NEURAL NETWORK INVERSION OF THE TARASOV FUNCTION USED FOR THE COMPUTATION OF POLYMER HEAT CAPACITIES*

D. W. Noid[1,2], *M. Varma-Nair*[1,2], *B. Wunderlich*[1,2] *and J. A. Darsey*[3]

[1]CHEMISTRY DIVISION, OAK RIDGE NATIONAL LABORATORY, OAK RIDGE, TN 37831–6182, U.S.A.
[2]DEPARTMENT OF CHEMISTRY, UNIVERSITY OF TENNESSEE, KNOXVILLE, TN 37996–1600, U.S.A.
[3]DEPARTMENT OF CHEMISTRY, UNIVERSITY OF ARKANSAS, LITTLE ROCK, AR 72204, U.S.A.

(Received August 12, 1991)

The neural network method is trained using back propagation with a series of heat capacities (C_V) in the temperature range 10 to 200 K for the skeletal vibration of 36 linear macromolecules. The trained network could then accurately extract both parameters of the Tarasov function (Θ_1 and Θ_3) from heat capacities of three computed test cases and a set of experimental measurements for polyethylene. The neural network method offers a major improvement in handling heat capacity data.

I. Introduction

Heat capacities of macromolecules in the solid state have been characterized in a variety of ways. One of the most widely used approximate methods is the Tarasov analysis. In this approach, a combination of one- and three-dimensional Debye functions is chosen to model the skeletal heat capacities of a linear macromolecule and compute the temperature dependence of C_V [1]. Since the Debye functions are not available in closed form, an

* Research sponsored by the Division of Materials Sciences, Office of Basic Energy Sciences, U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. and the Materials Research Division of the National Science Foundation, Polymers Program Grant # DMR-88-18412

inversion of heat capacities to the approximate vibrational spectrum can only be done by trial and error [2]. These techniques to extract the parameters in the Tarasov equation from experimental data are tedious and suffer from accuracy problems. In this paper we show that a neutral network can learn the mathematical transformation from C_v to the two parameters in the Tarasov equation, the temperatures or frequencies Θ_1 and Θ_3.

Previously we have used the neural networks to learn both molecular dynamics [3] and quantum mechanics [4] for some interesting examples. These are some of the first examples of application of neural net extrapolations to complex problems in physical chemistry. In the next section, we describe briefly the neural network approach along with the necessary background for the modeling of heat capacities of solid polymers. The results of our calculation are presented in Section III.

II. A. Heat capacities

The historical perspective of heat capacities for idealized models is well known and will not be discussed here [5]. For macromolecules it has been found that group vibrations contribute little to low temperature heat capacities(C_v) and are well approximated using a summation of Einstein functions [6]. For the skeletal vibrations, it has been found that they can be approximated by a combination of Debye functions. The one-, two-, and three-dimension Debye functions D_1, D_2 and D_3 in units of NR (where R is the gas constant and N is the number of vibrators) are given by

$$D_1(\Theta_1/T) = \left(\frac{T}{\Theta_1}\right) \int_0^{\Theta_1/T} \frac{(\Theta/T)^2 \cdot e^{(\Theta/T)}}{(e^{(\Theta/T)} - 1)^2}\, d(\Theta/T)\,, \tag{1}$$

$$D_2(\Theta_2/T) = 2\left(\frac{T}{\Theta_2}\right)^2 \int_0^{\Theta_2/T} \frac{(\Theta/T)^3 \cdot e^{(\Theta/T)}}{(e^{(\Theta/T)} - 1)^2}\, d(\Theta/T)\,, \tag{2}$$

and

$$D_3(\Theta_3/T) = 3\left(\frac{T}{\Theta_3}\right)^3 \int_0^{\Theta_3/T} \frac{(\Theta/T)^4 \cdot e^{(\Theta/T)}}{(e^{(\Theta/T)} - 1)^2}\, d(\Theta/T)\,, \tag{3}$$

respectively. $D_1(\Theta_1/T)$ is based on a constant frequency distribution with frequency, $D_2(\Theta_2/T)$ on a linear distribution, and D_3 (Θ_3/T) on a quadratic distribution. The parameters Θ_1, Θ_2, and Θ_3 are the characteristic upper frequencies for these approximations to the density of vibration states and represent $h\nu/k$, where h is Planck's constant and k is Boltzmann's constant $(1\,k = 0.695\text{ cm}^{-1})$.

Using a combination of Eqs (1) and (3) to approximate the heat capacities of macromolecules, Tarasov [1] has proposed the form

$$C_v(T)/NR = D_1(\Theta_1/T) - \frac{\Theta_3}{\Theta_1}\cdot\left[D_1(\Theta_3/T) - D_3(\Theta_3/T)\right]$$

to model the functional form of $C_v(T)$. Wunderlich *et al.* have successfully used this approach for a variety of polymer systems [2, 7]. However, it turns out that the inversion of C_v *vs.* T for Θ_1 and Θ_3 is not trivial and requires a complex procedure. In this paper, a neural network described in the next section is used for the inverse transformation.

II. B. Neural networks

Artificial neural networks (ANNs) are computational tools. They can be thought of as mathematical functions with generalized estimation and prediction capabilities, and as a scheme for complex computations that can be distributed among several processors [8]. This technique can be used to extend the data gained by simulation to longer times, extract parameter dependency, and even analyze the results of simulation to extract various functional properties. Whereas in classical approximation techniques such as parametric regression or spline smoothing the main problem involves the determination of a set of parameters and/or functions, ANNs require the computation of a set of weights associated with the connections of simple processing elements, called neurons.

Different types of networks are categorized by the manner that weights are computed – called the training or learning algorithm – and the manner in which neurons are arranged – called the architecture of the network. A typical architecture has an input layer of neurons, one or more middle or 'hidden' layers, and an output layer.

The ANNs considered in this research are given a set of known-to-be-correct input/output pairs, called examples, and their weights are adjusted in order to best represent these examples through the back propagation learning algorithm [9]. The name back propagation is due to the procedure for updating the weights during training in which, when the network corrects its internal parameters, the correction mechanisms start with the output units and propagate backwards through each hidden layer to the input layer.

Each neuron of an ANN constitutes a processing element (PE), and it is connected through various weights, w_{ij}, to other PEs. The processing element sums the product of each input and the connection weights from the previous layer of PEs and then filters it by a nonlinear thresholding function of the form

$$f(x)=\frac{1}{1+e^{-\beta x}},\quad \beta>0, \tag{5}$$

In general, for a multilayer network, the output of the j th PE in layer p is given by

$$x_j^p=f\left(\sum_i^p w_{ij}^{p-1}\cdot x_i^p+\Theta_j^p\right), \tag{6}$$

where the thresholding function f has the sigmoidal form of Eq. (5) and θ_j^p is the bias associated with the j th PE of layer p. The weights w_{ij} are initialized randomly and then adjusted during training. Training the network means establishing the values of these weights between pairs of processing elements. The local error at each node is computed and the connection weights are adjusted to minimize the global error through a gradient descent algorithm. The computational burden in the application of multilayered ANNs lies primarily in network training. The weights of the network are fixed after training is completed. These values are then used during 'recall' sessions, i.e., when an unknown input is presented to the network and it computes the appropriate output. The computations required in the recall sessions are negligible; i.e. larger training sessions can be carried out on supercomputers and, once the weights are fixed, they can be transferred for use on personal computers for quick data analysis.

III. Calculation and results

Our neural network procedure using the back propagation method is incorporated in the NNETS program [9]. We have installed the floating point version of this program [10] on a CRAY-XMP at ORNL. Our 4-layer network consists of an input layer containing $C_v(T)$ with 15 nodes. Two hidden layers contained 5 nodes each. A 2-node output layer contained the value of Θ_1 and

Θ_3. Thirty-six training examples were considered, and the network learned the relationship between $C_v(T)$ and Θ_1, Θ_3 to an average error of less than 1 K with a maximum error of 2.5 K. The ranges chosen for Θ_1 and Θ_3 were 440 to 540 K and 100 to 200 K, respectively, in steps of 20 K. The CRAY required ≈20 minutes for the ≈30.000 cycles of the learning process. In the recall process, an insignificant amount of CPU time is needed. A major advantage to this method is that a very large number of inversions can be performed on even an IBM PC as the weights generated are transportable. In Table 1, the predicted results are presented for three theoretical examples [C_v *vs.* T generated from Eq. (4) without added noise] and an experimental determination of the heat capacity for polyethylene (with the group vibration substracted out) [11]. In each of these cases, the errors are on the order of 1 to 2 K, as expected from the errors computed from the training data.

Table 1 Neural network predictions

Predicted values by the neural network / K		Input C_v computed from the given Θ-values / K*	
Θ_1,	Θ_3	Θ_1	Θ_3
450.4	148.6	450	150
529.7	191.0	530	190
489.4	108.5	490	110
518.8	156.9	519±10**	158±2**

*C_v generated from Eq. (4).

**Polyethylene data, Θ_1, Θ_3 previously found using the inversion described in Refs 2 and 11.

IV. Conclusion

We have presented a new method for extracting the Θ_1 and Θ_3 parameters in the Tarasov equation for modeling the temperature dependence of C_v for solid macromolecules. This technique was demonstrated to lead to an accuracy of ≈1 K, which is a considerable improvement over prior methods. Since other network configurations may produce even better results, we are presently working to optimize the method. Also, information contained in the weights can be analyzed to determine what features of C_v *vs.* T are most closely related to Θ_1 and Θ_3. In a future, more detailed report, we hope to address some these topics and produce a set of weights that permits easy access to this otherwise involved inversion.

* * *

We acknowledge helpful discussions with Dr. B. G. Sumpter on neural networks and Dr. A. Xenopoulos on heat capacities.

References

1 V. V. Tarasov, Zh. Fis. Khim., 24 (1950) 111; 27 (1953) 1430; 39 (1965) 2077; and Dokl. Acad. Nauk SSSR, 100 (1955) 307.

2 Yu.. V. Cheban, S. F. Lau and B. Wunderlich, Coll. Polym. Sci., 260 (1982) 9.

3 D. W. Noid and J. A. Darsey, Comp. Polym. Sci., 1 (1991) 157.

4 J. A. Darsey, D. W. Noid and B. R. Upadhyaya, Chem. Phys. Lett., 177 (1991) 189.

5 C. E. Hecht, Statistical Thermodynamics and Kinetic Theory, Ch. 4, W. H. Freeman and Co., 1990; M. Blackman, Enc. of Physics, Ed. S. Flügge, Vol. III, Part I, Springer, Berlin 1955.

6 B. Wunderlich and H. Baur, Adv. Polym. Sci., 7 (1970) 151.

7 H. S. Bu, S. Z. D. Cheng and B. Wunderlich, J. Phys. Chem., 91 (1987) 4179.

8 For a comprehensive discussion of neural networks, see R. Hecht-Nielson, Neurocomputing Addison-Wesley, New York 1990.

9 D. Rumelhart and J. McClelland, Parallel Distributed Processing, Vol. 1, Bradford Books/MIT Press, Cambridge MA 1986.

10 P. T. Baffes, NNETS Program, Version 2.01, Johnson Space Center Report No. 23366, September 1989.

11 J. Grebowicz, H. Suzuki and B. Wunderlich, Polymer, 26 (1985) 561.

Zusammenfassung — Die Neural-Network Methode wird zur Bestimmung der Wärme kapacität (C_V) von 36 linearen Makromolekül-Gerüstschwingungen im Temperaturbereich 10–200 K angewendet. Das präparierte System ist dann in der Lage, bei drei rechnerischen Testfällen und einer Reihe von experimentellen Messungen an Polyethylen aus den Wärmekapazitäten beide Parameter der Tarasov-Funktion (Θ_1 und Θ_2) genau zu ermitteln. Die Neural-Network Methode stellt eine wesentliche Verbesserung zur Handhabung von Wärmekapazitätsangaben dar.