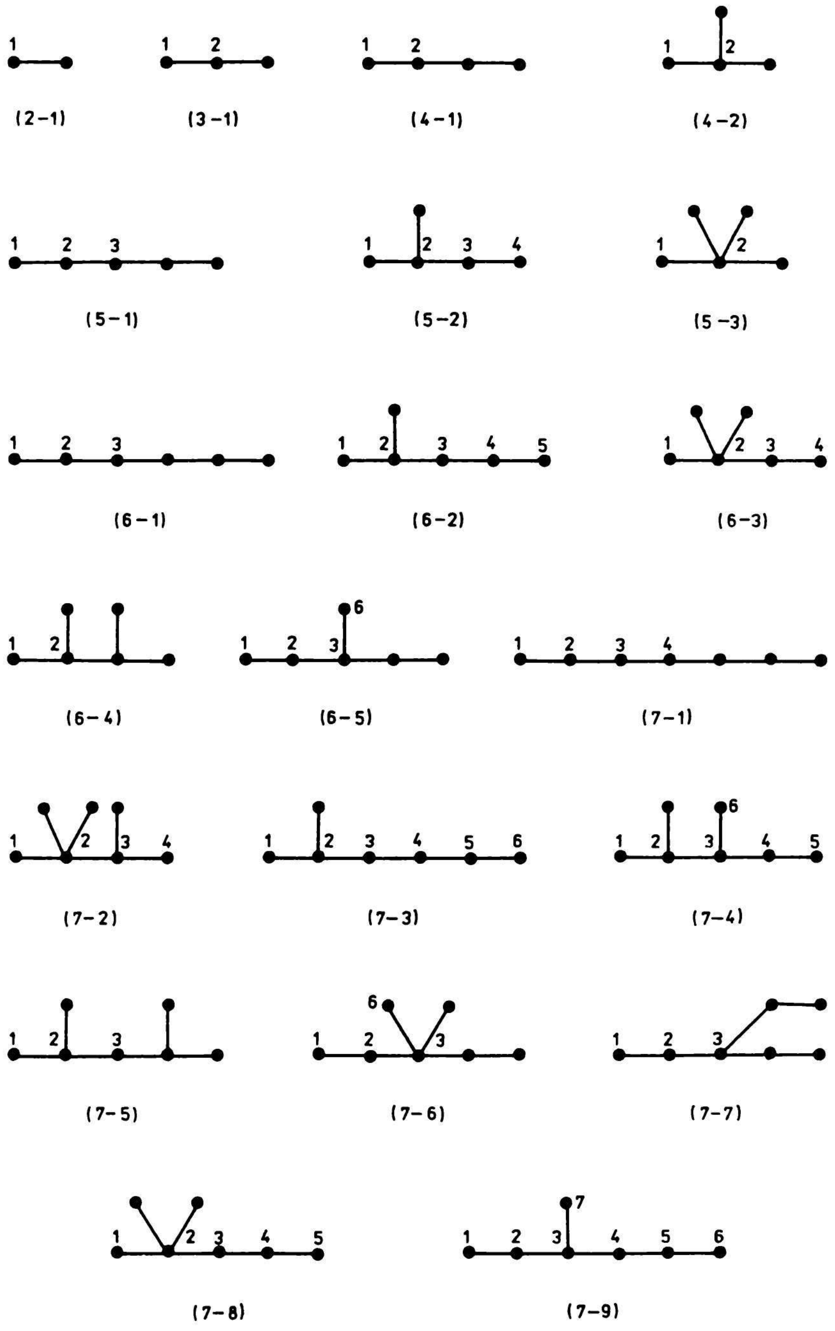

Fig. 3. Schematic plots of all alkanes through C_7.

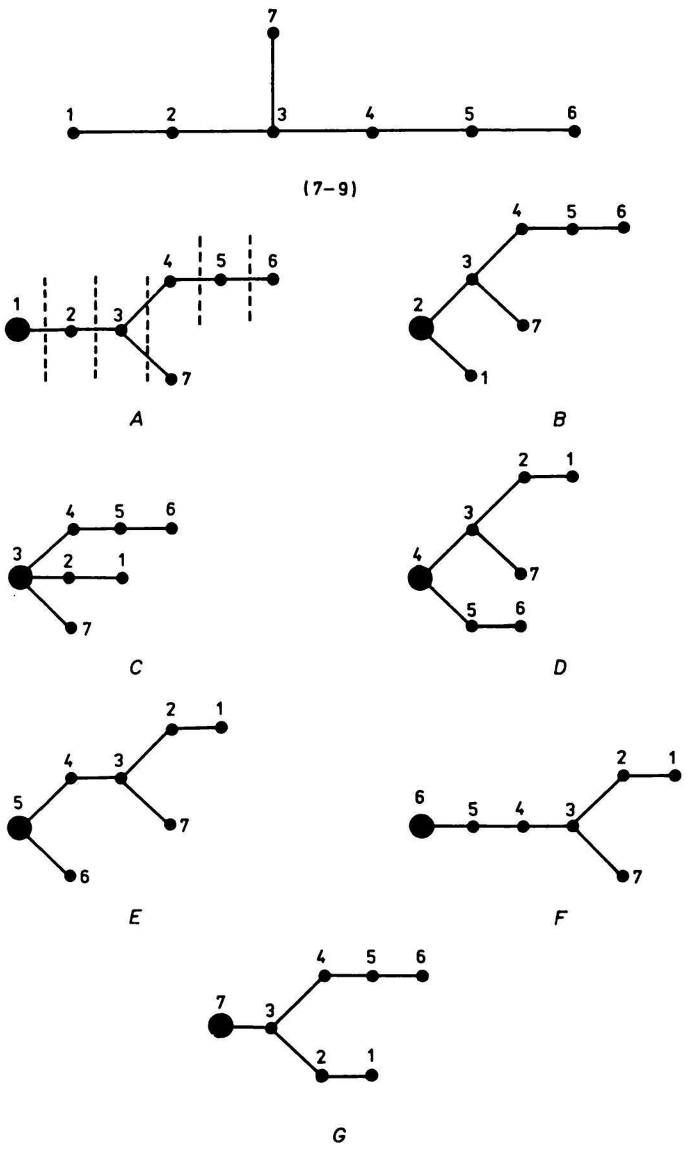

Fig. 4. All rooted trees assigned to the alkane indexed by 7—9 in Fig. 3. The first five entries d_1 to d_5 of the nine-dimensional descriptor vector $\boldsymbol{d}$ are simply constructed as a number of crossing points for cut (dashed) lines, *e.g.* for the tree A we get $d_1 = d_2 = 1$, $d_3 = 2$, and $d_4 = d_5 = 1$.

We emphasize that the chosen form of the descriptor vector $\boldsymbol{d}$ does not describe unambiguously the acyclic alkanes (graphs), *i.e.* it is easy to show two nonisomorphic trees with the same descriptor vector. The form of descriptor vector was suggested in such a way as to be simple and moreover its entries may be immediately related to chemical shifts *via* the formula (*15*).

The training set $\mathscr{T}$ for the adaptation process is composed of all topologically nonequivalent rooted trees assigned to all alkanes through to C_7 (Fig. 3). It contains 76 objects — rooted trees which are determined by nine-dimensional descriptors; the corresponding category vectors are composed of one real entry from the open interval (0, 1). Here we have to specify more precisely the actual

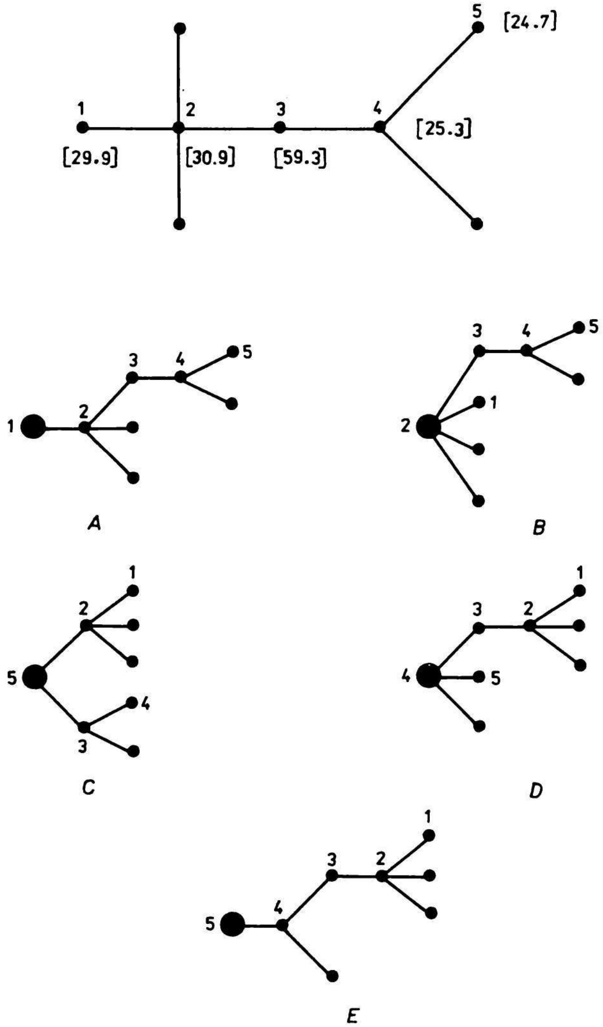

Fig. 5. Schematic graphs of an alkane C_8 used as an example of active process of the adapted neural network. In brackets are given the corresponding chemical shifts.

Table 1

Descriptors of rooted trees of the alkane indexed by (7—9) in Fig. 4

Alkane	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
A	1	1	2	1	1	0	1	0	0
B	2	2	1	1	0	1	0	1	0
C	3	2	1	0	0	1	2	0	0
D	2	3	1	0	0	0	1	1	0
E	2	1	2	1	0	1	1	0	0
F	1	1	1	2	1	0	1	0	0
G	1	2	2	1	0	0	0	1	0

Table 2

Illustrative example of active process of adapted neural network for the C_8 alkane in Fig. 5

Alkane	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	δ_{exp}	$\delta_{\text{incr.}}$	$\delta_{\text{n.n.}}$
A	1	3	1	2	0	0	0	0	1	29.9	29.7	30.0
B	4	1	2	0	0	3	1	0	0	30.9	25.6	29.0
C	2	5	0	0	0	0	0	1	1	59.3	52.9	58.5
D	3	1	3	0	0	2	1	0	0	25.3	23.2	23.6
E	1	2	1	3	0	0	0	1	0	24.7	22.9	23.2

meaning of the concept of category assigned to chemical shifts of carbon atoms. As follows from (*2*), the transfer function f maps the set $\mathscr{R}$ of real numbers onto the open interval (0, 1), *i.e.*

$$f: \mathscr{R} \rightarrow (0, 1)$$

This means that the entries of output state vector $\boldsymbol{x}^{(N)}$ belong merely to the interval (0, 1). That is, a category vector $\boldsymbol{c}$ should be also composed merely of real numbers from the open interval (0, 1). Since the chemical shifts of carbon atoms are, let us say, ranged within $5 \leq \delta \leq 50$, this interval should be compressed by an analogue of (*2*) to a subinterval of (0, 1). Such a mapping was in our studies realized by

$$y = g(x) = \frac{1}{1 + e^{(-ax - b)}} \qquad (16)$$

Table 3

^{13}C NMR chemical shifts of alkanes

Alkane[a]	Type	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7
(2—1)	Exp.[b]	6.5						
	Incr.[c]	6.8						
	N.n.[d]	6.3						
(3—1)	Exp.	16.1	16.3					
	Incr.	16.2	15.9					
	N.n.	16.0	16.3					
(4—1)	Exp.	13.1	24.9					
	Incr.	13.7	25.3					
	N.n.	13.4	25.0					
(4—2)	Exp.	24.6	23.3					
	Incr.	24.5	25.0					
	N.n.	24.5	23.3					
(5—1)	Exp.	13.5	22.2	34.1				
	Incr.	14.0	22.8	34.7				
	N.n.	13.8	22.4	34.1				
(5—2)	Exp.	21.9	29.9	31.6	11.5			
	Incr.	22.0	30.7	32.2	11.2			
	N.n.	22.0	30.0	31.8	10.6			
(5—3)	Exp.	27.4	31.4					
	Incr.	31.6	28.1					
	N.n.	27.4	31.4					
(6—1)	Exp.	13.7	22.7	31.7				
	Incr.	14.1	23.1	32.2				
	N.n.	13.6	22.6	31.7				
(6—2)	Exp.	22.7	27.9	41.9	20.8	14.3		
	Incr.	22.3	28.2	41.6	20.3	14.3		
	N.n.	22.4	27.9	41.8	20.1	14.3		
(6—3)	Exp.	28.7	30.3	36.1	8.6			
	Incr.	29.1	30.6	36.6	8.7			
	N.n.	29.0	30.3	36.0	8.1			
(6—4)	Exp.	19.2	34.0					
	Incr.	19.5	34.3					
	N.n.	19.3	33.8					
(6—5)	Exp.	11.4	29.4	36.8			18.7	
	Incr.	11.5	29.7	36.4			19.5	
	N.n.	11.1	28.9	36.7			19.3	
(7—1)	Exp.	13.7	22.6	32.0	29.0			
	Incr.	14.1	23.2	32.5	29.7			
	N.n.	13.6	22.6	31.9	29.3			
(7—2)	Exp.	27.0	32.7	37.9	17.7			
	Incr.	26.6	33.4	38.2	17.0			
	N.n.	26.2	32.7	37.9	17.1			

Table 3 (Continued)

Alkane[a]	Type	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7
(7—3)	Exp.	22.4	28.1	38.9	29.7	23.0	13.6	
	Incr.	22.4	28.5	39.1	29.7	23.4	14.2	
	N.n.	22.4	28.2	38.9	29.3	22.9	13.6	
(7—4)	Exp.	20.0	31.9	40.6	26.8	11.6	17.0	
	Incr.	19.8	31.8	40.0	27.2	11.8	17.0	
	N.n.	19.6	32.1	40.3	26.2	11.5	17.1	
(7—5)	Exp.	22.7	25.7	49.0				
	Incr.	23.4	25.7	48.5				
	N.n.	22.9	25.6	50.3				
(7—6)	Exp.	7.7	33.4	32.2			25.6	
	Incr.	9.0	34.1	33.1			26.6	
	N.n.	8.5	33.5	32.3			26.2	
(7—7)	Exp.	10.5	25.2	42.4				
	Incr.	11.8	27.2	42.1				
	N.n.	11.5	26.2	42.3				
(7—8)	Exp.	29.5	30.6	47.3	18.1	15.1		
	Incr.	29.4	28.1	46.0	17.8	14.6		
	N.n.	29.5	30.5	47.4	18.4	14.8		
(7—9)	Exp.	10.6	29.5	34.3	39.0	20.2	13.9	18.8
	Incr.	11.6	30.0	33.9	39.1	20.6	14.4	19.8
	N.n.	10.6	29.5	34.4	38.9	20.3	14.1	19.6

a) Alkanes are indexed as in Fig. 3. *b*) Experimental values of chemical shifts measured with respect to TMS [14], *c*) calculated values by making use of formula (*15*) and *d*) produced by neural network.

where the constants a and b are determined as follows

$$a = \frac{A_{\min} - A_{\max}}{x_{\min} - x_{\max}} \qquad b = -ax_{\min} - A_{\min}$$

with

$$A_{\min} = \ln \frac{y_{\min}}{1 - y_{\min}} \qquad A_{\max} = \ln \frac{y_{\max}}{1 - y_{\max}}$$

The entries $x_{\min}$ and $x_{\max}$ correspond to minimal and maximal values, respectively, of the chemical shifts (in our case we put $x_{\min} = 5$ and $x_{\max} = 50$), whereas the entries $y_{\min}$ and $y_{\max}$ are minimal and maximal values of the "compressed" chemical shifts (we put $y_{\min} = 0.05$ and $y_{\max} = 0.95$); for these actual values of minimal and maximal entries the constants a and b are $a = 0.1308640$ and $b = -3.5987588$. An inverse transformation of (*16*) is

Table 4

Optimal values of weight w_{ji} and threshold ϑ_j coefficients*

				$w_{ji}^{(1)}$					$w_{ji}^{(2)}$	$\vartheta_j^{(2)}$	$\vartheta_j^{(3)}$
1.601	0.165	−0.583	0.163	0.861	−0.585	0.508	0.796	−0.003	12.688	−4.508	2.233
0.740	0.574	−0.566	0.173	0.872	0.173	1.037	0.716	−0.530	−12.474	−4.700	
1.105	−0.435	0.010	−0.029	−0.175	1.647	−1.429	1.052	−0.886	7.487	−0.808	
1.344	−0.736	−5.790	−2.179	−0.234	0.416	−2.856	−1.268	1.329	− 4.041	0.449	
0.482	−0.755	0.123	−0.032	−0.029	1.112	−0.957	0.767	−0.331	−10.784	1.132	

* Objective function $E = 2.615 \times 10^{-3}$.

$$x = g^{-1}(y) = \frac{1}{a}\left(\ln \frac{y}{1-y} - b\right) \qquad (17)$$

This function is used for a "back" transformation of the output categories of neural network to actual chemical shifts.

The optimal values of weight and threshold coefficients resulting from the adaptation process are given in Table 4. The values of NMR chemical shifts determined by trained neural network are listed in Table 3. This neural network was also used for the evaluation of chemical shifts of alkanes which do not belong to the training set $\mathscr{T}$ (the active process of neural network already adapted), *e.g.* 2,2,4-trimethylheptane was classified (Fig. 5 and Table 2).

From simple inspection of Tables 2 and 3 one may conclude that the used neural network provides chemical shifts that are more closely related to their experimental values than those ones calculated empirically by eqn (*15*).

V. Conclusion

We have demonstrated that the approach based on neural networks offers simple tool potentially well suited for classification of objects (*i.e.* molecular systems) with respect to their properties. In order to "translate" the molecular structure parameters to a code useful as an input of neural networks, the parameters are rewritten in a form of descriptor vectors. The neural network is adapted in such a way that an output state vector (resulting as an image of the descriptor) is composed of entries — categories that are closely related to the properties of molecular system under study. The most important step of neural--network approach is the so-called active process, where already adapted neural network is used as a classifier of molecular systems (objects) outside the training set, the mapping (*3*) is "extrapolated" to molecular systems that were not used in the adaptation process. The results obtained by other authors [8—10] and also the results presented in this communication indicate that the neural networks are able to give predictions not available in any other way. Neural--network approaches do not replace other forms of computing predictions, but they promise to be a useful tool for approaching computationally problems that would not be satisfactorily solved by standard numerical methods.

References

1. *IEEE First Conference on Neural Networks*, Vols. *1*—*4*. San Diego, California, June 1987.
2. Eckmiller, R. and Malsburg, C.v.D., *Neural Computers*. Springer-Verlag, Berlin, 1988.

3. Rumelhart, D. E. and McClelland, J. L., *Parallel Distributed Processes*, Vols. *1*, *2*. MIT Press, Cambridge, Mass., 1986.
4. McCulloch, W. S. and Pitts, W., *Bull. Math. Biophys. 5*, 115 (1943).
5. Rosenblatt, F., *Principles of Neurodynamics*. Spartan Books, Washington D.C., 1962.
6. Minsky, M. L. and Papert, S. A., *Perceptrons*. MIT Press, Cambridge, Mass., 1969.
7. Hopfield, J. J., *Proc. Natl. Acad. Sci. U.S.A. 79*, 2554 (1982).
8. Borman, S., *Chem. Eng. News*. April 24, 1989, p. 24.
9. Ningh Qian and Sejnowski, T. J., *J. Mol. Biol. 202*, 865 (1988).
10. Holley, L. H. and Karplus, M., *Proc. Natl. Acad. Sci. U.S.A. 86*, 152 (1989).
11. Polak, E., *Computational Methods in Optimization*. Academic Press, New York, 1971.
12. Fletcher, R. and Powell, M. J. D., *Comput. J. 6*, 163 (1963).
13. Varmuza, K., *Pattern Recognition in Chemistry*. Springer-Verlag, Berlin, 1980.
14. Kalinowski, H.-O., Berger, S., and Braun, S., *^{13}C NMR Spektroskopie*. G. Thieme Verlag, Stuttgart, 1984.
15. Harary, F., *Graph Theory*. Addison—Wesley Publishing Co., Reading, Mass., 1969.
16. Grant, D. M. and Paul, E. G., *J. Am. Chem. Soc. 86*, 2984 (1964).

Translated by V. Kvasnička