Journal of Thermal Analysis, Vol. 49 (1997) 1553-1564

NEURAL NETWORKS FOR KINETIC PARAMETERS DETERMINATION, SIGNAL FILTERING AND DECONVOLUTION IN THERMAL ANALYSIS

N. Sbirrazzuoli, D. Brunel and L. Elégant

Laboratory of Experimental Thermodynamics U.M.R.-C.N.R.S-139, University of Nice-Sophia Antipolis, 06108 Nice Cedex 2, France

Abstract

Feedforward neural networks have been used for kinetic parameters determination and signal filtering in differential scanning calorimetry. The proper learning function was chosen and the network topology was optimized, using an empiric procedure. The learning process was achieved using simulated thermoanalytical curves. The resilient-propagation algorithm have led to the best minimization of the error computed over all the patterns. Relative errors on the thermodynamic and kinetic parameters were evaluated and compared to those obtained with the usual thermal analysis methods (single scan methods). The errors are much lower, especially in presence of noisy signals. Then, our program was adapted to simulate thermal effects with known thermodynamic and kinetic parameters, generated electrically, using a PC computer and an electronic interface on the serial port. These thermal effects have been generated by using an inconel thread.

Keywords: deconvolution, differential scanning calorimetry, feedforward neural networks, kinetics, signal filtering, simulations, thermal analysis

Introduction

Artificial neural networks is a computing approach based on an analogy with the working of the nervous system of the brain, in which connections organize the units (neurons) into networks. One of the main advantages of the artificial neural networks is that they are massively parallel and can improve their performances through examples by a dynamic learning process. Therefore, the use of artificial neural networks allows a modelling of complex systems by way of interconnection weights, without requiring the explicit formulation of the relationships that may exist between variables. These weights are generated by the training that starts from random values of the coefficients (bias). It has been established that a standard layered feedforward network architecture can approximate any function of interest, provided that a sufficient number of hidden neurons are used [1, 2]. Furthermore, artificial neural networks are highly tolerant to noisy or missing data in the training or test set samples, and suitable for pattern recognition, reconstruction applications, filtering or time-series predictions.

One of the greatest developments of these last years is an attempt to extracting kinetic information of a transformation by thermal analysis [3]. An application of

0368–4466/97/ $ 5.00

John Wiley & Sons Limited
Chichester

artificial neural networks to DSC is presented in this study, and is generalizable to any thermal analysis [4].

Deconvolution of differential scanning calorimetric curves

Usual deconvolution methods

Several methods have been proposed for the determination of thermokinetics and are reported in the work of Cesari *et al.* [5]. Four kinds of methods used for the reconstruction of thermogenesis from experimental calorimetric data may be distinguished: Fourier transform analysis, dynamic optimization method, method using the state functions, differentiation method (numerical inverse filters). Nevertheless, these methods may be sensitive to noise.

New approach developped

Some workers have already reported on several applications in which neural networks have been used with success for deconvolution purposes in various fields of research [6–9], but not for calorimetric field. Furthermore, in addition to the high capability of neural networks to deal with nonlinear problems, it is known that they are highly tolerant to noise, and numerous authors have reported on very good robustness in the presence of noisy signals [10–12].

In the new approach developped in our laboratory, the idea is to use simulated thermal effects with known thermodynamic and kinetic parameters, generated electrically, using a PC computer and an electronic interface on the serial port. These thermal effects have been generated by using an inconel thread and recorded by the apparatus as for an experimental study. The differential scanning calorimeter used

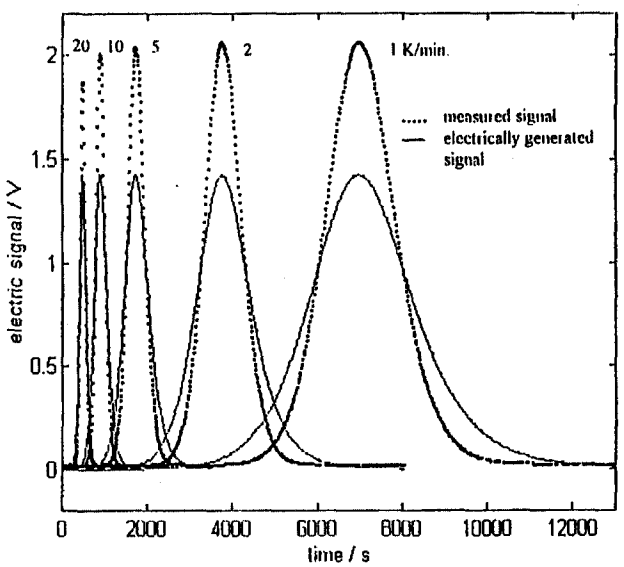

Fig. 1 Electrically generated and measured signals at 1, 2, 5, 10 and 20 K min^{-1}

was a DSC-111 heat-flux calorimeter (Setaram). This provides a way to evaluate the transfer function for a transformation occuring during a certain interval of time, depending on its kinetic parameters and not only applicable to a Dirac pulse or an unit step function. Figure 1 shows the various thermoanalytical curves obtained at 1, 2, 5, 10 and 20 K min^{-1} for electrically generated signals. Signals of Fig. 1 are the electric signals in Volts; for calorimetric purposes these signals are transformed in Watts (or mW) by an appropriate calibration.

The aim of this first study is to evaluate the ability of neural networks for the determination of thermodynamic and kinetic parameters from the analysis of a thermoanalytical curve, to find the network topology that best modelises the DSC signals, and to optimise the network topology. In order to simulate experimental data, neural networks performances were investigated using simulated noisy signals.

Network performance evaluation

Our previous studies [3, 13] were based on the minimization of a criterion called *LSM* that expresses the accuracy of the fit between experimental (i.e. a thermogram of a transformation) and calculated data (i.e. the values computed from the kinetic parameters of the transformation) and that is expressed as:

$$LSM = \frac{1}{N}\sum_{i=1}^{N}(Y_{i,exp} - Y_{i,calc})^2 \quad (1)$$

where N is the number of recorded points of a thermoanalytical curve and Y_i represents the heat flow measured (exp) or calculated (calc) from the kinetic parameters, at time i.

On the other hand, the accuracy of the net may be evaluated, for supervised learning algorithms, by computing a parameter which represents the residual between outputs (O) and targets values (T), we computed the residuals as,

$$Sr = \left(\frac{1}{p}\right)\sum_{j=1}^{p}(O_j - T_j)^2 \quad (2)$$

where p represents the number of patterns. In addition to this parameter the neural network simulator used [14] gives an evaluation of the global error (E) of the net over all the patterns and all the outputs. The error E is defined as:

$$E = \frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{n}(O_{ij} - T_{ij})^2 \quad (3)$$

where n is the dimensionality of the output vector (here the dimensionality of this vector is 3) and p the number of patterns used in the learning phase. This parameter has to be minimized during the learning phase where the mapping of the input vector on to the output vector determines the connection weights of the net.

Numerical simulations of thermoanalytical curves

The networks were trained and tested using simulated thermoanalytical curves with a computer code previously described [13, 15]. These simulations consisted in the computation of the temperature (T_i), of the power (P_i) and the conversion range (α_i) of a known transformation from known kinetic parameters, sampling rate, heating rate (V=5°C min^{-1} here), temperature of the beginning of the phenomenon, function f(α_i) and enthalpy (Q). The general kinetic equation for the reaction rate is expressed as,

$$\left(\frac{d\alpha}{dt}\right)_i = k_i f(\alpha_i) \tag{4}$$

where $\alpha_i = H_i/Q$ for a DSC analysis and where the dependence of the rate constant (k_i) *vs.* the temperature is given by the Arrhenius law, $k_i = k_o \exp(-E_a/RT_i)$. So that, E_a, k_o and R represent respectively the activation energy, the pre-exponential factor and the gas constant (8.31 J mol^{-1} K^{-1}), and H_i the partial area computed at time i. We can write:

$$\left(\frac{dH}{dt}\right)_i = Q k_i f(\alpha_i) \tag{5}$$

where $(dH/dt)_i$ represents the heat flux (P_i) of a flux-meter DSC scan at time i.

Network description

Input, output and patterns definition

In this study, the set of examples (patterns) is constitued by various simulated thermograms, used for training. Each thermoanalytical curve of the set was discretised in 320 inputs presented to the network. These inputs were generated using a constant value for the order of reaction (n=2), while the activation energy (E_a) was varying from 74 to 80 kJ mol^{-1} and the logarithm of the pre-exponential factor from 18 to 20, according to the kinetic parameters experimentaly found, for the reaction of our interest [16]. The input vector is constructed with the various powers, and the number of inputs had been adjusted in order to optimize the network configuration. The output vector, called « target » for the learning phase, is constructed with the three parameters to be determined i.e., Q, E_a and $\ln k_o$ (because n was set constant here), and are the same as the parameters used to construct the thermoanalytical curve. They are sufficient to define the thermoanalytical curve in the case of a single step process. In order to simulate experimental data, additional gaussian noise may be added to the patterns. In that case, simulated thermoanalytical curves were computed in the same way, and then noise was added, so that the output vector (target) may be constructed with the known kinetic and thermodynamic parameters obtained for the signal without noise, and the network was so trained to identify the

noise. The standard deviation of the gaussian noise used for all this study is of 0.2 and the mean is 0. This value was chosen in regard to the standard deviation of the noise of the apparatus used (DSC-111 Setaram). In the temperature domain studied, the distribution of this noise can be considered as gaussian, and the standard deviation was experimentally evaluated of about 0.07, so that the experimental noise is greatly inferior to the value used in simulated data.

For the test set, the patterns are generated in the same way, but correspond to different values of the thermodynamic and kinetic parameters, choosen in the range of the values used in the training set.

Learning function

Several algorithms have been checked and the best results were obtained using the "resilient propagation", developed by Riedmiller [17]. It is not the purpose of the present paper to give an extensive description of any of them, and more information on these algorithms may be found in the references [18–19]. A comparison of the results obtained using the well known backpropagation and the resilient-propagation algorithms is given in the result part. For more details on this algorithm as well as on the backpropmomentum or the backpercolation algorithms, also presented as comparison in this study, the reader may consult references [17–19].

Experimental

The selection of the learning function, of the number of patterns and of the number of inputs, was made using noiseless data and one hidden layer with fifteen units. In a first step, optimization of the network was realised by comparing the errors (E formula 10) after 500 learning cycles. This value was chosen because of the limitation in the precision of the software used. In fact, the variables are set in SNNS [14] as «float» whose precision may be too limited for this kind of calculation using simulated phenomena. The error values (E) quickly reach, in some cases, the limit of the precision on the numbers. Finally, the type «float» has been changed in the program used (about 600 modifications) by «double» precision, in regard to a first set of results obtained for the test phase, for which the precision of the «float» was not sufficient.

Learning algorithm, number of patterns and network topology

The learning parameters used for each function are set out in Table 1. The superiority of resilient-propagation is obvious. The activation energy and logarithm of the pre-exponential factor values were incremented from $E_{a_i}=74$ to $E_{a_f}=80$ kJ mol^{-1} and $\ln k_{0_i}=18$ to $\ln k_{0_f}=20$, using various incrementation steps. Among the tested values, the lowest errors were obtained using 420 patterns, that is to say a step on the activation energy of 300 and a step on the logarithm of the pre-exponential factor of 0.1 (Table 2). An optimal value of 320 inputs was kept for all the networks tested in the following. Nevertheless, we think that this value should be optimized

Table 1 Simulated data without noise: errors (E) obtained using various learning algorithms, after 500 and 3000 learning cycles

Algorithms	E[a] (500 cycles)	E[a] (3000 cycles)
backpropagation[b]	0.09836	0.09289
backpropmomentum[c]	0.09761	0.07690
quickpropagation[d]	0.09933	0.04592
backpercolation[e]	0.04291	0.01437
resilient-propagation[f]	0.00005	0.00003

[a] $E = \frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{n}(O_{ij} - T_{ij})^2$

n is the dimensionality of the output vector and p the number of patterns used in the learning phase

[b] learning rate $\eta = 0.1$

[c] learning rate $\eta = 0.1$, momentum term $\mu = 0.1$

[d] learning rate $\eta = 0.1$, maximum growth parameter $\mu = 0.1$

[e] global error magnification $\lambda = 1$, $\Theta = 1$

[f] $\Delta_0 = 0.1$, $\Delta_{max} = 50$ [14]

by selecting a great number of points in the parts of the thermograms containing more information (beginning, top, inflexions and end of peak), and less in the other parts. For noiseless data, the lowest error was obtained using one hidden layer with fifteen neurons (Table 3). Nevertheless, there is no general theory for the prediction of the best topology to choose, so that an experimental procedure is generally used. For simulated data with gaussian noise, the following procedure was used to optimize the network topology: when the number of neurons was selected for the first hidden layer, a second layer was added and the number of neurons leading to the lowest error retained, and so on. A hidden layer was only added if the error was decreased. A network with only one hidden layer would lead to satisfactory results

Table 2 Simulated data without noise: influence of the number of patterns obtained using various incrementation steps, after 500 learning cycles[a]

Step on E_a	Step on $\ln k_0$	$Sr^b(Q)$	$Sr^b(E_a)$	$Sr^b(\ln k_0)$
300	0.05	$4.7 \cdot 10^{-9}$	$4.14 \cdot 10^{-6}$	$4.30 \cdot 10^{-6}$
300	0.1	$1.2 \cdot 10^{-9}$	$9.08 \cdot 10^{-7}$	$1.51 \cdot 10^{-6}$
300	0.2	$1.2 \cdot 10^{-7}$	$1.20 \cdot 10^{-5}$	$4.34 \cdot 10^{-3}$
100	0.1	$7.2 \cdot 10^{-9}$	$3.25 \cdot 10^{-6}$	$4.22 \cdot 10^{-6}$
150	0.1	$3.1 \cdot 10^{-9}$	$3.69 \cdot 10^{-6}$	$5.05 \cdot 10^{-6}$
200	0.1	$4.9 \cdot 10^{-9}$	$3.12 \cdot 10^{-6}$	$2.80 \cdot 10^{-6}$
300	0.1	$1.2 \cdot 10^{-9}$	$9.08 \cdot 10^{-7}$	$1.51 \cdot 10^{-6}$

[a] same learning parameters and network topology as in Table 1

[b] $Sr = \left(\frac{1}{p}\right)\sum_{j=1}^{p}(O_j - T_j)^2$

p represents the number of patterns, and Sr is computed for the learning set of examples

Table 3 Simulated data without noise: errors (E) obtained for various numbers of neurons of the first hidden layer, after 500 learning cycles[a]

Number of neurons	E[b]
10	0.0310
12	0.0521
15	0.0225
17	0.0488
20	0.0335
50	0.0402
100	0.2740

[a] same learning parameters and network topology as in Table 1
[b] see Table 1

for noiseless data, while when noise is added the learning is improved by using several hidden layers. Our results confirm the statement of Ishiwatari *et al.* [20], who found more robustness for five-layered network (i.e. three hidden layers) than for classical three-layered network, in the presence of noise.

Finally, the network used is constituted by an input layer of 320 units, 3 hidden layers with respectively 15, 20 and 20 units, and an output layer with 3 units. The inputs were generated using simulated thermograms, and the targets were the parameters used for the simulation of the thermograms. The learning phase was achieved on 420 patterns, and the learning function retained was the resilient-propagation.

Results and discussion

Comparisons between the backpropagation and the resilient-propagation algorithms may be drawn from the evolution of the error *vs.* the number of learning cycles with noiseless data. The backpropagation algorithm always led to higher values of the error E and the results are slightly improved by performing more learning cycles. The error value is respectively of $9.8 \cdot 10^{-2}$ after 500 cycles and $9.3 \cdot 10^{-2}$ after 3000 cycles, while for the resilient-propagation, these values are respectively $5 \cdot 10^{-5}$ and $2.52 \cdot 10^{-5}$ (Table 1). In the classic «gradient descent» minimization technique used in the conventional backpropagation algorithm, the learning rate must be defined at the beginning of the training. The choice of the learning rate cannot be optimal because this parameter depends on the shape of the error function, which changes with the learning task. For the resilient-propagation the adaptative individual update-value Δ_{ij} introduced for each weight and modified during the learning process according to the learning rule, improve the convergence. This method is robust with respect to the choice of the initial learning parameter. The resilient propagation algorithm out-performs backpropagation because of the existence of local minima. The existence of such minima had also be shown by comparing various multiple linear regression algorithms, in a previous study on single scan methods [15].

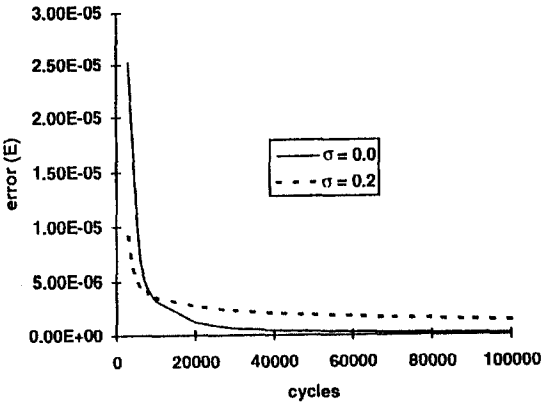

Fig. 2 Resilient-propagation algorithm: evolution of the error (E) for the learning set *vs.* the number of learning cycles for noiseless data ($\sigma=0.0$) and data with gaussian power noise ($\sigma=0.2$)

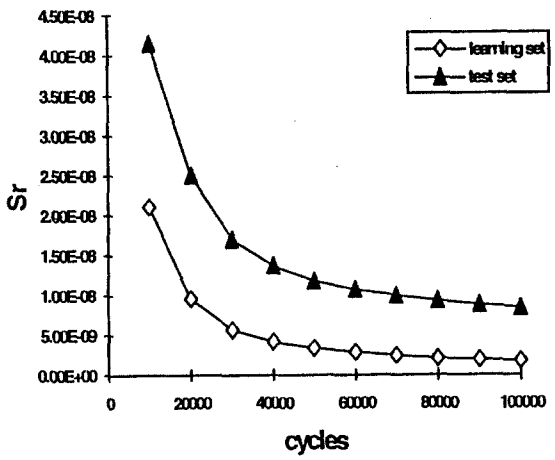

Fig. 3 Resilient-propagation algorithm: evolution of *Sr* (mean squared residuals) *vs.* the number of learning cycles for the activation energy

Excepted for the resilient-propagation, the backpercolation and the quickpropagation algorithms led to the best results, but the error is much higher compared to the resilient-propagation (Table 1). For the resilient-propagation algorithm the errors of the learning phase presented in Fig. 2, have been drawn *vs.* the number of learning cycles for noiseless and noisy data. In such a case, the SNNS simulator gives the error value for the outputs every 10000 cycles. The errors show a rapid decrease to a final value after 100000 cycles of respectively $1.91 \cdot 10^{-7}$ and $1.34 \cdot 10^{-6}$ for noiseless data and for noisy data ($\sigma=0.2$).

In Fig. 3 the evolution of the *Sr* for the various targets is presented *vs.* the number of learning cycles, for simulated data without noise. Comparisons can be made

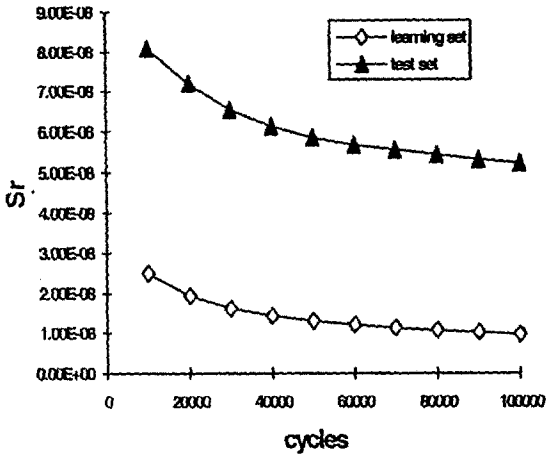

Fig. 4 Resilient-propagation algorithm: evolution of *Sr* (mean squared residuals) *vs.* the number of learning cycles for the activation energy, gaussian noise $\sigma=0.2$

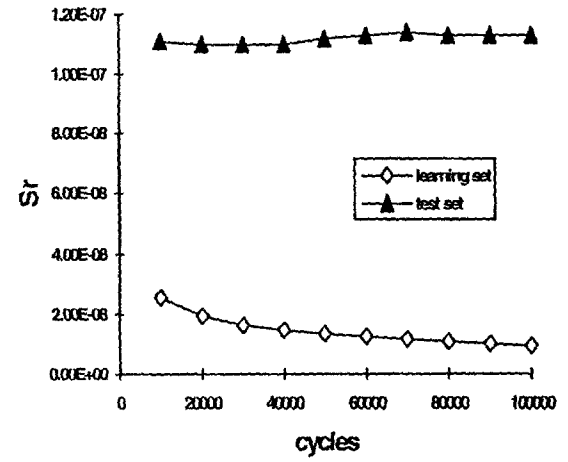

Fig. 5 Resilient-propagation algorithm: evolution of *Sr* (mean squared residuals) *vs.* the number of learning cycles for the logarithm of the pre-exponential factor, gaussian noise $\sigma=0.2$

between the values obtained for the learning and for the test sets (which are different sets). These variations may be compared to those obtained for simulated noisy data (Fig. 4). In that case, the *Sr* values are always decreased when the number of learning cycles is increased, as is obtained with noiseless data. Nevertheless, this parameter increases for the test set after 40000 cycles, for the logarithm of the pre-exponential factor (Fig. 5). This observation indicates that above this value the network is no longer trained to learn pertinent information and tend to learn noise. This corresponds to what is generally called overtraining and implies that the learning phase must be stopped at this stage of the learning process.

In order to give comparisons between the results presented in this study and those obtained with the usual methods (Achar-Brindley-Sharp or multiple linear regression methods) used in thermal analysis and presented elsewhere [4, 15], absolute relative errors were computed. These errors were computed for a parameter x, as:

$$er(x) = |(T - O)/T| \quad (6)$$

where T represents the target value of the parameter x and O the output estimated by the network. As an example, the relative errors can be found in Table 4 for data with and without noise. In the case of noisy data, average relative errors obtained for five sets of thermograms. The kinetic parameters of reference were always those specified in Table 4.

Table 4 Simulated data with various amounts of gaussian noise[a] on the power [15]: average relative errors for the Achar-Brindley-Sharp (ABS) and multiple linear regression (MLR(n)) methods, for five sets of thermoanalytical curves

Method	$er(n)^{a,b}$	$er(\ln k_0)^{a,b}$	$er(E_a)^{a,b}$
$\sigma=0$		$er(Q)^b=9.66\cdot10^{-6}$	
ABS	$5.00\cdot10^{-3}$	$2.52\cdot10^{-4}$	4.17·10–4
MLR(n)	$6.73\cdot10^{-5}$	3.53·10–4	$4.91\cdot10^{-4}$
$\sigma=0.2$		$er(Q)^b=1.15\cdot10^{-2}$	npmr[c] = 1.45%
ABS	$4.60\cdot10^{-2}$	3.49·10–2	$2.55\cdot10^{-2}$
MLR(n)	$4.65\cdot10^{-2}$	3.44·10–2	$2.51\cdot10^{-2}$

[a] n, kinetic exponent, $\ln k_0$, logarithm of pre-exponential factor, E_a, activation energy (kJ mol^{-1})
[b] theoretical parameters used for the simulations: $Q=77$ J, $n=2$, $\ln k_0=19$, $E_a=77$ kJ mol^{-1}
$er(x)$: relative error on the parameter x
[c] npmr: noise to peak maximum ratio, characteristics of the noise: mean=0 and standard deviation $\sigma=0.2$

Several thermoanalytical phenomena were generated from various parameters incrementation and were used for the learning (420 patterns), for artificial neural network. Table 5 gives the relative errors values obtained using the resilient-propagation algorithm, that always led to a very good accuracy. In the case of noiseless data, the resilient-propagation algorithm improves the results by a factor of 10 for the enthalpy, by a factor varying from 10 to 100 for the logarithm of the pre-exponential factor and by a factor of 10 for the activation energy. When noisy data are used, the difference between this algorithm and the usual methods is greater, the results are improved by a factor of 100 for the enthalpy and for the logarithm of the pre-exponential factor and by a factor varying from 100 to 1000 for the activation energy.

Our recent studies tend to reduce the size of the network, to save time during the learning phase, without decreasing the precision obtained. We have shown that this is possible when a lower number of pattern is used, as with experimental curves (even measured simulated curves). In this case, the measured thermoanalytical

Table 5 Relative errors obtained for the parameters with the network

σ^a	$er(Q)^a$	$er(\ln k_0)^a$	$er(E_a)^a$	Number of cycles
0.0	$7.00 \cdot 10^{-7}$	$3.36 \cdot 10^{-5}$	$3.05 \cdot 10^{-5}$	100000
0.0	$8.60 \cdot 10^{-7}$	$2.78 \cdot 10^{-5}$	$5.18 \cdot 10^{-5}$	
0.0	$4.80 \cdot 10^{-7}$	$3.55 \cdot 10^{-6}$	$5.13 \cdot 10^{-5}$	
0.2	$1.04 \cdot 10^{-4}$	$4.55 \cdot 10^{-4}$	$2.82 \cdot 10^{-4}$	30000
0.2	$1.09 \cdot 10^{-4}$	$2.67 \cdot 10^{-4}$	$3.68 \cdot 10^{-3}$	
0.2	$9.97 \cdot 10^{-5}$	$1.28 \cdot 10^{-4}$	$1.07 \cdot 10^{-4}$	
0.2	$1.04 \cdot 10^{-4}$	$4.80 \cdot 10^{-4}$	$2.61 \cdot 10^{-4}$	40000
0.2	$1.02 \cdot 10^{-4}$	$4.30 \cdot 10^{-4}$	$2.57 \cdot 10^{-5}$	
0.2	$1.00 \cdot 10^{-4}$	$1.36 \cdot 10^{-4}$	$8.93 \cdot 10^{-5}$	
0.2	$1.02 \cdot 10^{-4}$	$5.32 \cdot 10^{-4}$	$2.63 \cdot 10^{-4}$	100000
0.2	$1.03 \cdot 10^{-4}$	$4.82 \cdot 10^{-4}$	$2.70 \cdot 10^{-4}$	
0.2	$1.05 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$9.19 \cdot 10^{-5}$	

[a] For abbreviations see Table 4

curves are the inputs of the network and the kinetic parameters of the electrically generated Joule effects are used as targets values. All the results concerning this part are not yet exploited. On the other hand, the targets can directely be the powers values of the thermoanalytical curves, so that no reconstruction of the curve, using a given kinetic model, is necessary.

Conclusion

Thermodynamic and kinetic parameters may be computed from calorimetric data by using neural networks. In this work, the optimization of the network topology has been made, as well as the selection of the best learning function. The well known backpropagation algorithm have led to very high values of the error compared to the resilient-propagation algorithm, because of the existence of local minima. Furthermore, neural networks are highly tolerant to the presence of noisy signals. The results obtained in this study in which a high decrease of the relative errors has been found in all cases and especially in the case of noisy signals, in comparison with the usual methods, confirm this finding. This signifies that neural networks may be used for the determination of thermodynamic and kinetic parameters from the analysis of calorimetric data, as well as for the filtering of calorimetric signals.

Thermal effects with known thermodynamic and kinetic parameters, have been generated electrically, using a PC computer, an electronic interface and an inconel thread. Then, these effects have been measured by the apparatus and the difference between the input signal and the measured one have been shown for various scan-

ning rates. These first results are very promising and encourage us in the use of neural networks for the development of our deconvolution method applicable to the study of materials being transformed. These measured signals will be used for the training of the network, while the known parameters of the curve (or the power values) will be set as «targets» values. In this way, the network will be able to account for the transfer function of the apparatus (deconvolution) and for the noise (filtering). In a more simple way, neural networks can be used for calibration of the power and of the temperature of any thermal analysis apparatus, using simulated thermal effects.

References

1 K. Hornik, M. Stinchcombe and H. White, Neural Networks, 2 (1989) 359.
2 B. Irie and S. Miyake, In IEEE Second Int. Conf. Neural Networks, Vol.1, San Diego 1988, p. 641.
3 N. Sbirrazzuoli, D. Brunel and L. Elégant, J. Thermal Anal., 38 (1992) 1509.
4 N. Sbirrazzuoli and D. Brunel, Neural Comput. and Applic., 5 (1997) in press.
5 E. Cesari, P. C. Gravelle, J. Gutenbaum, J. Hatt, J. Navarro, J. L. Petit, R. Point, V. Torra, E. Utzig and W. Zielenkiewicz, J. Thermal Anal., 20 (1981) 47.
6 T. J. McAvoy, H. T. Su, N. S. Wang, M. He, J. Norwath and H. Semerjian, Biotech. Bioeng., 40 (1992) 53.
7 A. P. De Weijer, C. B. Lucasius, G. Kateman, H. M. Heuvel and H. Mannee, Anal. Chem., 66 (1994) 23.
8 S. R. Gallant, S. P. Fraleigh and S. M. Cramer, Chemometrics and Intelligent Laboratory Systems, 18 (1993) 41.
9 P. M. J. Coenegracht, H. J. Metting, E. M. van Loo, G. J. Snoeijer and D. A. Doornbos, J. Chromatogr., 631 (1993) 145.
10 T. Fechner, Third International Conference on Artificial Neural Networks, London, IEEE, 1993, p. 143–147.
11 H. Leung and S. Haykin, Neural Computation, 5 (1993) 928.
12 L. Ying, J. Astola and Y. Neuvo, IEEE Transactions on Signal Processing, 41 (1993) 1201.
13 N. Sbirrazzuoli, Y. Girault and L. Elégant, Thermochim. Acta, 260 (1995) 147.
14 SNNS: Stuttgart Neural Network Simulator, University of Stuttgart, Institute for parallel and distributed high performance systems (IPVR), release 3.1.
15 N. Sbirrazzuoli, Thermochim. Acta, 273 (1996) 169.
16 N. Sbirrazzuoli, Y. Girault and L. Elégant, Thermochim. Acta, 249 (1995) 179.
17 M. Riedmiller and H. Braun, Proc. IEEE International of Neural Networks, Vol. 1, San Francisco 1993, p. 586-591.
18 M. Riedmiller, Computer Standards and Interfaces, 16 (1994) 265.
19 W. Shiffman, M. Joost and R. Werner, Proc. European symposium on artificial neural networks ESANN 93, Brussels, Verleysen (Ed.), 1993, p. 97-104.
20 H. Ishiwatari, J. Kammruzzaman and Y. Kumagai, Proc. 35th Midwest Symposium on Circuits and Systems, New York, Vol. 2, IEEE, 1992, p. 875-878.