Application of Recurrent Neural Networks in Chemistry. Prediction and Classification of ^{13}C NMR Chemical Shifts in a Series of Monosubstituted Benzenes

Vladimír Kvasnička,* Štěpán Sklenák, and Jiří Pospíchal

Department of Mathematics, Slovak Technical University, 81237 Bratislava, Czechoslovakia

Received June 10, 1992

The recurrent neural network is a feed-forward network ascribed to a parent neural network with feed-back connections (or in another term, oriented cycles). Its adaptation is performed by an analog of the standard back-propagation adaptation method. The recurrent neural network approach is illustrated by prediction and classification of ^{13}C NMR chemical shifts in a series of monosubstituted benzenes. The descriptors (input activities) of functional groups are determined by 11 nonnegative integers that correspond to numbers of appearance of some substructural features in the corresponding molecular graphs. The obtained results indicate that these descriptors properly describe the basic physical and chemical nature of functional groups.

INTRODUCTION

Recent progress[1,2] of the neural network paradigm, mainly the layered feed-forward neural networks adapted by the back-propagation strategy,[3] offers new mathematical and computational tools equipped by learning features. Their impact to chemistry[4] is manifested not only by effective correlations between molecular structure and activity but also by algorithms for building expert systems that, in turn, classify chemical reactions semiquantitatively or qualitatively.

Recently, Maggiora et al.[5] have published a very interesting computational look at general properties of the feed-forward neural networks. Since these are analytically very complicated, there is a small chance to derive conclusions that are satisfied for all of them, irrespective of their structure. In order to overcome this serious drawback of their mathematical complexity, it is of great importance to carry out some computer experiments for better understanding of their properties. Maggiora et al.[5] have focused on a number of problems that arise in essentially all neural-network applications and are especially important in chemical applications of structure–activity and/or structure–property relationships. Moreover, they gave special emphasis on the critical issues of small data sets and noisy data that plague almost all chemical applications of neural networks.

The purpose of this paper is to study the so-called recurrent neural networks[3,6] (assigned to neural networks with feed-back connections) and demonstrate their simple applicability for classification and prediction of ^{13}C NMR chemical shifts in a series of monosubstituted benzenes. Recently, we have studied[7] the same application by the neural networks that are composed of feed-back connections or graph-theoretically, oriented cycles of connections. For these neural networks the standard back-propagation adaptation is inapplicable; the hidden and output activities are determined by a string of coupled nonlinear equations. An updating of threshold and weight coefficients must be done very carefully to avoid inappropriate displacement of activities from their current equilibrium values. The approach of recurrent neural network[3,6] allows us to overcome this very serious difficulty of neural networks with feed-back connections. We assign to a parent neural network with feed-back connections the so-called recurrent neural network, which is of the standard feed-forward type. It corresponds to the simple iterative solution of the above-mentioned string of equations that determines the activities of hidden and output neurons. After a finite number of the iterative steps the process of solution is stopped, and then the activities of the last kth layer correspond to the kth iterative solution.

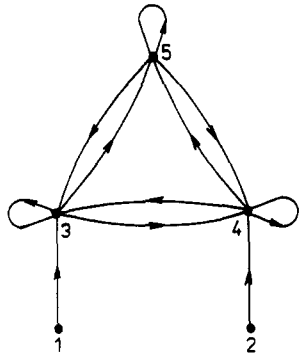

Figure 1. Illustrative example of simple parent neural network composed of two input neurons (1 and 2), two hidden neurons (3 and 4), and one output neuron (5). The hidden and output neurons induce in the parent network a complete subgraph, where every two neurons are adjacent by two oppositely oriented connections and each neuron is adjacent with the oriented loop that starts and ends at the neuron.

The recurrent neural networks, adapted by the standard back-propagation strategy, overcome the divergence difficulties of networks with feed-back connections. Their potential applicability in chemistry consists of a possibility to improve classification of patterns described by complex descriptors. The significance of descriptors for prediction of properties is reflected more adequately by feed-back interactions between input and output activities.

THEORY AND RECURRENT NEURAL NETWORKS

A recurrent neural network[3,6] may be assigned to the so-called parent neural network that contains oriented cycles (feed-back connections). We shall postulate that the parent network is determined by an oriented graph[8] composed of neurons (vertices) and connections (oriented edges). The symbol Γ_i (Γ_i^{-1}) denotes a subset composed of all successors (predecessors) of the neuron v_i in the parent network. The set of neurons is divided into three disjointed subsets of the so-called input neurons, hidden neurons, and output neurons. All the hidden and output neurons induce in the parent network a complete subgraph (i.e., each neuron is adjacent with other

0095-2338/92/1632-0742$03.00/0

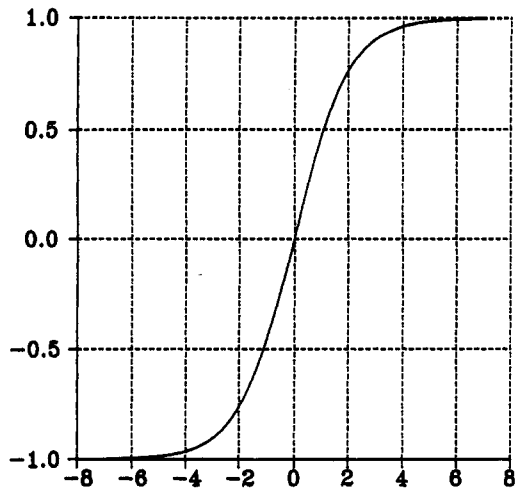

Figure 2. Plot of the transfer function $f(\xi)$ determined by eq 2a for $A = -1$ and $B = 1$.

neurons by two oppositely oriented connections and with itself by an oriented loop) whereas the input neurons are incident only with connections that are outgoing from them (see Figure 1). The activities of input neurons are kept fixed while the activities of hidden and output neurons are determined by

$$x_i = f(\xi_i) \tag{1a}$$

$$\xi_i = \vartheta_i + \sum_j \omega_{ij} x_j \tag{1b}$$

where ϑ_i and ω_{ij} are the threshold coefficient (assigned to the neuron v_i) and weight coefficient (assigned to the connection outgoing from v_j and incoming to v_i), respectively. The transfer function $f(\xi)$ is determined by

$$f(\xi) = \frac{B + A \exp(-\xi)}{1 + \exp(-\xi)} \tag{2a}$$

It maps real axis $(-\infty,\infty)$ onto an open interval (A,B), where $A \leq 0 < B$, and its first derivative is

$$f'(\xi) = \frac{[f(\xi) - A][B - f(\xi)]}{B - A} \tag{2b}$$

The transfer function is monotonously increasing and fulfilling $f(\xi) \to A$ as $\xi \to -\infty$ and $f(\xi) \to B$ as $\xi \to \infty$ (see Figure 2).

The activities of the parent neural network are determined by a string of coupled nonlinear equations (1a and 1b) that may be solved (for fixed activities of input neurons and for given threshold and weight coefficients) only iteratively (providing that they have a common solution—fixed point). This is the main obstacle for broader applications of neural networks with feed-back connections, their activities are not determined in such a simple and finite recurrent scheme as the activities of the feed-forward neural networks.[3]

An iterative approach of solution (eq 1) may be outlined as follows: Initially (zero step), the input activities x_i of input neurons are kept fixed and other activities of hidden and output neurons are put equal to zero. The kth step ($k \geq 1$) consists in the evaluation of activities of hidden and output neurons by

$$x_i^{(k)} = \begin{cases} f(\xi_i^{(k-1)}), & (\exists\ j \in \Gamma_i^{-1}: x_j^{(k-1)} \neq 0) \\ 0, & (\forall\ j \in \Gamma_i^{-1}: x_j^{(k-1)} = 0) \end{cases} \tag{3}$$

The resulting activity is nonzero (upper raw) if there exists at least one neuron $j \in \Gamma_i^{-1}$ with nonzero activity evaluated in the previous $(k-1)$th step. In the opposite case (lower row), if all neurons $j \in \Gamma_i^{-1}$ have zero activities in the previous $(k-1)$th step, then the resulting activity is also equal to zero. The above iterative process is recurrently repeated until all hidden and output activities remain constant within a prescribed precision.

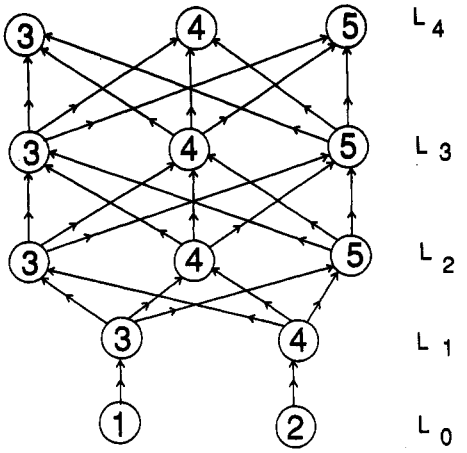

Figure 3. Recurrent neural network assigned to the parent network displayed in Figure 1. This network is composed of five layers L_0, L_1, ..., L_4. The bottom layer L_0 is composed entirely of input neurons, the next layer L_1 is composed entirely of hidden neurons, and the forthcoming higher layers are composed of both hidden and output neurons. The value of k (right column) determines an iteration that produces the given layer. To keep the drawing of the network simple, the connections between the bottom layer ($k = 0$) and the higher layers ($k = 2, 3, 4$) are omitted.

What is very interesting is that the above iterative scheme may be considered as a straightforward generalization of the evaluation of activities in a feed-forward neural network.[3] The iterative scheme (eq 3) may be simply schematically visualized by expanding the recurrent neural network in a feed-forward counterpart[3] (see Figure 3) composed of L_0, L_1, L_2, ... layers. The bottom layer L_0 is composed entirely of input neurons, whereas a layer L_k (for $k \geq 1$) is composed of those neurons that are the Γ image of neurons from the previous layer L_{k-1}, and for which the formula gives nonzero activites

$$L_k = \{\Gamma(L_{k-1});\quad x_i^{(k)} \neq 0\} \tag{4}$$

The neurons from L_k are linked by up-oriented connections with neurons from the lower layers with nonzero activities. Moreover, all neurons indexed by the same index i but from different layers are evaluated by the same threshold coefficient ϑ_i. Similarly, all connections [i,j] between neurons v_i and v_j from the juxtaposed layers are always evaluated by the same weight coefficient ω_{ji}.

We see that it is possible to assign to each parent network the recurrent neural network (a kind of feed-forward neural network). It may have an infinite number of layers, but practically this number is bounded from above by a positive finite number. Going successively through all its layers we are doing nothing but a recurrent repetition of eq 3.

An approach to the adaptation process of recurrent neural networks is based on its interpretation as a simple feed-forward recurrent network. Since the activities of hidden and output neurons are now determined by strings of coupled nonlinear equations, their iterative solution (eq 3) consists in a successive evaluation of activities in the iteratively constructed recurrent network. After a finite number of steps, the activities of output neurons will be nonzero, and these activities are successively turning (assuming that this process converges) to their exact (self-consistent) values. This is true not only for output neurons

from the last layer but also for output neurons from lower layers, which activity values may differ from each other in different layers. We stress that weight and threshold coefficients of corresponding connections and neurons in different layers are equal.

The presence of output neurons in different layers opens a very interesting new possibility[3] of how to use the recurrent neural networks for generation of different output activities for the same vector of input activities. For instance, an object may be classified by a set of different classifiers that correspond to successively appearing properties of the given object (e.g., developed in the successive time steps). Then these different classifiers will be joined with different layers of the neural network. After adaptation of neural network, it can give for one input a set of successive outputs taken from output neurons from different layers. During the adaptation process for the method, the agreement of outputs with the required values must be observed at all those layers where we are taking output classifiers.

The recurrent neural networks involve a special kind of adaptation process. As follows from the above discussion, the iterative solution (eq 3) gives after a finite number (say $r \geq 1$) of steps nonzero output activities. Hence, starting from the kth step (where $k \geq r$) the resulting output activities may be compared with the required ones, and applying standard formulas[3] it is possible to calculate the gradient of objective function determined as a sum of quadrates of differences between calculated and required output activities. This calculation is realized with respect to a current feed-forward network generated from the given parent network at the kth step. The gradient calculated at each step is added to a total gradient, which was initialized by setting all its entries to zero. Finally, after all the required output states $\hat{x}_1$, $\hat{x}_2$, ..., $\hat{x}_q$ have been used [i.e., after q steps, where the first (last) vector $\hat{x}_1$ ($\hat{x}_q$) has been compared with output activities produced by the assigned recurrent network composed of r ($r + q$) layers], the weight and threshold coefficients are updated by an analog of the steepest-descent method.[3] The major problem with this simple adaptation process is the memory consumption. Not only does the method have to store the total gradient while the different required output activities are used, but each hidden and output neuron must store the sequence of its activities through which it was passing during the processing.

SIMPLE ILLUSTRATIVE EXAMPLE—SUMMATION OF TWO 2-DIGIT BINARY NUMBERS

The theory of recurrent neural networks outlined in the previous section will be illustrated by a simple example of summation of two 2-digit binary numbers

$$\begin{array}{ccc} & \alpha_1 & \alpha_2 \\ & \alpha_3 & \alpha_4 \\ \hline \alpha_5 & \alpha_6 & \alpha_7 \end{array} \qquad (5)$$

where α's are 0–1 digits. This scheme may be realized by 16-different ways (see Table I). The parent neural network is composed of four input neurons (with activities x_1–x_4), five hidden neurons (x_5–x_9), and one output neuron (x_{10}); the hidden and output neurons induce in this network a complete oriented subgraph (see Figure 4). The corresponding recurrent neural network is horizontally structured into four layers, L_0–L_3, where the bottom layer L_0 is composed of input neurons while the forthcoming layers L_1–L_3 are the same and are composed of all hidden and output neurons. The neurons

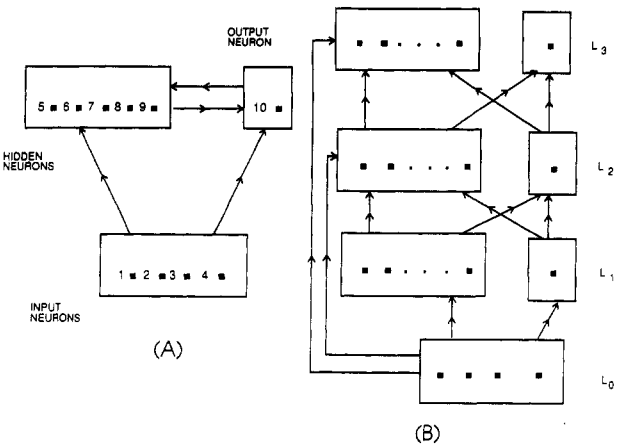

Figure 4. (A) Illustrates the parent neural network used for the classification of sum of two 2-digit binary numbers. The separate blocks correspond to input neurons (1–4), hidden neurons (5–9), and output neuron (10). All hidden and output neurons are adjacent by pairs of opposite-oriented connections, and each of these neurons is adjacent to an oriented loop. (B) Illustrates the corresponding recurrent neural network composed of four layers. The activities of output neuron (10) from layers L_1, L_2, and L_3 are required outputs of the network.

Table I. Summations of Two 2-Digit Binary Numbers

no.	α_1	α_2	α_3	α_4	α_5	α_6	α_7	meaning
1	0	0	0	0	0	0	0	0 + 0 = 0
2	0	0	0	1	0	0	1	0 + 1 = 1
3	0	0	1	0	0	1	0	0 + 2 = 2
4	0	0	1	1	0	1	1	0 + 3 = 3
5	0	1	0	0	0	0	1	1 + 0 = 1
6	0	1	0	1	0	1	0	1 + 1 = 2
7	0	1	1	0	0	1	1	1 + 2 = 3
8	0	1	1	1	1	0	0	1 + 3 = 4
9	1	0	0	0	0	1	0	2 + 0 = 2
10	1	0	0	1	0	1	1	2 + 1 = 3
11	1	0	1	0	1	0	0	2 + 2 = 4
12	1	0	1	1	1	0	1	2 + 3 = 5
13	1	1	0	0	0	1	1	3 + 0 = 3
14	1	1	0	1	1	0	0	3 + 1 = 4
15	1	1	1	0	1	0	1	3 + 2 = 5
16	1	1	1	1	1	1	1	3 + 3 = 6

from two juxtaposed layers are fully adjacent by up-oriented connections, and furthermore, the neurons from L_0 and L_i (for $1 \leq i \leq 3$) are also adjacent by up-oriented connections (see Figure 4). In such a way we have arrived at the feed-forward neural network with the constraint that corresponding connections with the same terminal neurons in different layers are evaluated by the same weight coefficients. For fixed threshold and weight coefficients the iterative solution (eq 3) is performed so that, successively going from the bottom layer L_0 to the top layer L_3, we calculate initially activities of neurons from the layer L_1. This layer is fed by input activities from the layer L_0, in particular we put $x_i^{(0)} = \alpha_i$, for $i = 1, 2, 3, 4$. Then knowing the activities of neurons from L_1 (and also the input activities from L_0), we calculate the activities of neurons from the layer L_2, and so on. Activities of the output neuron from the layers L_1 to L_3, denoted by $x_{10}^{(1)}, x_{10}^{(2)}, x_{10}^{(3)}$, are considered as outputs of neural network that are compared with required output equal to 0–1 digits α_7, α_6, α_5 from the scheme (eq 5), respectively. The partial derivatives of minimized objective function are calculated by standard back-propagation method[3] separately for each level L_i, where $1 \leq i \leq 3$, and the derivatives are summed up through all these layers. The updating of threshold and weight coefficients, done separately for each input pattern from Table I, is performed by the usual steepest-descent approach accelerated by the so-called momentum method.[3] The recurrent neural

Table II. Illustrative Output from Recurrent Neural Network for Pattern 1 + 1 = 2

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$x_{10}^{(k)}$
1	0	1	0	1	
2	0	1	0	1	0.03 (α_7)
3	0	1	0	1	0.98 (α_6)
4	0	1	0	1	0.01 (α_5)

network has been adapted after a few hundred cycles with the steepest-descent parameter $\lambda = 0.1$ and the momentum parameter $\alpha = 0.7$. An illustrative example of the adapted network is given in Table II. We see that the present type of recurrent neural network is able to generate for each input pattern from Table I a sequence of output activities that are closely related to the required 0–1 digits. It may be understood as a simple generalization of standard feed-forward neural networks that are adapted by back-propagation approach, which is potentially well applicable in the cases when the input patterns are classified by a sequence of "properties".

APPLICATION—^{13}C CHEMICAL SHIFTS OF MONOSUBSTITUTED BENZENES

The present theory of recurrent neural networks outlined and illustrated in the previous sections will be applied as a classifier of monosubstituted benzenes with respect to their ^{13}C NMR chemical shifts (four different positions on the benzene skeleton called ipso, ortho, meta, and para position). The used parent neural networks are composed of 11 input neurons (their activities are equal to descriptors, which determine in a proper way the structure of a given substituent –X), 0–6 hidden neurons, and 4 output neurons (with activities equal to chemical shifts in ipso, ortho, meta, and para position, respectively). The chemical shifts of monosubstituted benzenes[9] belong to a relatively large interval of real numbers. Therefore, they should be compressed (see refs 7 and 11) to values from a smaller open interval (A,B) by the following nonlinear transformation (cf. eq 2)

$$t(x) = \frac{y_{max} + y_{min}e^{-\alpha(x+\beta)}}{1 + e^{-\alpha(x+\beta)}} \tag{6}$$

The coefficients α and β are adjusted such that $x_{min} \le x \le x_{max}$ is mapped onto $(y_{min} + \epsilon) \le y \le (y_{max} - \epsilon)$. A positive number ϵ is determined by $\epsilon = (y_{max} - y_{min})\kappa$, for a small positive κ ranged by $0 < \kappa < 1/2$. An inverse of eq 6 is

$$x = t^{-1}(y) = \frac{1}{\alpha} \ln \frac{y - y_{min}}{y_{max} - y} - \beta \tag{7}$$

The coefficients α and β are determined by

$$\beta = -1/2(x_{min} + x_{max}) \tag{8a}$$

$$\alpha = \frac{1}{x_{min} + \beta} \ln \frac{\epsilon}{y_{max} - y_{min} - \epsilon} \tag{8b}$$

For our purposes, the above parameters have been selected as follows: $x_{min} = -18$, $x_{max} = 36$, $y_{min} = 0$, $y_{max} = 2$, and $\kappa = 0.025$ (i.e., $\epsilon = 0.05$). This means that all chemical shifts taken from the open interval (–18,36) are compressed by the transformation (eq 6) to the open interval (0.05,1.95). The compressed values of chemical shifts are appropriate for recurrent neural network applications with transfer function (eq 2a) determined by $A = 0$ and $B = 2$.

The calculated output activities of recurrent neural networks are compared with required output activities (transformed chemical shifts) after three or four iterations ($k = 3$ or $k = 4$). We use the same training (44 objects) and testing sets (20 objects) of monosubstituted benzenes as in our recent paper[7] solving the same problem by neural networks with feed-back connections. The substituents –X are determined for purposes of the present recurrent network by 11 descriptors (nonnegative integer input activities) that determine in an additive way the topology and physicochemical nature of functional groups.[10,11] These descriptors are divided into three categories: the first-, second-, and third-level descriptors, in a dependence on the distance of atoms of the functional group from the ipso carbon atom on the benzene ring. That is, the descriptors determine the α, β, and γ effects with respect to the carbon atoms of the benzene ring, the higher effects (in particular δ and ϵ effects) are neglected. The descriptors are determined in Table III, where only non-hydrogen atoms are explicitly considered, the hydrogen atoms linked to the non-hydrogen atoms of functional groups are considered as their property. Simple illustrative examples of the descriptors are given in Table IV.

Table III. Descriptors of Functional Groups

d_i	meaning of descriptor
	First-Level Descriptors
d_1	number of lone electron pairs on the first level atoms
d_2	sum of the main quantum numbers (each decreased by one) of the first level atoms
d_3	number of hydrogen atoms attached to the first-level atoms
	Second-Level Descriptors
d_4	number of lone electron pairs on the second level atoms
d_5	sum of the main quantum numbers (each decreased by one) of the second-level atoms
d_6	number of hydrogen atoms attached to the second-level atoms
d_7	number of π bonds that connect the first- and second-level atoms
	Third-Level Descriptors
d_8	number of lone electron pairs on the third-level atoms
d_9	sum of the main quantum numbers (each decreased by one) of the third level atoms
d_{10}	number of hydrogen atoms attached to the third-level atoms
d_{11}	number of π bonds that connect the second- and third-level atoms

The recurrent neural network applied for prediction and classification of ^{13}C NMR chemical shifts in a series of monosubstituted benzenes has been performed for different numbers of iterative steps k (3–5) and hidden neurons N_H (0–6). The most minimal value of the objective function ($E = 0.08$) and the best predicted values of chemical shifts for monosubstituted benzenes from the training set are obtained for $k = 3$ and $N_H = 6$. That is, the parent neural network contains six hidden neurons, and the recurrent neural network assigned to this parent network is constructed as a four-layer feed-forward network, where the activities of output neurons from the fourth layer ($k = 3$) are compared with required activities (compressed chemical shifts). For this recurrent neural network, the experimental[9] and calculated chemical shifts are listed in Table V. We see that the functional group $-Si(CH_3)_3$ shows the greatest discrepancy in ipso position from testing-set objects. The singularity of this functional group has been also observed in our recent work[11] devoted to an application of feed-forward neural networks for prediction and classification of reaction yields of nitration of monosubstituted benzenes. The same observation has been achieved also by Elrod et al.[12] (see also ref 13), who studied the same classification problem but with descriptors determined as elements of the BE-matrix (an analog of adjacency matrix). It seems that the silicon atom probably needs further, more specific descriptors.

Table IV. Illustrative Examples of Descriptors Assigned to Simple Functional Groups

–X	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}
$-CH_3$	0	1	3	0	0	0	0	0	0	0	0
$-NO_2$	0	1	0	4	2	0	2	0	0	0	0
$-N(CH_3)NO$	1	1	0	1	2	3	0	2	1	0	1

Table V. Results of ^{13}C NMR Chemical Shifts of Monosubstituted Benzenes from Testing Set

no.	–X		ipso	ortho	meta	para
1	$-CH_2COCH_3$	exp	6.0	1.0	0.2	–1.6
		cal	2.4	2.9	–0.3	–0.8
2	$-CH_2N(CH_3)_2$	exp	11.1	0.8	–0.2	–1.5
		cal	8.3	0.9	–1.0	–3.2
3	$-CH_2SOCH_3$	exp	0.8	1.5	0.4	–0.2
		cal	–0.5	1.7	0.9	1.9
4	$-CH_2SO_2CH_3$	exp	–0.1	2.1	0.6	0.6
		cal	2.9	3.4	3.4	6.0
5	$-CCl_3$	exp	16.3	–1.7	–0.1	1.8
		cal	10.5	–9.0	0.9	2.8
6	$-CH_2Br$	exp	9.5	0.7	0.3	0.2
		cal	10.5	–0.8	0.7	0.1
7	$-CH_2I$	exp	10.5	0.0	0.0	–0.9
		cal	12.6	–1.7	1.3	0.5
8	$-C(CH_3)=CH_2$	exp	12.6	–3.1	–0.4	–1.2
		cal	15.4	–1.7	–0.3	1.9
9	$-COCH_3$	exp	8.9	0.1	–0.1	4.4
		cal	8.0	–0.1	–0.1	4.3
10	$-COCH_2CH_3$	exp	8.8	0.2	–0.5	4.3
		cal	7.2	–6.4	–0.3	–1.2
11	$-CON(CH_3)_2$	exp	8.0	–1.5	–0.2	1.0
		cal	1.9	–9.5	–0.3	–6.8
12	$-COOCH_3$	exp	2.0	1.2	–0.1	4.3
		cal	7.8	7.6	–0.3	1.1
13	$-Si(CH_3)_3$	exp	11.7	5.9	–0.7	0.3
		cal	–0.5	9.6	–2.2	2.4
14	$-NHCH_2CH_3$	exp	20.0	–15.7	0.7	–11.4
		cal	16.5	–8.5	0.3	–8.5
15	$-N(CH_2CH_3)_2$	exp	19.3	–16.5	0.6	–13.0
		cal	9.4	–12.7	0.1	–13.5
16	–NCO	exp	5.1	–3.7	1.1	–2.8
		cal	8.9	–6.0	–0.3	2.9
17	$-OCH_3$	exp	31.4	–14.4	1.0	–7.7
		cal	30.7	–18.0	0.9	–8.1
18	–OCN	exp	25.0	–12.7	2.6	–1.0
		cal	18.5	–10.6	1.9	–1.7
19	$-SCH_3$	exp	10.0	–1.9	0.2	–3.6
		cal	12.0	–3.2	0.9	–0.3
20	–Cl	exp	6.3	0.4	1.4	–1.9
		cal	9.8	5.2	0.7	1.9

SUMMARY

The recurrent neural network approach offers a simple possibility on how to overcome the convergence difficulties of the neural networks with feed-back connections. These neural networks may be adapted by a simple analog of the standard back-propagation strategy widely used in the feed-forward layered neural networks. The functional groups are described by 11 descriptors that reflect in an additive manner the topology and physicochemical nature of their atoms and bonds. The obtained results for ^{13}C NMR chemical shifts of monosubstituted benzenes indicate that the recurrent neural networks together with the used form of descriptors offer a promising technique for classification and prediction of molecular properties, which is based namely on the topology of the corresponding structural formulas.

REFERENCES AND NOTES

(1) Anderson, J. A.; Rosenfeld, Z., Eds.; *Neurocomputing: Foundation of the Research*; MIT Press: Cambridge, MA, 1989.
(2) Simpson, P. K. *Artificial Neural Systems*; Pergamon Press: New York, 1990.
(3) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing*; Rumelhart, D. E., McClelland, J. L., Eds.; MIT Press: Cambridge, MA; Vol. I, pp 318–362.
(4) Župan, J.; Gasteiger, J. Neural networks: A new method for solving chemical problems or just a passing phase. *Anal. Chim. Acta* **1991**, *248*, 1–30.
(5) Maggiora, G. M.; Elrod, D. W.; Trenary, R. G. Computational Neural Nets as Model-Free Mapping Devices. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, this issue.
(6) Minsky, M.; Papert, S. *Perceptrons. An Introduction to Computational Geometry*; MIT Press: Cambridge, MA, 1969.
(7) Kvasnička, V.; Sklenák, S.; Pospíchal, J. Application of neural networks with feed-back connections in chemistry. Prediction of ^{13}C NMR

chemical shifts in a series of monosubstituted benzenes. *J. Mol. Struct. (THEOCHEM)* **1992**, in press.
(8) Harary, F. *Graph Theory*; Addison-Wesley: Reading, MA, 1969.
(9) Kalinowski, H. O.; Berger, S.; Braun, S. *^{13}C NMR Spektroskopie*; Thieme: Stuttgart, 1984.
(10) Zou, Y.; Johnson, M.; Tsai, C. C. Modeling Aromatic Nitration Reactions Using Graph Theoretic Transforms. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 442–450.
(11) Kvasnička, V.; Pospíchal, J. Application of neural networks in chemistry. Prediction of product distribution of nitration in a series of monosubstituted benzenes. *J. Mol. Struct. (THEOCHEM)* **1990**, *235*, 227–242.
(12) Elrod, D. W.; Maggiora, G. M.; Trenary, R. G. Applications of Neural Networks in Chemistry. I. Prediction of Electrophilic Aromatic Substitution Reactions. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 477–484.
(13) Elrod, D. W.; Maggiora, G. M.; Trenary, R. G. Applications of Neural Networks in Chemistry. 2. A General Connectivity Representation for the Prediction of Regiochemistry. *Tetrahedron Comput. Methodol.* **1990**, *3*, 163–174.