Int. J. Heat Mass Transfer. Vol. 34, No. 8, pp. 2063–2070, 1991
Printed in Great Britain

0017-9310/91 $3.00+0.00

A neural network methodology for heat transfer data analysis

JULES THIBAULT and BERNARD P. A. GRANDJEAN
Department of Chemical Engineering, Laval University, Sainte-Foy, Quebec, Canada G1K 7P4

(*Received* 4 *May* 1990 *and in final form* 1 *October* 1990)

Abstract—Neural networks have been until very recently a topic of academic research. Recent developments of powerful learning algorithms and the increasing number of applications in a great number of disciplines suggest that neural networks can provide useful tools for modelling and correlating practical heat transfer problems. This paper presents an introduction to computing with neural networks. To evaluate the potential of neural networks for correlating heat transfer data, three different examples are solved, using a three-layer feedforward neural network. Two different learning algorithms, including the traditional backpropagation algorithm, are used to teach the neural network. It is shown that neural networks can be used to adequately correlate heat transfer data.

INTRODUCTION

MODELLING of heat transfer phenomena has always been a major preoccupation of scientists. It is usually desired to represent the experimental data with the most compact equation or set of equations. This can be performed analytically by solving the appropriate differential equations or empirically by using traditional regression analysis. The analytical approach is often inapplicable because of the difficulties in obtaining or solving the relevant set of partial differential equations. Heat transfer literature abounds of heuristic relationships attempting to fit experimental data to a system model for the purpose of representing the essential aspects of the system and of presenting the knowledge in a usable form.

In many important applications, observable inputs are difficult to describe analytically. The best or even a good structure for the model cannot be determined in advance. Therefore, it is highly desirable to have a model that is trainable in both its structure and parameter values [1]. An alternative to structured modelling is the use of neural networks which allows models of complex systems to be built without requiring the explicit formulation of the possible relationships that may exist between variables. The idea is to build a network that is able to learn, through representative examples, to associate, to a vector of input variables, a vector of output variables. Artificial neural networks, coupled with an appropriate learning algorithm, can be used to learn complex relationships from a set of associated input–output vectors. In essence, a neural network can simply be viewed as a large dimensional regression model which can be used to model heat transfer phenomena. A more detailed description on neural networks is presented in the next section.

This paper provides an introduction to the field of neural networks applied to heat transfer problems. The nomenclature and characteristics of neural networks are presented and their application for correlating heat transfer data is demonstrated with three examples.

NEURAL NETWORKS

The desire of human beings to build artificial systems that do the kinds of interesting things that they do has long existed. Current research on neurocomputers, which has for its objective the birth of a system with some human abilities such as learning and reasoning, is a good example [2]. Neural networks, as they are known today, originate from the work of McCulloch and Pitts [3] who demonstrated, in 1943, the ability of interconnected neurons to calculate some logical functions. Hebb [4], in 1949, pointed out the importance of the synaptic connections in the learning process. Later, Rosenblatt [5], based on the work of his predecessors, presented the first operational model of a neural network: the 'perceptron'. The perceptron, built as an analogy to the visual system, was able to learn some logical functions by modifying the synaptic connections. The perceptron gave rise to tremendous excitement and research in this field until Minsky and Papert, in 1969, demonstrated the severe limitations of the perceptron [6]. Their publication, as a consequence, almost brought to a halt the important research that was performed at that time. It was not until 1982, when the work published by Hopfield [7] led to a resurgence of interest in neural networks. Today, research in artificial neural networks is being performed in a great number of disciplines, ranging from neurobiology and psychology to engineering sciences. It is not possible, yet, to have access to a true neural computer. However, it is possible to use the methodology of neural networks on sequential digital or parallel computers to solve engineering problems.

NOMENCLATURE

EMF	electromotive force [mV]
F	objective function
f	non-linear transfer function
f'	first derivative of the linear transfer function
Gr	Grashof number
H	calculated output of a neuron in the hidden layer
I	number of neurons in the input layer, including the bias
J	number of neurons in the hidden layer, including the bias
K	number of neurons in the output layer
n	exponent of power law variation of surface heat flux
Nu	Nusselt number
Pr	Prandtl number
Ra	Rayleigh number
S	calculated output of a neuron in the output layer
T	temperature [°C]
U	input vector to the network
W	weight matrices
Y	scaled target output vector.

Greek symbols

α	intermediate parameter in equation (9b)
β	fraction of the previous weight correction
ε	learning rate in the gradient descent
Ω	surface curvature.

Subscripts

i	input layer
j	hidden layer
k	output layer
L	average along the length of the cylinder
x	axial coordinate along the cylinder.

Superscript

m	iteration counter.

Artificial neural networks are massively parallel, distributed and adaptive systems, modelled on the general features of biological networks with the potential for ever-improving performance through a dynamical learning process [8]. Neural networks are made up of a great number of individual processing elements, the neurons, which perform simple tasks. A neuron, schematically represented in Fig. 1, is the basic building block of neural network technology which performs a non-linear transformation of the weighted sum of the incoming inputs to produce the output of the neuron. Inputs and outputs of the network are normally numeric values scaled between 0 and 1. The input to a neuron can come from other neurons or from outside the network. The non-linear transfer function can be a threshold, a sigmoid, a sine or a hyperbolic tangent function.

Neural networks are comprised of a great number of interconnected neurons. There exists a wide range of network architectures. The choice of the architecture depends on the task to be performed. For the modelling of physical systems, a feedforward layered network is usually used. It consists of a layer of input neurons, a layer of output neurons and one or more intermediate or hidden layers. In this investigation, a three-layer feedforward network was used, that is a network with a single hidden layer as shown in Fig. 2. 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 [9, 10]. The following description will therefore be restricted to a three-layer network architecture. Using the nomenclature of Fig. 2, the details of the calculation using a three-layer neural network are summarized by the following set of equations:

Hidden layer

$$H_j = f\left[\sum_{i=1}^{I} W_{ij} U_i\right] \quad 1 \leqslant j \leqslant J-1 \qquad (1a)$$

where U is the scaled input vector and H the output vector of the neurons contained in the hidden layer.

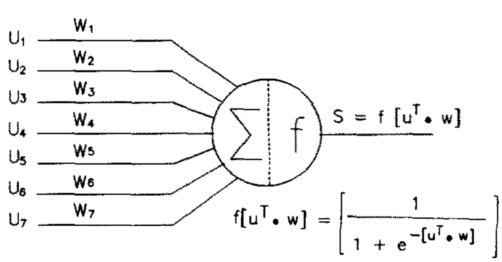

FIG. 1. A simple processing neuron.

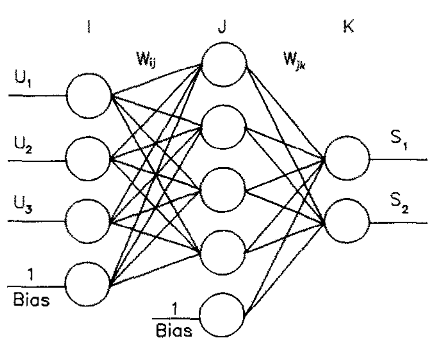

FIG. 2. Typical architecture of a three-layer neural network.

The last elements of these two vectors, U_I and H_J, are the bias and they are set equal to 1.

Output layer

$$S_k = f\left[\sum_{j=1}^{J} W_{jk} H_j\right] \quad 1 \leqslant k \leqslant K. \tag{1b}$$

The extension to a multi-layer neural network is straightforward.

LEARNING ALGORITHMS

In a neural network, the knowledge lies in the interconnection weights between neurons and the topology of the net [11]. Therefore, one important aspect of a neural network is the learning process whereby representative examples of the knowledge to be acquired are presented to the network so that it can integrate this knowledge within its structure. Learning implies that the processing element somehow changes its input/output behaviour in response to the environment. The learning process thereby consists in determining the weight matrices, W_{ij} and W_{jk}, that produce the best fit of the predicted outputs over the entire training data set. The basic procedure is to first set the weights between adjacent layers to random values. An input vector is then impressed on the input layer and it is propagated through the network to the output layer. The difference between the computed output vector of the network and the scaled target output vector is then used to adapt the weight matrices using an iterative optimization technique in order to progressively minimize the sum of squares of the errors. The process is repeated over the entire training set, a great number of times, to achieve the desired degree of accuracy.

1. *Backpropagation*

The most versatile learning algorithm for this feedforward layered network is backpropagation in which a quadratic cost function is minimized by a modified gradient descent [11–15]. As the name of this algorithm suggests, it simply consists of redistributing or backpropagating the output errors to the network by appropriately modifying the weight matrices. The backpropagation algorithm requires the transfer function to be differentiable [16] and it is restricted to feedforward neural networks.

Like all gradient descent algorithms, backpropagation is not guaranteed to actually find the set of weights that corresponds to the global minimum of the sum of squares of the errors. Indeed, it may easily get stuck in a local minimum [17]. However, it is important to point out that a different set of weights can map equally successfully without being unique. In addition, as the size of the network is increased, the backpropagation algorithm becomes less efficient and it has greater difficulty to converge to a proper solution [18]. Another important drawback of backpropagation is that it generally requires a great number of iterations. Indeed, it is necessary to present all examples of the learning set many thousand times. However, once an appropriate set of weights has been obtained, the use of the neural model to recall stored information is straightforward.

Two different strategies are commonly used to update the network weight matrices. In the first approach, the corrections of weights for each example are summed over the entire training data set before updating the weights (equations (4) and (5)). In the second approach, changes in weights are performed after the presentation of each example. These two strategies are referred to respectively as periodic updating and continuous updating [19]. The second strategy is usually preferred since it normally leads to faster learning. In this investigation, the error was corrected after each presentation of an example.

At the end of each example, it is desired to minimize the following sum of squares of the prediction errors over all output neurons:

$$F(W) = \tfrac{1}{2} \sum_{k=1}^{K} (Y_k - S_k)^2 \tag{2}$$

which is used to modify each weight of the two matrices as follows:

$$W_{ij}^{m} = W_{ij}^{m-1} - \varepsilon^{m} \left[\frac{\partial F(W)}{\partial W_{ij}}\right]_{m-1} + \beta^{m}[W_{ij}^{m-1} - W_{ij}^{m-2}] \tag{3}$$

where m is an iteration counter. The second term represents the fraction of the error gradient that is backpropagated through the network. ε^m is the learning rate which provides the step size during gradient descent. The last term is called the momentum term which forces the change of weight to proceed in the same direction as the previous change. It is claimed that the introduction of momentum speeds up the convergence of the algorithm and allows the escape from narrow minima [20].

The error gradient $(\partial F(W)/\partial W_{ij})$ for each processing element on the output layer, used to correct the weights between the hidden layer and the output layer is given by

$$\frac{\partial F(W)}{\partial W_{jk}} = f'\left[\sum_{j=1}^{J} W_{jk} H_j\right](S_k - Y_k) H_j \tag{4}$$

and for the weights from the input layer to the hidden layer

$$\frac{\partial F(W)}{\partial W_{ij}} = f'\left(\sum_{i=1}^{I} W_{ij} U_i\right) \times \left[\sum_{k=1}^{K} f'\left(\sum_{j=1}^{J} W_{jk} H_j\right)(S_k - Y_k) W_{jk}^{m-1}\right] U_i \tag{5}$$

where indexes and variables in Fig. 2 have been used. $f'(\cdot)$ represents the derivative of the non-linear transfer function. In the present investigation, a sigmoidal function (Fig. 1) was used and its derivative is given by

$$f'(z) = f(z)(1 - f(z)). \tag{6}$$

For more details on neural networks, the reader is referred to the paper of Lippman [21] who published a clear and well-written introductory article on computing with neural networks, including the backpropagation algorithm.

2. *Quasi-Newton learning algorithm*

One of the most successful methods for calculating the minimum of a function $F(W)$ of several variables, when the first derivatives are available, is the popular BFGS algorithm [22, 23]. This method is superior to the popular steepest descent method because it takes implicitly into account second derivatives. The acronym BFGS stands for Broyden, Fletcher, Goldfarb and Shanno who contributed significantly to this method. It is now a commercial subroutine: Harwell, VA13A. Powell [22] has shown that its rate of convergence is superlinear.

This method simply requires the numerical values of the objective function $F(W)$, that is the sum of squares of residuals (equation (2)), and the vector of first derivatives evaluated for the current set of weights. The vector of first derivatives was evaluated with equations (4) and (5) but could also be evaluated with the finite difference approximation.

RESULTS

To demonstrate the methodology of neural networks for correlating heat transfer data, three examples were chosen: a thermocouple lookup table [24], a series of correlations between Nusselt and Rayleigh numbers for the free convection around horizontal smooth cylinders [25] and the problem of natural convection along slender vertical cylinders with variable surface heat flux [26]. Each of these examples has been chosen to cover some important characteristics of neural networks. The results for each example are presented and discussed in turn.

1. *Thermocouple*

It is desired to derive a model that will give the temperature corresponding to the electromotive force (EMF) generated by a chromel–alumel (type K) thermocouple [24]. This is the simplest example, that one can possibly have, to use a neural network and, it is undoubtedly an overkill for this problem. However, it was purposely chosen as simple as possible to show some important characteristics of neural networks.

The first thing that must be accomplished when using a neural network is to determine the architecture of the network. For a three-layer neural network used in this investigation, the number of neurons in the input and output layers are fixed when the problem is set. For this example, with one input and one output, the number of neurons in the input and output layers are respectively 2 (including the bias) and 1. The problem of selecting the architecture of the network, therefore boils down to selecting the number of neurons in the hidden layer. Unfortunately, there does not exist yet a procedure to determine a priori the number of neurons in a hidden layer that will give the desired degree of accuracy. This information must be obtained experimentally. Figure 3 presents the variations of the sum of squares of the prediction errors as a function of the number of neurons in the hidden layer, obtained with the two weight correction methods, tested in this investigation. The maximum number of iterations was 5000. Also shown in Fig. 3, is the sum of squares of the errors for a polynomial regression analysis as a function of the order of the polynomial equation. The sum of the squares of the errors are based on 201 temperature values in the range of 0–200°C.

Out of the two methods used in this investigation, the quasi-Newton method is by far the most reliable and the most accurate optimization method. It is more accurate by a factor of two orders of magnitude than the traditional backpropagation. Theoretically, the backpropagation method can lead to similar results provided the appropriate set of initial random weights is given. It can more easily get stuck into a local minimum than the quasi-Newton method which uses implicitly the second derivatives to correct the weight connections between neurons in adjacent layers. The downhill simplex algorithm [23], due to Nelder and Mead [27], has also been tested and the modelling accuracy obtained with this method was of the same order of magnitude as the backpropagation method.

Backpropagation is the standardly used learning algorithm with neural networks. The main reason for this widespread use is that it can be used safely for systems with binary outputs (image processing, fault detection, contour detection, etc.). Its main disadvantage is the slow convergence rate. The tuning parameters, β^m and ε^m, of equation (3) can sometimes be used to speed up convergence and to escape from narrow minima. However, there are no available techniques to choose these parameters. These parameters

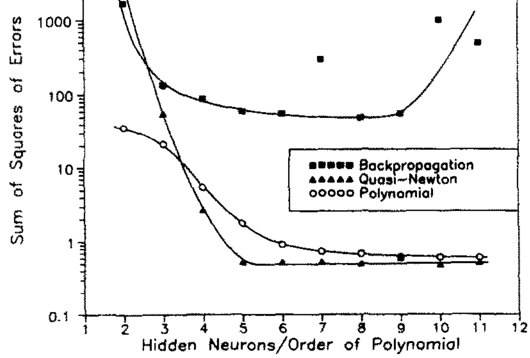

FIG. 3. Variation of the sum of the squares of the temperature errors as a function of the number of hidden neurons or order of the polynomial.

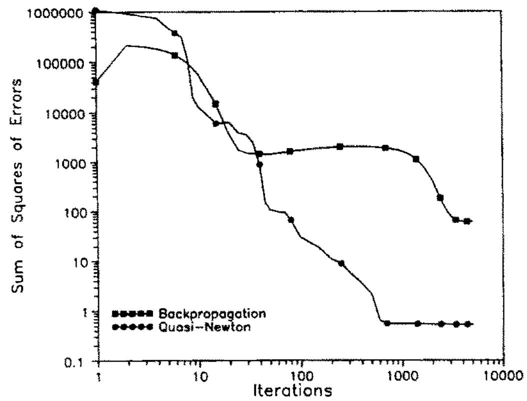

FIG. 4. Variation of the sum of the squares of the temperature errors as a function of the number of iterations.

are not necessarily kept constant but may be changed throughout the learning process [14, 28]. Different values were tested in this investigation without significant improvement. In this investigation, all results presented using the backpropagation algorithm were obtained with $\varepsilon^m = 0.95$ and $\beta^m = 0$.

The quasi-Newton method can be used for systems with continuous outputs and normally not for systems with binary outputs since the first derivatives of the objective function with respect to the weights may not be continuous. However, for continuous systems, this method is extremely efficient as compared with the other two methods. Indeed, a neural model containing five hidden neurons is slightly more accurate than a tenth-order polynomial to fit the EMF-temperature relationship. A feedforward neural network containing five hidden neurons has 13 parameters (weight connections) to evaluate whereas the tenth-order polynomial contains 11.

In addition to the better accuracy obtained with the quasi-Newton method, the number of iterations required to converge to a solution is significantly smaller than for the backpropagation method as observed in Fig. 4 for a neural network containing five hidden neurons, including the bias. Because the quasi-Newton method is more accurate and its speed of convergence is superior to the other two methods, it was used throughout this investigation.

2. *Free convection for horizontal smooth cylinders*

The second example considers the correlation between Nusselt and Rayleigh numbers for horizontal smooth cylinders. Morgan [25] proposed the following set of equations to describe the natural convection for smooth horizontal cylinders:

$$Nu = 0.675Ra^{0.058} \quad 10^{-10} \leqslant Ra \leqslant 10^{-2} \quad (7a)$$

$$Nu = 1.02Ra^{0.148} \quad 10^{-2} \leqslant Ra \leqslant 10^{2} \quad (7b)$$

$$Nu = 0.850Ra^{0.188} \quad 10^{2} \leqslant Ra \leqslant 10^{4} \quad (7c)$$

$$Nu = 0.480Ra^{0.250} \quad 10^{4} \leqslant Ra \leqslant 10^{7} \quad (7d)$$

$$Nu = 0.125Ra^{0.333} \quad 10^{7} \leqslant Ra \leqslant 10^{12} \quad (7e)$$

Table 1. Input/output information of neural networks used in this investigation

Thermocouple	$I = 2$	$U_1 = \mathrm{EMF}$ (mV)
		$U_2 = 1$
	$K = 1$	$S_2 = T$ (°C)
Free convection from horizontal cylinders	$I = 2$	$U_1 = \log_{10}(Ra)$
		$U_2 = 1$
	$K = 1$	$S_1 = \log_{10}(Nu)$
Natural convection along slender vertical cylinders	$I = 4$	$U_1 = \log_{10}(Pr)$
		$U_2 = n$
		$U_3 = \Omega$
		$U_4 = 1$
	$K = 2$	$S_1 = Nu_L\, Gr_L^{-1/5}$
		$S_2 = Nu_x\, Gr_x^{-1/5}$

To properly determine the relationship between the Nusselt and the Rayleigh numbers, over a wide range of Rayleigh numbers, it was necessary to separate this range into five segments. The objective of this section is to investigate the possibility of replacing the five equations with a unique neural model.

The architecture of the neural model is identical to the previous example except that it is preferable to perform a logarithmic transformation of the input and output variables (Table 1) in order to obtain a good accuracy over the full range of Rayleigh numbers. Figure 5 presents the plot of *Nu* as a function of *Ra* for the prediction of the data (without added noise) generated with the set of five equations using a neural network containing five hidden neurons. The neural model gives a very accurate representation of the generated data over the full range of Rayleigh numbers. It is clear that a unique neural model can adequately replace the set of five equations. In addition, the neural model is able to model the discontinuity at a Rayleigh number of 10^{-2}. The corresponding neural model, with five hidden neurons, is given by the following set of equations:

$$U_1 = \left[\frac{\log_{10} Ra - U_{1,\min}}{U_{1,\max} - U_{1,\min}}\right] \quad \text{with} \quad U_{1,\min} = -10, \quad U_{1,\max} = 12 \quad (8a)$$

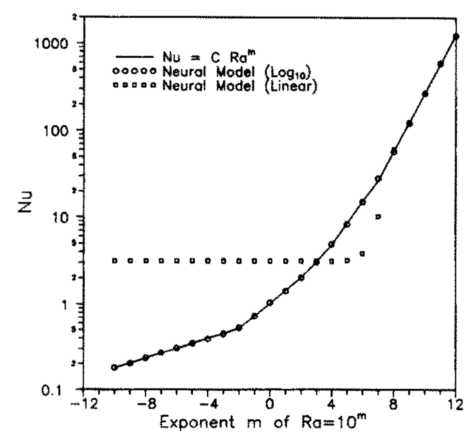

FIG. 5. Plot of the Nusselt number as a function of Rayleigh number for the free convection around smooth horizontal cylinders.

$$\begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{bmatrix} = f\left[\begin{bmatrix} 0.267 & 1.858 \\ 10.332 & -3.041 \\ -9.134 & 16.703 \\ 70.725 & 28.616 \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ 1 \end{bmatrix}\right] \quad (8b)$$

$$[S_1] = f\left[[157.50 \quad -0.9527 \quad -2616.0 \quad 0.1831 \quad 2476.9] \cdot \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ 1 \end{bmatrix} \right] \quad (8c)$$

$$\log_{10}(Nu) = S_{1,\min} + S_1[S_{1,\max} - S_{1,\min}]$$
$$\text{with} \quad S_{1,\min} = -1.0, \quad S_{1,\max} = 3.5. \quad (8d)$$

A neural network is not a magic box where information can be entered in any fashion. In the present example, it is important to perform a logarithmic transformation of the input and the output vectors to find an appropriate relationship between the two variables. Failure to do so, results in a very inaccurate neural model in the lower range of Rayleigh numbers whereas the accuracy in the upper range is preserved as evidenced by the curve in Fig. 5 for the neural model using a linear relationship between Rayleigh and Nusselt numbers. Since the lower range of Rayleigh numbers is assimilated to a unique value of zero with respect to the upper range, a corresponding Nusselt number of 3.14, which minimized the sum of the squares of the errors, is obtained.

3. *Natural convection along slender vertical cylinders*

The third example is concerned with the use of a neural network for correlating the average and the local Nusselt numbers as a function of the Prandtl number (Pr), the exponent of the power law variation of the surface heat flux (n) and the surface curvature (Ω) to describe the natural convection along slender vertical cylinders with variable heat fluxes. Heckel *et al.* [26] presented an analysis of this problem and provided tables of values along with relatively complex equations to correlate the average and the local Nusselt numbers.

For the local Nusselt number, the equation is

$$Nu_x\, Gr_x^{*-1/5} = \alpha(Pr)[A(\Omega) + f_1(Pr)\Omega](1 + VW) \quad (9a)$$

where

$$\alpha(Pr) = Pr^{2/5}(4 + 9Pr^{1/2} + 10Pr)^{-1/5} \quad (9b)$$

$$A(\Omega) = 1 + 0.09\Omega^{1/2} \quad (9c)$$

$$f_1(Pr) = (0.032 + 0.176Pr^{-0.384}) \quad (9d)$$

$$V = [[0.328 + 0.343\exp(-2.12Pr^{1/5})] - 0.195n]n \quad (9e)$$

$$W = \exp[-(0.0265 + 0.0907Pr^{-0.444})\Omega^{0.8}] \quad (9f)$$

and for the average Nusselt number for the same ranges of parametric values the equation is given by

$$Nu_L\, Gr_L^{*-1/5} = \tfrac{5}{4}\alpha(Pr)[B(\Omega) + f_2(Pr)\Omega](1 + \bar{V}) \quad (10a)$$

where

$$B(\Omega) = 1 + 0.08\Omega^{1/2} \quad (10b)$$

$$f_2(Pr) = (0.026 + 0.14Pr^{-0.39}) \quad (10c)$$

$$\bar{V} = (4V\bar{W} - n\bar{W})/(4 + n\bar{W}) \quad (10d)$$

$$\bar{W} = \exp(-0.5\Omega^{0.8}). \quad (10e)$$

In addition to the three independent variables (Pr, n and Ω), the highly non-linear correlation equations contain a total of 32 parameters. This example is therefore an ideal problem for the use of neural networks. As shown in Table 1, the neural network architecture for this problem is comprised of four input neurons (including the bias) and two output neurons. For the same reason as for the second example, a logarithmic transformation was performed on the Prandtl number.

The comparison of the correlation equations proposed by Heckel *et al.* [26] and the neural model for the prediction of the local and the average Nusselt number for the 132 values presented respectively in Tables 1 and 3 of their paper is presented in Fig. 6 in terms of prediction errors. All values of the average Nusselt number contained in Table 3 have to be divided by $5^{0.2}$ to get the correct values [29, 30]. To be fair in the comparison, a neural network with six hidden neurons was used. With a six hidden neurons

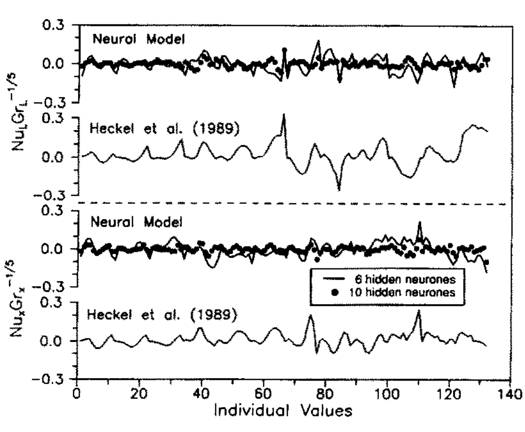

FIG. 6. Plot of the average and the local Nusselt number prediction errors for the natural convection along slender vertical cylinders with variable surface heat flux.

architecture, the total number of weight connections (neural model parameters) is identical to the number of parameters for the two correlation equations. With the average Nusselt number, the average sums of squares of the errors were respectively 1.14 and 0.49 for the correlation equations and the neural model whereas for the local Nusselt number, they were 0.40 and 0.49. These results clearly indicate that a neural network is able to correlate very efficiently complex systems. In this particular example, for similar accuracy, the effort required to obtain an appropriate model to describe the average and local Nusselt numbers is far less for neural networks than the very complex set of equations obtained by Heckel *et al.* [26].

Significantly more accurate neural models are obtained when the number of hidden neurons is increased as evidenced by the dot curves of Fig. 6, representing the prediction errors for a network with ten hidden neurons. With ten neurons, the sums of the squares of the errors are respectively 0.088 and 0.072 for the average and the local Nusselt numbers. Another way to obtain a more accurate prediction would be to separate the prediction of the local and average Nusselt numbers into individual neural models.

CONCLUSION

The results presented in this paper have clearly shown that neural network methodology can be used efficiently to model and correlate heat transfer data. The main advantage of neural networks is to remove the burden of finding an appropriate model structure to fit experimental data or to find a useful regression equation. However, as opposed to most standard mathematical models, the knowledge of a neural model is scattered throughout the network and lies in its inter-neuron connections and their associated weights. It is not possible simply by inspection to determine the influence that one input variable has on an output variable. Neural models have the advantage of not requiring a formal model structure but at the expense of a loss of model transparency.

In the applications described in this paper, the time required to find a good approximation is not important, since a given problem needs to be solved only once, and its solution can be used many times. However, for a system that changes over time, it is possible to use an adaptive correction scheme [31].

The years to come will see a drastic development of neural networks in all areas including all spheres of heat transfer. The potential for correlating heat transfer data has been clearly demonstrated in the present paper. Neural networks will find wider application when 'rules of thumb' for choosing the proper architecture, learning rules and parameter adjustment are developed along with the increased parallelization of computation.

REFERENCES

1. R. L. Barron, Learning networks improve computer-aided prediction and control, *Comput. Des.* 65–70 (August 1975).
2. J. Anderson, Cognitive and psychological computation with neural models, *Trans. Syst. Man Cybernet.* **13**(5), 799–815 (1983).
3. W. S. McCulloch and W. Pitts, A logical calculus in the ideas immanent in nervous activity, *Bull. Math. Biophys.* **5,** 115–133 (1943).
4. D. O. Hebb, *The Organization of Behaviour.* Wiley, New York (1949).
5. F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, *Psychol. Rev.* **68,** 386–408 (1958).
6. M. Minsky and S. Papert, *Perceptrons.* MIT Press, Cambridge, Massachusetts (1969).
7. J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natn. Acad. Sci. USA* **79,** 2554–2558 (1982).
8. B. Bavarian, Introduction to neural networks for intelligent control, *IEEE Control Syst. Mag.* 3–7 (April 1988).
9. K. Hornik, M. Stinchcombe and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* **2,** 359–366 (1989).
10. B. Iric and S. Miyaki, Capabilities of three layer perceptrons. In *IEEE Second Int. Conf. on Neural Networks*, San Diego, Vol. I, pp. 641–648 (1988).
11. W. P. Jones and J. Hoskins, Back-propagation a generalized delta learning rule, *BYTE* **Oct.,** 155–162 (1987).
12. E. Davalo and P. Naim, *Des réseaux de neurones.* Eyrolles, Paris (1989).
13. Y. Owechko, Optoelectronic resonator neural networks, *Appl. Optics* **26**(23), 5104 (1987).
14. D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning representations by back-propagating errors, *Nature* **323,** 533 (1986).
15. P. J. Werbos, Beyond regression: new tools for prediction and analysis in the behavioral sciences, Ph.D. Thesis, Harvard University, Cambridge, Massachusetts (1974).
16. A. Petrowski, L. Personnaz, G. Dreyfus and C. Girault, Parallel implementations of neural network simulations. In *First European Workshop on Hypercube and Distributed Computers.* Elsevier, Amsterdam (1989).
17. M. W. Roth, Neural-network technology and its applications, *Johns Hopkins APL Tech. Dig.* **9,** 3 (1988).
18. J. Barhen, S. Gulati and M. Zaz, Neural learning of constrained nonlinear transformations, *IEEE Comput.* 67–76 (June 1989).
19. S. E. Falham, Faster-learning variations on back-propagation: an empirical study. In *Proc. 1988 Connectionist Models Summer School* (Edited by D. Touretzhy, G. E. Hinton and T. J. Sejnowski), pp. 38–51. Morgan Kaufman, San Mateo, California (1988).
20. K. A. Benedict, Learning in the multilayer perceptron, *J. Phys. A: Math. Gen.* **21,** 2643–2650 (1988).
21. R. P. Lippman, An introduction to computing with neural nets, *IEEE ASSP Mag.* 4 (April 1987).
22. M. J. D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, ASM/SIAM Symp. on Nonlinear Programming, New York (1975).
23. H. W. Press, B. P. Flannery, S. A. Teukosky and W. T. Vetterling, *Numerical Recipes, The Art of Scientific Computing.* Cambridge University Press, London (1988).
24. *Handbook of Chemistry and Physics*, 68th Edn. CRC Press, Boca Raton, Florida (1987).
25. V. T. Morgan, The overall convective heat transfer from smooth circular cylinders. In *Advances in Heat Transfer*, Vol. 11, pp. 199–266. Academic Press, New York (1975).

26. J. J. Heckel, T. S. Chen and B. F. Armaly, Natural convection along slender vertical cylinders with variable surface heat fluxes, *ASME J. Heat Transfer* **111**, 1108–1111 (1989).
27. J. A. Nelder and R. Mead, Simplex method for function minimization, *Comput. J.* **7**, 308–313 (1965).
28. T. J. McAvoy, N. S. Wang, S. Naidu, N. Bhat, J. Gunter and M. Simmons, Interpretation biosensor data via backpropagation, American Control Conf., June (1989).
29. T. S. Chen, Personal communication (March 1990).
30. Errata, *ASME J. Heat Transfer* **112**, 450 (1990).
31. V. Van Breusegem, J. Thibault and A. Chéruy, Adaptive neural models for on-line prediction in fermentation. *Can. J. Chem Engng* (in press).

UNE METHODOLOGIE SUR LE RESEAU NEURAL POUR L'ANALYSE DES DONNEES DE TRANSFERT THERMIQUE

Résumé—Le développement récent d'algorithmes puissants et le nombre croissant d'application dans un grand nombre de disciplines suggèrent que les réseaux neuraux peuvent fournir des outils utiles pour modéliser et unifier des problèmes pratiques de transfert thermique. On présente une introduction au calcul par les réseaux neuraux. Pour évaluer la potentialité de ces réseaux pour unifier les données de transfert thermique, trois exemples différents sont résolus en utilisant un réseau à trois couches. Deux algorithmes de connaissance, incluant l'algorithme classique de rétropropagation, sont utilisés. On montre que les réseaux neuraux peuvent unifier correctement les données de transfert thermique.

NEURALE NETZE ZUR ANALYSE VON WÄRMEÜBERGANGSDATEN

Zusammenfassung—Neurale Netze sind bis in die jüngste Vergangenheit Gegenstand akademischer Forschung gewesen. Die neueste Entwicklung leistungsfähiger Lernalgorithmen und die zunehmende Anwendung in vielen Disziplinen lassen neurale Netze auch zur Modellierung und Korrelation praktischer Wärmeübergangsprobleme interessant erscheinen. Die vorliegende Arbeit gibt eine Einführung in die Datenverarbeitung mit Hilfe neuraler Netze. Um das Leistungsvermögen neuraler Netze zur Korrelation von Wärmeübergangsdaten auszuloten, werden drei Beispiele mit einem dreischichtigen vorwärtsarbeitenden Netzwerk behandelt. Zwei verschiedene Lernalgorithmen, auch der bekannte rückwärts modifizierende Algorithmus, werden für Lehrzwecke verwendet. Es wird gezeigt, daß neurale Netze verwendet werden können, um Wärmeübergangsdaten dem jeweiligen Anwendungsfall entsprechend zu korrelieren.

МЕТОДОГИЯ ИСПОЛЬЗОВАНИЯ НЕЙРОННЫХ СЕТЕЙ ДЛЯ АНАЛИЗА ДАННЫХ ПО ТЕПЛОПЕРЕНОСУ

Аннотация—До недавнего времени нейронные сети являлись предметом академических исследований. Последние разработки алгоритмов обучения и увеличение области их применения в большом количестве дисцилин позволяют предположить, что нейронные сети могут успешно использоваться при моделировании практических задач теплопереноса. В данной статье представлено введение к расчетам с применением нейронных сетей. С целью оценки потенциала нейронных сетей, коррелирующих данные по теплопереносу, решаются три различных примера с использованием трехсолойной нейронной сети с прямой связью. Для изучения нейронных сетей используются два различных обучающих алгоритма, включая традиционный алгоритм обратного распространения. Показано, что нейронные сети могут применяться для адекватной корреляции данных по теплопереносу.