$\mathcal{ACH}$ – MODELS IN CHEMISTRY *132 (6), pp 887-901 (1995)*

Predicting the dynamics of an oligo-oscillatory reaction by an artificial neural network

ISTVÁN Z. KISS and VILMOS GÁSPÁR*

Department of Physical Chemistry, Kossuth L. University, P.O. Box 7, 4010 Debrecen, Hungary

Received April 27, 1995

Artificial neural networks (ANNs) are model-free computational tools that can "learn" the linear or nonlinear rules embedded in a dataset. We report the results of an attempt to utilize ANNs in the field of reaction kinetics. A feedforward network is trained to predict the main dynamical features of *oligo-oscillations* in the acidic bromate–ascorbic acid–malonic acid reacting mixture, in which the concentration of bromide ion (an intermediate) shows three extrema as a function of time. Inputs to the network are the initial concentrations of reactants, while outputs are the predicted values of bromide-ion concentration and reaction time at the extrema. The network is first tested on a numerically generated dataset and then applied to experiments. The results provide evidence that ANNs can be efficiently employed for the prediction of the dynamics of complex chemical systems, especially, when the mechanism of a reaction is not fully understood.

Introduction

The classical method of reaction kinetics for the prediction of the course of a simple reaction at various conditions is based on first developing an as detailed model for the system as possible. Once a plausible mechanism has been established, that is, the component processes (or elementary steps) and the values of rate constants are known, the numerical solution of the corresponding system of ordinary differential equations (ODEs) generally gives a good estimate on the system's behavior. However, it is extremely difficult to work out the right mechanism for complex chemical systems showing highly nonlinear (exotic) phenomena such as explosion, periodic or aperiodic oscillations, chaos, etc. A good example is the celebrated Belousov–Zhabotinsky oscillating reaction [1], the mechanism of which is still searched for [2] after more than thirty years of investigations by a number of laboratories. In this paper, we suggest an alternative method for the prediction of the dynamics of complex chemical reactions when there is only little or no hope at all to develop an adequate mechanism for a given system. As the kinetic experiments always result in a large number of data

* To whom correspondence should be addressed

1217-8969/95/$ 5.00

(for example, concentrations as a function of time), the new skill is based on handling of such data by artificial neural networks.

ANNs are model-free mapping devices that have provided solution to a broad range of difficult problems in chemistry during the last few years [3–5]. The success of neural computing is based on the following: presented with training data, a neural network can "learn" the linear or non-linear rules embedded in the data [6]. We contemplated that, perhaps, this remarkable feature of neural networks can be utilized in the field of reaction kinetics, too. To test the idea, we have chosen a simple problem, and tried to train an ANN to predict only the main dynamical features of oligo-oscillations in the acidic bromate–ascorbic acid–malonic acid (BAM) reacting mixture.

The paper is structured as follows. Networking the flow of data is not a familiar concept to chemists, therefore, an introduction to ANNs is given in the section entitled *Artifical neural networks*. We will focus on the error-backpropagation algorithm that has been employed in this work. In the next section, the dynamics of oligo-oscillations in the BAM-system is characterized in order to construct an ANN with the right architecture for our purpose. The network is first tested on a numerically generated dataset and then applied to experiments in the section entitled *An artifical neural network for the BAM system*. The results are summarized in the concluding section. Also, the advantages and disadvantages of applying ANNs in the field of reaction kinetics are briefly discussed.

Artificial neural networks

Artificial neural networks can be considered as computational tools that *mimic* certain elements of the learning behavior of a living neural system. The adjective "artificial" is to clarify that they are not biological systems. The neural network domain of computer software is best identified by a "black box" that simply transforms an m-variable input into an n-variable output. The variables can be binary numbers (i.e., 0 and 1), bipolar numbers (for example, −1 and +1), or real numbers (generally scaled to the range from 0 to +1, or from −1 to +1).

The artificial neuron itself (Fig. 1a) has been designed to model the activity of a neural cell in the brain [6]. It collects inputs from neighboring units and transforms the net input into an output. According to the theory of ANNs, the total input I_j to a particular unit j is the weighted sum of the output of all neurons ($i = 1, 2, \ldots, n$) connected to it:

$$I_j = \theta_j + \Sigma_i w_{ji} o_i , \qquad (1)$$

where o_i is the output of the ith unit, and w_{ji} is a weight characterizing the "strength" of connection between units j and i. The weights may have positive or negative values representing, respectively, activatory or inhibitory connections similar to those found in real neural systems. The constant θ_j is related to a threshold that has to be exceeded by the net input; otherwise, no output is produced. To simplify the notation, the role of θ_j can be taken up by an additional weight that will represent the strength of connection between the chosen unit and an imaginary neuron called *bias*. A bias unit receives no input and has a constant output of unity.

The total input I_j is transformed into an output o_j by being passed through a transfer (or activation) function. The "on-off" behavior of real neural activity could be modeled simply by a step function, nevertheless, a widely used formula to evaluate the output of an artificial neuron is the following sigmoid function:

$$o_j = \{1 + \exp(-I_j)\}^{-1} . \tag{2}$$

The motivation for choosing this formula comes from the smoothing the "on-off" step function and, by the same token, forcing the output to be within the range from 0 to +1 regardless of the magnitude and sign of the total input.

Artificial neurons can be connected to a network in an ample number of ways. Here we give a brief introduction to the so called *feedforward* networks only, through which the information flows in just one direction (no recurrent signals). For example, Fig. 1b shows a simple 3-4-2 feedforward network with three, four, and two units in the input, hidden, and output layers, respectively. (In general, the number of hidden layers is not limited.) Notice that a strip of bias units has been added to the network. These units receive no input but send out constant (unit) signals to every neuron of the next layer. The symbol of input and bias units, in which the transfer function is not activated, is a square. The hidden and output units are represented by a circle divided into two parts. The net input is collected in the top half and then the sigmoid function is applied in the bottom part to generate the output.

An ANN with randomly set weights is "dull" but can be trained by successive repetitions of the same problem. A network "learns" by iteratively correcting the weights (the only adjustable parameters) so as to produce the previously specified output values (target sets) for as many input sets as possible. One of the most successful of such supervised learning methods is the so called *backpropagation algorithm* [6], in which the correction of weights always starts at the output layer, then it continues backwards, layer by layer toward the input. After the weights have been adjusted, the error is recalculated by doing a forward pass through the network. Then, the backpropagation algorithm is applied again, and the weights are newly updated,

etc. This procedure is repeated until the error of the network is less than an acceptable value or the number of iterations exceeds a preset limit.

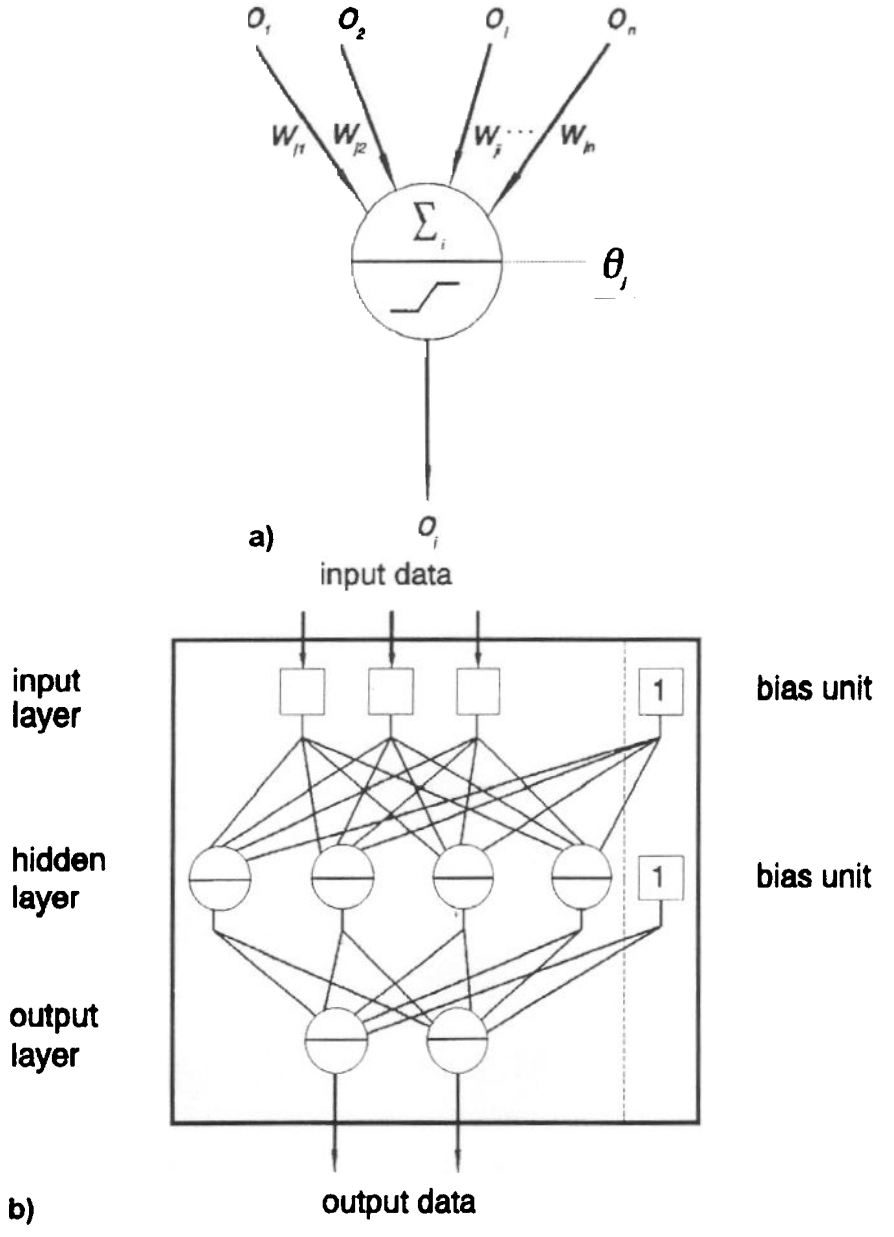

Fig. 1. a) Graphical representation of the activity of an artificial neuron; b) The architecture of a simple 3-4-2 feedforward network

A commonly used equation for calculating the error of a network is the following:

$$E = 1/2\ \Sigma_j (o_j^{out} - t_j)^2 \qquad (3)$$

where o_j^{out} and t_j are, respectively, the actual and desired (target) output values of unit j in the output layer. The weights are corrected according to the so-called *delta-rule* of learning:

$$\Delta w_{ji} = \eta \, \delta_j \, o_i \, ,$$

where $\eta > 0$ is the *learning rate*, δ_j is a correction term, and o_i is the output of unit i in the previous layer. The value of δ_j should be proportional to the output-error. In the back-propagation algorithm the correction-term is obtained by applying the so-called *gradient descent* method, which leads to the following expression for the delta-term of an output unit:

$$\delta_j = - (\partial E / \partial o_j) (\partial o_j / \partial I_j) = (t_j - o_j^{out}) \, o_j \, (1 - o_j) \, .$$

Since there exists no specified target set to calculate the error for a hidden layer, the corresponding δ_j values are figured out by backpropagating the error through the network (hence, the name of the algorithm). The following recursive formula is applied to calculate the correction term for a hidden unit:

$$\delta_j = o_j \, (1 - o_j) \, \Sigma_k \delta_k w_{kj} \, , \tag{6}$$

where subscript k refers to units in the lower layer to which a hidden unit j connects, and w_{kj} is the weight of those connections. Notice that by inserting Eq. (6) into Eq. (4), there are *three subsequent layers* involved in the calculations for the weight-corrections in case of hidden units.

It has been found that the performance and also the stability of a training process is greatly enhanced if a so-called *momentum term* is added to the learning rule:

$$\Delta w_{ji} = \eta \, \delta_j \, o_i + \mu \, \Delta w_{ji}^{prev} \, , \tag{7}$$

where $0 < \mu < 1$ is a constant called *momentum*, and Δw_{ji}^{prev} is the adjustment to the same weight in the previous iteration cycle.

The great power of neural computing is that a trained network can be used to the prediction of outputs at a new set of inputs by simply doing a forward pass through the network with the fixed weights. An important limitation is, however, that the new input values should fall into the ranges of those applied for training.

A computer program for a feedforward backpropagating artificial neural network with one hidden layer was written in FORTRAN and run on a VAX-6000. The program allows an arbitrary number of units in each layer to be defined and includes both training and predictive faculties. This model-free "number-crunching-device" can solve, in essence, almost any chemical (or nonchemical) problem, if the input and output data characterizing the system are appropriately chosen.

Because of the deterministic nature of chemical reactions, the future behavior of a chemical system is set by the initial concentrations of its components. (We consider only homogenous chemical reactions run in well-stirred batch reactors and kept at constant temperature.) For this reason, one can always choose the initial concentrations as the input data to an ANN. However, the choice for output data always depends on the investigated system itself, and is also limited by the analytical methods applied to follow the course of a reaction. After characterizing the dynamics of oligo-oscillations in the BAM-system, we shall make a decision on what type of output data should be used for our purpose.

The bromate–ascorbic acid–malonic

The bromate–ascorbic acid–malonic acid–perchloric
first example of at least two dozen other oligo-oscillatory reactions known today [9]. The adjective *oligo* has been introduced by Beck and Rábai [10] to indicate that in these extended "clock-reactions" the concentrations of some intermediates may exhibit only few extrema as a function of time. The oligo-oscillatory behavior of the BAM-system has been calculated in good agreement with the experiments [8] by assuming the following four component processes :

$$BrO_3^- + 3\ H_2A \rightarrow 3\ A + Br^- + 3\ H_2O\ , \qquad (a)$$

$$BrO_3^- + 5\ Br^- + 6\ H^+ \rightarrow 3\ Br_2 + 3\ H_2O\ , \qquad (b)$$

$$H_2A + Br_2 \rightarrow A + 2\ Br^- + 2\ H^+\ , \qquad (c)$$

$$H_2M + Br_2 \rightarrow HBrM + Br^- + H^+ \qquad (d)$$

and by applying the corresponding empirical rate laws (a')–(d'):

$$-\,d[BrO_3^-]/dt = k_a[BrO_3^-]\,[H_2A]\ , \qquad (a')$$

$$-\,d[BrO_3^-]/dt = k_b[BrO_3^-]\,[Br^-][H^+]^2\ , \qquad (b')$$

$$-\,d[Br_2]/dt = k_c[Br_2]\,[H_2A]\ , \qquad (c')$$

$$-\,d[Br_2]/dt = k_e k_d[Br_2]\,[H_2M]/(k_{-e} + k_d[Br_2])\ , \qquad (d')$$

where $k_a = 8.3\times10^{-3}$ $M^{-1}s^{-1}$, $k_b = 3.6$ $M^{-3}s^{-1}$, $k_c = 1.7\times10^{3}$ $M^{-1}s^{-1}$ $k_d = 1.7\times10^{7}$ $M^{-1}s^{-1}$, $k_e = 3.3\times10^{-3}s^{-1}$, and $k_{-e} = 1.7\times10^{3}$ s^{-1} ($M \equiv mol\,dm^{-3}$). The last two constants appear in rate equation (d') for the following reason: the rate-

determining step in process (d) is the slow but reversible enolization ($K_{enol} = k_e/k_{-e}$) of the more stable keto-form of malonic acid initially added to the reacting mixture.

Based on Eqs (a)–(d) and (a')–(d'), changes in the concentrations of bromate ion, bromine, bromide ion, malonic and ascorbic acids (the five dynamical variables of the BAM-system) can be calculated by numerically integrating [11] the corresponding system of ODEs. The concentration of hydrogen ion is assumed to be constant. The solid curve in Fig. 2a shows the calculated change in the concentration of bromide ion (an intermediate) as a function of time at the initial concentrations given in Table I, No. 1. The reaction is oligo-oscillatory: there are three extrema (two maxima and one minimum) on the concentration (M) versus time (s) curve. Oligo-oscillatory curves have been found in a well-defined range of initial concentrations, however, the shape of these curves greatly depends on the initial concentrations of the reactants [7, 8]. Therefore, the following three pairs of data: $(t, [Br^-])_{max,1}$, $(t, [Br^-])_{min}$, and $(t, [Br^-])_{max,2}$ can be looked at as the "fingerprint" of the oligo-oscillatory dynamics, and their values will be used as output data in our ANN algorithm.

An artificial neural network for the BAM system

A 4–4–6 feedforward neural network has been constructed for the prediction of the dynamical fingerprint of the BAM-system. The four input data to the network are the initial concentrations of BrO_3^-, H_2A, H_2M, and H^+ (M), while the six output data are those defined earlier. Since the range of the sigmoid function is between zero and one, both the input and output data are normalized according to the following formula:

$$x_i = \{0.9(X_i - X_{i,min})/(X_{i,max} - X_{i,min})\} + 0.001 \,, \qquad (8)$$

where x_i is the scaled (dimensionless) value of an original data X_i. The upper and lower limits for scaling are set, respectively, by the values of $X_{i,max}$ and $X_{i,min}$. (It is not essential to scale the data or to use this particular formula, but it worked well in our case. For other formulas see, for example, [12–14].)

Selecting the number of hidden units is always a crucial task that has great impact on the effectiveness of the algorithm [15–18]. With too few hidden units the network may not be capable of extracting all the rules embedded in the training data, while with too many hidden units the network will simply "memorize" the data (over-training). A good rule of thumb is that the ratio of the number of datapoints *vs.* the number of bonds in the network must be > 1. Since there are $13 \times 4 = 52$ input datapoints in both the numerical and experimental tests (see next section), the maximum number of hidden units is limited to four (resulting in 50 bonds in the network).

The two parameters affecting the rate of convergence to a local minimum, the learning rate and momentum in Eq. (7) were fixed during training the network ($\eta=0.25$ and $\mu=0.9$) [3]. (We did not investigate the dependence of model performance by changing the values of these parameters.) In an attempt to optimize the network, we changed the number of hidden units in a bottom-up approach starting with two hidden units and then adding more during training. The cumulative error of the network (the sum of output-errors evaluated by Eq. (3) for the whole training set) has been calculated and found to be minimal by using four hidden units. Training of the 4–4–6 network was stopped at 200 thousand iterations after which the generalization performance of the network began to degrade. (The value of cumulative errors for the numerically generated and the experimentally obtained training sets were 1.9×10^{-3} and 4.7×10^{-2}, respectively.)

Testing the ANN on numerically generated data

Results of testing the ANN on a numerically generated dataset are shown in Table I. All values are given in their original dimensions. The applied scaling limits are listed in the bottom rows. The bromide-ion concentrations at the extrema: $[Br^-]_{max,1}$, $[Br^-]_{min}$, and $[Br^-]_{max,2}$ have been multiplied by 10^2, 10^3, and 10^3, respectively. The first rows of output data were generated by solving the ODEs of the BAM-system at the corresponding values of initial concentrations (input rows). The second rows of outputs (in italics) were calculated by the trained network. The input concentrations written in bold indicate the change in the initial conditions compared to the reference system No. 1.

The network has been trained by using input and output data (first rows) No. 1–13. Notice that at every new run only one of the initial concentrations is changed. The close matching between the first and second rows of output data indicates that the network has "learned" to emulate the dynamical fingerprint of the BAM-system at the *given* values of input concentrations. The errors of predictions vary within a range from –2 to +2 %. Fig. 2a shows, for example, that the network almost accurately *reproduces* the values of $[Br^-]$ and reaction time corresponding to the extrema (■) at the initial concentrations given in Table I. No. 1.

Table I

Testing the ANN for the BAM-system on numerically generated data

	Input data				Output data					
No.	$[BrO_3^-]_0$	$[H_2A]_0$	$[H_2M]_0$	$[H^+]_0$	$t_{max,1}$	$[Br^-]_{max,1}$	t_{min}	$[Br^-]_{min}$	$t_{max,2}$	$[Br^-]_{max,2}$
1.	0.0200	0.040	0.040	1.00	89.84	1.1427	154.83	2.0811	235.05	2.4627
					89.40	*1.1445*	*154.15*	*2.0619*	*232.50*	*2.4762*
2.	**0.0195**	0.040	0.040	1.00	93.07	1.1441	156.80	2.4545	231.71	2.9585
					92.95	*1.1459*	*156.72*	*2.4554*	*230.80*	*2.9848*
3.	**0.0205**	0.040	0.040	1.00	86.86	1.1413	152.60	1.7918	236.00	2.0596
					86.52	*1.1426*	*152.02*	*1.7757*	*233.74*	*2.0611*
4.	**0.0210**	0.040	0.040	1.00	84.10	1.1399	150.32	1.5657	234.40	1.7398
					84.23	*1.1408*	*150.16*	*1.5876*	*234.30*	*1.7749*
5.	0.0200	**0.038**	0.040	1.00	88.57	1.0768	154.54	1.6438	225.42	1.7933
					87.91	*1.0757*	*153.34*	*1.6714*	*226.26*	*1.7812*
6.	0.0200	**0.042**	0.040	1.00	91.31	1.2090	154.25	2.7427	234.86	3.4906
					90.72	*1.2061*	*152.99*	*2.7702*	*235.26*	*3.4940*
7.	0.0200	0.040	**0.045**	1.00	90.59	1.1308	150.05	2.3979	209.71	2.7372
					90.17	*1.1312*	*149.97*	*2.3751*	*209.95*	*2.7161*
8.	0.0200	0.040	**0.050**	1.00	91.34	1.1188	146.34	2.7136	191.18	2.9915
					90.99	*1.1187*	*146.26*	*2.7138*	*189.72*	*2.9822*
9.	0.0200	0.040	**0.055**	1.00	92.10	1.1066	143.43	3.0259	176.99	3.2332
					91.89	*1.1072*	*143.10*	*3.0452*	*175.37*	*3.2513*
10.	0.0200	0.040	**0.060**	1.00	92.87	1.0942	141.20	3.3328	165.71	3.4679
					92.91	*1.0965*	*140.54*	*3.3396*	*166.41*	*3.4981*
11.	0.0200	0.040	0.040	**0.80**	130.75	1.1456	213.51	3.1690	255.72	3.2691
					129.49	*1.1449*	*211.74*	*3.1510*	*255.19*	*3.2446*
12.	0.0200	0.040	0.040	**0.90**	107.32	1.1446	180.06	2.5515	244.08	2.8067
					109.35	*1.1463*	*182.66*	*2.5846*	*244.20*	*2.8477*
13.	0.0200	0.040	0.040	**1.20**	65.93	1.1362	118.83	1.4403	223.44	1.9747
					66.66	*1.1359*	*119.48*	*1.4407*	*223.39*	*1.9674*
14.	**0.0203**	0.040	0.040	1.00	88.03	1.1419	153.51	1.8988	235.92	2.2102
					87.60	*1.1434*	*152.83*	*1.8771*	*233.32*	*2.2117*
15.	0.0200	**0.039**	0.040	1.00	89.18	1.1097	154.73	1.8405	231.39	2.0880
					88.70	*1.1063*	*154.11*	*1.8236*	*230.05*	*2.0563*
16.	0.0200	0.040	**0.043**	1.00	90.21	1.1368	152.28	2.2395	221.31	2.6031
					89.78	*1.1378*	*152.01*	*2.2135*	*221.34*	*2.5918*
17.	0.0200	0.040	0.040	**1.10**	76.44	1.1398	134.93	1.7202	228.37	2.1930
					75.13	*1.1405*	*132.89*	*1.6815*	*224.86*	*2.1803*
18.	**0.0207**	**0.040**	**0.047**	**1.15**	68.20	1.1047	117.82	1.4740	189.77	1.7318
					70.15	*1.1038*	*122.31*	*1.4909*	*194.73*	*1.6994*
19.	**0.0197**	**0.041**	**0.052**	**0.85**	124.33	1.1446	184.91	4.6244	200.05	4.6560
					124.51	*1.1480*	*189.39*	*3.9267*	*208.15*	*4.0279*
20.	**0.0180**	0.040	0.040	1.00	104.70	1.1478	160.17	4.3236	210.96	5.0011
					107.40	*1.1377*	*168.88*	*3.6679*	*227.62*	*4.0414*
21.	0.0200	**0.036**	0.040	1.00	87.46	1.0111	154.72	1.3462	207.15	1.3850
					86.27	*1.0408*	*150.24*	*1.5304*	*215.31*	*1.5194*
22.	0.0200	0.040	**0.025**	1.00	87.67	1.1772	186.81	1.1487	436.82	1.3514
					87.77	*1.1836*	*168.77*	*1.4436*	*267.74*	*1.9982*
23.	0.0200	0.040	0.040	**0.73**	152.28	1.1455	247.28	3.7075	262.42	3.7135
					140.14	*1.1420*	*228.04*	*3.4912*	*260.65*	*3.4975*
$X_{i,min}$	0.015	0.030	0.020	0.60	50.00	1.0000	80.00	1.000	150.00	1.0000
$X_{i,max}$	0.025	0.055	0.060	1.40	150.00	1.2100	250.00	4.0000	260.00	4.0000

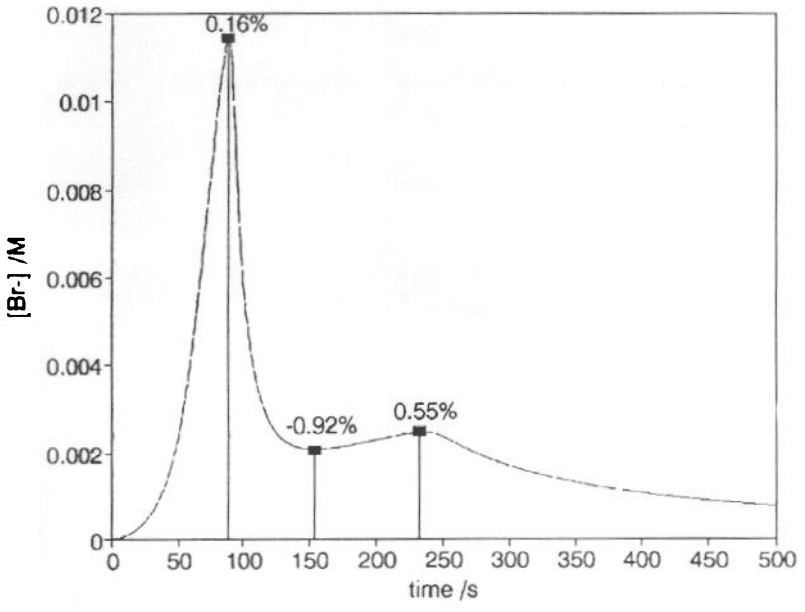

Fig. 2a. Testing the ANN on numerically generated data: the trained network accurately reproduces the values of $[Br^-]$ and reaction time corresponding to the extrema (▪). Initial concentrations are given in Table I, No. 1

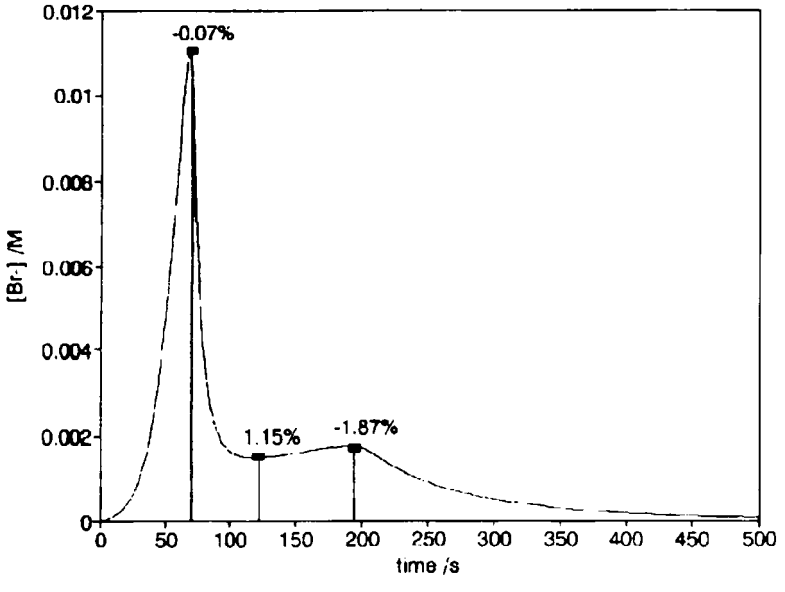

Fig. 2b. Prediction for the dynamical fingerprint of the BAM-system at initial concentrations not included in the training (Table I, No. 18). The percent values (error of prediction) shown in the figures have been calculated by the following formula: $100\times([Br^-]_{ANN}-[Br^-]_{NUM})/[Br^-]_{NUM}$, where $[Br^-]_{ANN}$ is the predicted bromide ion concentration at a given extremum, and $[Br^-]_{NUM}$ is the numerically calculated value (by solving the ODEs of the BAM-system). The vertical lines in the figures are to better read the predicted locations of the extrema on the time axes

The predictive power of the trained network has been tested first by using input data No. 14–17. These input values fall into the ranges of initial concentrations applied for training but were not included in the learning algorithm. Notice that only one of the initial concentrations has been changed in these runs, too. The predicted output values are in good agreement with those calculated by integrating the ODEs. On the average, the accuracy of predictions is within the same range that has been achieved during training, which is a clear indication that no over-training of the network took place. In other words, the network did not "memorize" but, instead, properly generalized the nonlinear rules embedded in the training data.

To further test the network, in run No. 18–19 *none* of the initial concentrations were chosen from those applied earlier. The prediction for the dynamical fingerprint of the system is still quite good for run No. 18 (Fig. 2b) but is rather poor for run No. 19. Note that in the latter case the numerically calculated bromide concentrations are very close to or above the upper limits applied for scaling. Since the algorithm optimized the weights so as to produce (scaled) output values within the range of training, it is not surprising that the trained network is unable to accurately predict the values outside this range. This finding indicates that the prediction of dynamical behavior might easily fail in case of highly nonlinear chemical systems, even if the new input values were within the range of training. A quick solution to this problem is the rescaling of data followed by a new training session. However, the only chance to avoid such a failure at first is to apply as many training datasets as possible.

Another important limitation of ANN calculations mentioned earlier is that, generally, a trained network does not work well with input values outside the range of training. It is also illustrated by the results listed in Table I, No. 20–23, where one of the input values is always outside this range. After all, the results shown above justified that a feedforward artificial network can be trained to predict the dynamical fingerprint of the BAM-system.

Testing the network on experimental data

Experiments were carried out as described earlier by Rábai et al. [7, 8]. The change in bromide-ion concentration was followed by measuring the potential of a bromide-ion selective electrode. The output data for training were taken at the extrema of the measured bromide-ion concentration versus time curves.

The network has been trained by using input and output data (after scaling) listed in Table II, No. 1–13. The accuracy of predictions is much less than that has been found by using numerically generated data. Nevertheless, on the average, it matches the range of the experimental error (approx. ±5%). Fig. 3a shows, for example, that

the trained network is capable of predicting the dynamical fingerprint of the BAM-system within the accuracy of the experiments (except for the value of $t_{max,1}$).

Table II

Testing the ANN for the BAM-system on experimental data

	Input data				Output data					
No.	$[BrO_3^-]_0$	$[H_2A]_0$	$[H_2M]_0$	$[H^+]_0$	$t_{max,1}$	$[Br^-]_{max,1}$	t_{min}	$[Br^-]_{min}$	$t_{max,2}$	$[Br^-]_{max,2}$
1.	0.0200	0.040	0.040	1.00	87.00	1.0437	121.50	1.1140	192.25	2.2620
					77.47	*1.0472*	*115.43*	*1.1571*	*196.57*	*2.2533*
2.	**0.0175**	0.040	0.040	1.00	84.00	0.8500	119.00	2.1290	182.50	3.7280
					87.16	*0.8634*	*118.39*	*2.0767*	*175.60*	*3.7877*
3.	**0.0225**	0.040	0.040	1.00	65.00	0.8910	99.00	0.6480	179.50	1.0790
					68.10	*0.9084*	*98.30*	*0.5957*	*173.67*	*1.0581*
4.	0.0200	**0.035**	0.040	1.00	82.50	0.8572	112.50	0.6210	186.00	1.1800
					79.06	*0.8465*	*111.74*	*0.6832*	*186.13*	*1.1304*
5.	0.0200	**0.045**	0.040	1.00	80.75	1.0940	110.75	1.5440	183.50	3.4680
					74.40	*1.0971*	*110.50*	*1.6060*	*186.81*	*3.3403*
6.	0.0200	0.040	**0.025**	1.00	90.25	1.0339	143.50	1.1140	265.50	2.2690
					91.04	*1.0189*	*140.91*	*1.1080*	*248.15*	*2.2715*
7.	0.0200	0.040	**0.030**	1.00	83.50	0.9801	131.00	1.1520	229.00	2.1650
					86.52	*1.0229*	*132.63*	*1.1061*	*233.36*	*2.2396*
8.	0.0200	0.040	**0.050**	1.00	85.50	1.2244	113.50	1.2620	168.50	2.5490
					73.74	*1.1252*	*106.59*	*1.4564*	*159.22*	*2.6075*
9.	0.0200	0.040	**0.055**	1.00	70.00	1.1234	106.50	1.9710	146.00	3.0840
					76.98	*1.2004*	*109.67*	*1.8170*	*144.62*	*3.0536*
10.	0.0200	0.040	0.040	**0.80**	136.50	0.9107	188.00	1.7660	229.50	2.5400
					132.76	*0.8785*	*183.11*	*1.8282*	*224.51*	*2.5263*
11.	0.0200	0.040	0.040	**0.90**	95.00	0.8500	139.50	1.5340	198.50	2.3350
					100.71	*0.9256*	*145.48*	*1.3505*	*213.29*	*2.2401*
12.	0.0200	0.040	0.040	**1.10**	58.75	1.2411	94.00	1.1290	165.50	2.3830
					61.35	*1.2033*	*94.15*	*1.0519*	*173.46*	*2.3014*
13.	0.0200	0.040	0.040	**1.20**	47.00	1.3088	80.25	0.9540	156.50	2.2350
					51.08	*1.3434*	*81.10*	*0.9550*	*149.63*	*2.2726*
14.	0.0200	0.040	**0.035**	1.00	79.00	1.0683	115.50	1.1480	208.00	2.3330
					81.80	*1.0313*	*123.77*	*1.1176*	*215.88*	*2.2265*
15.	0.0200	0.040	**0.045**	1.00	77.50	0.9696	111.50	1.4380	168.50	2.4620
					74.39	*1.0760*	*109.14*	*1.2530*	*177.02*	*2.3592*
16.	0.0200	0.040	**0.020**	1.00	90.00	0.9847	156.50	0.7180	314.00	1.3750
					95.05	*1.0174*	*148.03*	*1.1163*	*260.14*	*2.3111*
$X_{i,min}$	0.015	0.030	0.020	0.60	40.00	0.5000	70.00	0.5000	100.00	1.0000
$X_{i,max}$	0.025	0.055	0.060	1.40	160.00	1.5000	200.00	2.5000	350.00	4.0000

In run No. 14–15, the initial concentrations of malonic acid have been changed to such values that were not included in the learning algorithm. The results of prediction for set No. 14 are shown in Fig. 3b. Again, the accuracy of predictions for the dynamical fingerprint of the BAM-system is comparable to the error of the experiments. In run No. 16, the input concentration of malonic acid is outside the range applied for training, therefore, the prediction is quite poor.

The predictive power of the network could surely be improved by using a much larger data set. Since such an effort would have no practical value in this case (the mechanism of the reaction is known), no further experiments were performed to achieve a better agreement between predictions and experiments.

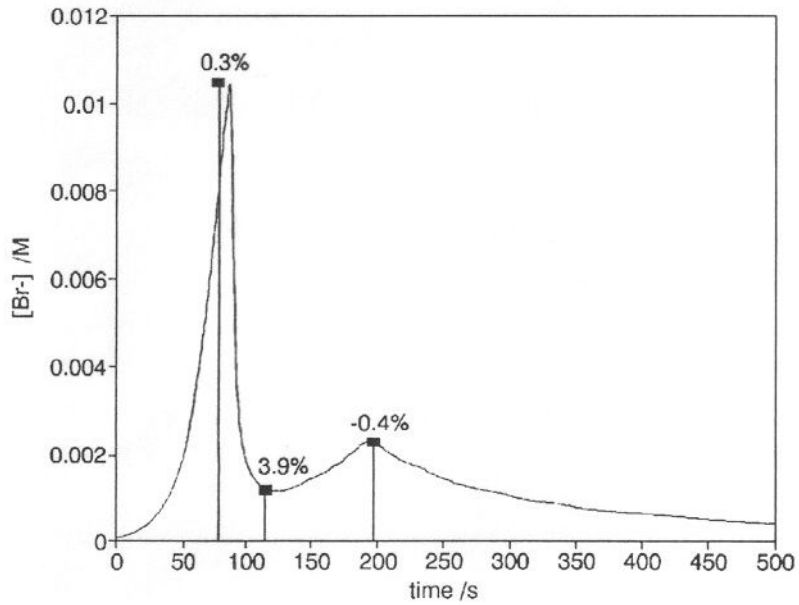

Fig. 3a. Testing the ANN on experimental data: the trained network reproduces the experimental values of reaction time and [Br^-] corresponding to the extrema (▪). Initial concentrations are given in Table II, No. 1

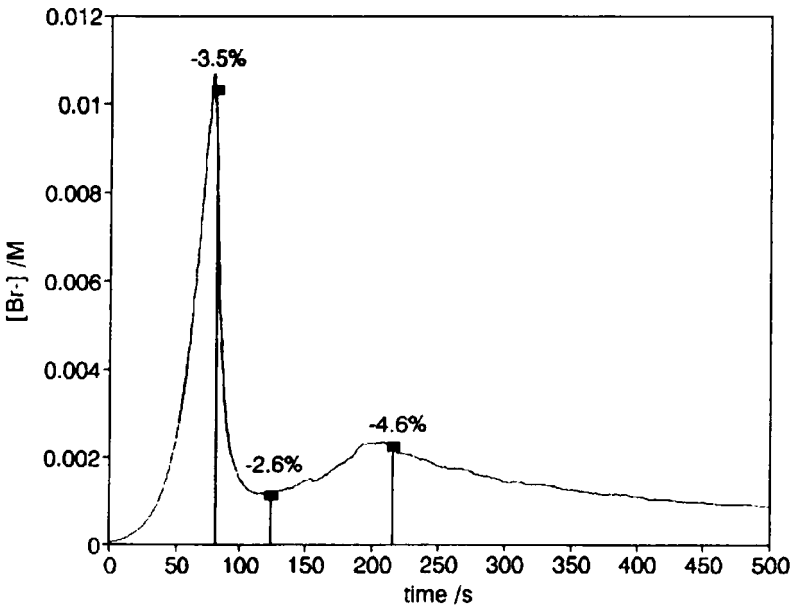

Fig. 3b. The network predicts the dynamics of the BAM-system within the accuracy of the experiments. For initial concentrations (not included in the training set) see Table II, No. 14

Conclusion

It has been shown by using numerically generated and experimentally obtained data that a feedforward neural network can be trained to predict the main dynamical features of oligo-oscillations in the BAM-system. An advantage of the neural network approach over the classical method of reaction kinetics is that it does not require the prior knowledge of the underlying chemical mechanism. Another advantage of ANNs shown by the results is that the handling of nonlinear data is easy for neural networks because of their inherently nonlinear response. The presented results indicate that ANNs can be efficiently employed for the prediction of the dynamics of complex chemical systems, especially, when there is only little hope to fully establish a deterministic model. However, as shown also by the results, the algorithm should be used with great care when predictions are made close to the limits of or outside the range of training.

A serious disadvantage of neural computations is that they lack the potential for *physical* interpretation of the adjustable parameters of an ANN model (weights) as well as of the investigated system itself. Therefore, by boosting the ANN approach for solving problems in reaction kinetics, we do not want to trigger a tendency in research that neglects or demotes studies on the mechanistic details of complex chemical systems. Such an aberration would surely initiate a rapid degeneration of our field. At the same time, one can find practical problems to which neural computations seem to be the perfect solution. For instance, predicting time series data of highly nonlinear chemical systems in laboratory experiments [19–20], controlling industrial processes (Chapter 16. in ref. [4]), controlling chaos [21], or estimating the decay of drugs or other substances in metabolic processes, etc. Investigations into using different neural networks for such tasks are currently being carried out in our group, and the results will be reported elsewhere.

*

This work has been partially supported by the Hungarian Scientific Research Fund (OTKA), grant No. T 014243. V.G. is grateful to Prof. K. Showalter and Dr. D. Horváth, Dept. of Chemistry, West Virginia University, Morgantown, WV, USA for introducing him to the field of artificial neural networks. The authors thank Dr. I. Koós, Dept. of Theoretical Physics, Kossuth L. University, Debrecen, Hungary and Dr. P. Érdi, Res. Inst. for Particle and Nuclear Physics, Budapest, Hungary for enlightening discussions on the subject.

References

[1] ZHABOTINSKY, A. M.: *Dokl. Akad. Nauk SSSR*, **157**, 392 (1964)

[2] GAO, Y., FÖRSTERLING, H.-D., NOSZTICZIUS, Z., MEYER, B.: *J. Phys. Chem.*, **98**, 8377 (1994)

[3] BODOR, N., HARGET, A., HUANG, M.-J.: *J. Am. Chem. Soc.*, **113**, 9480 (1991)

[4] ZUPAN, J., GASTEIGER, J.: *Neural Networks for Chemists: An Introduction*, Verlag Chemie, Weinheim, 1993

[5] SPINING, M. T., DARSEY, J. A., SUMPTER, B.G., NOID, D.W.: *J. Chem. Educ.*, **71**, 406 (1994)

[6] RUMELHART, D. E., MCCLELLAND, J. L.: *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT Press, Cambridge, 1986

[7] RÁBAI, G., BAZSA, G., BECK, M. T.: *J. Am. Chem. Soc.*, **101**, 6746 (1979)

[8] RÁBAI, G., BAZSA, G., BECK, M. T.: *Int. J. Chem. Kinetics*, **13**, 1277 (1981)

[9] BECK, M. T., RÁBAI, GY.: *Kémiai Közlemények*, **68**, 116, (1987-88)

[10] BECK, M. T., RÁBAI, GY.: *J. Chem. Soc. Dalton*, 1687 (1982)

[11] HINDMARSH, A.C.: *Livermore Solver of Ordinary Differential Equations (LSODE)*, Lawrence Livermore Laboratory, Livermore, CA, 1980

[12] ANDREA, T. A., KALAYEH, H.: *J. Med. Chem.*, **34**, 2824 (1991)

[13] TETKO, I. V., LUIK, A. I., PODA, G. I.: *J. Med. Chem.*, **36**, 811 (1993)

[14] AJAY: *J. Med. Chem.*, **36**, 3565 (1993)

[15] SUMPTER, B. G., GETINO, C., NOID, D. W.: *Annu. Rev. Phys. Chem.*, **45**, 439 (1994)

[16] BODOR, N., HUANG, M.-J., HARGET, A.: *J. Mol. Structure (Theochem)*, **309**, 259 (1994)

[17] LIVINGSTONE, D. J., MANALLACK, D. T.: *J. Med. Chem.*, **36**, 1295 (1993)

[18] CHITRA, S. P., BULSON, R. J., MORRELL, D. G.: *Chemtech*, 18 (Febr. 1995)

[19] HUDSON, J. L., KUBE, M., ADOMAITIS, R. A., KEVREKIDIS, I. G. LAPEDES, A. S., FARBER, R. M.: *Chem. Eng. Sci.*, **45**, 2075 (1990)

[20] LEBENDER, D., SCHNEIDER, F. W.: *J. Phys. Chem.*, **97**, 8764 (1993)

[21] ALSING, P. M., GAVRIELIDES, A., KOVANIS, V.: *Phys. Rev. E*, **49**, 1225 (1994)