ESL-PA-94/07-02 (6)

OR-94-17-6

ARTIFICIAL NEURAL NETWORK BACKPROPAGATION MODEL WITH THREE-PHASE ANNEALING DEVELOPED FOR THE BUILDING ENERGY PREDICTOR SHOOTOUT

Minoru Kawashima
Member ASHRAE

ABSTRACT

An artificial neural network backpropagation model with three-phase annealing was used for the first building energy prediction competition held by the American Society of Heating, Refrigerating and Air-Conditioning Engineers in 1993. Three-phase annealing is an empirical method to gradually reduce the learning rate during the training period in order to improve accuracy in a relatively short time. In this paper, the preprocessing of the competition data, methods of backpropagation training (including three-phase annealing), modeling guidelines for the network, and results of the prediction are presented and discussed.

INTRODUCTION

Prediction techniques are important to engineer any kind of complex system. The ability to predict the dynamic behavior of a target system allows for proper design. Dynamic simulations, which have physical models for heating, ventilating, and air-conditioning (HVAC) systems, have been used to determine the capacity of the equipment during the design stage. These are powerful and essential tools for proper system design; however, they have not been used for adaptive controls because of their lack of adaptability and simplicity.

Recently, a new type of prediction technique has been spotlighted since it has potential applications in real-time prediction, control, and system diagnostics. The significant difference from the former prediction is that the new type of prediction technique adapts to changes in the system. This adaptability requires the use of real data. It can be used for real-time prediction, adaptive system control, real-time system optimization, or real-time self-fault detection.

The building energy prediction competition involved the prediction of building loads. The modeling methods suited for this category are as follows:

1. regression model (linear regression, multiple regression, recursive regression, etc.) (for example, Forrester and Wepfer 1984),
2. time-series model (ARIMA, ARMA, AR, MA, etc.) (MacArthur et al. 1989),
3. kalman filter model (Nakahara and Hachisuka 1977),
4. fuzzy set model (Tobi and Hanafusa 1991), and
5. artificial neural network model (Kreider and Wang 1991; Kreider et al. 1991).

The artificial neural network (ANN) model was used for the energy prediction competition since it has a high potential to model nonlinear processes such as building energy loads.

ANN BACKPROPAGATION MODEL

An ANN model is based on simple models of the human brain. The ANN model has fundamental processing units called neurons. Figure 1 shows a conceptual model of the neuron, which receives several inputs through connections called synapses. The incoming activations are multiplied by the synaptic weights and summed up. The outgoing activation is determined by applying a threshold function to the summation. The neuron has only one outgoing activation value although it might have several connections to the other neurons.

The threshold function is a nonlinear function that decides the output of a particular neuron. Figure 2 shows a popular activation function, called the logistic or sigmoid function.

Backpropagation is the name of a supervised training algorithm by a teacher's (known) data. It is necessary to have both input and output data to train the network. During the training process, the synaptic weights connecting the neurons are gradually adjusted to suitable values. After training, the ANN can estimate the outputs with a given input pattern.

Minoru Kawashima is a research engineer in the Institute of Technology at Shimizu Corporation, Tokyo, Japan.

THIS PREPRINT IS FOR DISCUSSION PURPOSES ONLY, FOR INCLUSION IN ASHRAE TRANSACTIONS 1994, V. 100, Pt. 2. Not to be reprinted in whole or in part without written permission of the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 1791 Tullie Circle, NE, Atlanta, GA 30329. Opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of ASHRAE. Written questions and comments regarding this paper should be received at ASHRAE no later than July 6, 1994.

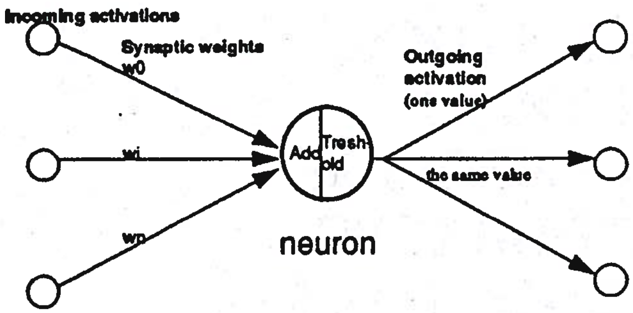

Figure 1 *Diagram of abstract neuron model.*

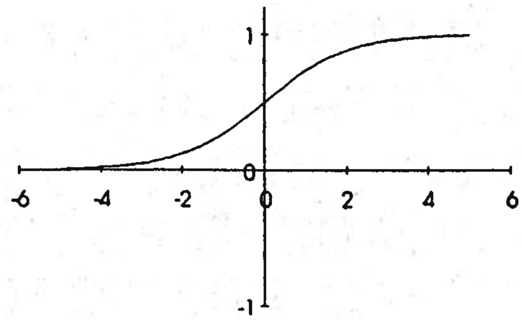

Figure 2 *Shape of logistic function.*

SELF-LEARNING PROCEDURE

Figure 3 shows the fundamental model of the backpropagation ANN with one hidden layer and two synaptic weighting matrices (**W1**, **W2**). Several hidden layers and any combination of numbers for inputs and neurons could be employed.

The procedure of the backpropagation algorithm is as follows:

A. Set random values between −1 and +1 to **W1**, **W2**, **thresh1**, and **thresh2** as the initial values. **thresh1** and **thresh2** are bias vectors.

B. Calculate the activations of the hidden-layer neurons:

$$\mathbf{h} = F(\mathbf{i} \cdot \mathbf{W1} + \mathbf{thresh1}) \tag{1}$$

where

h = hidden-layer neuron vector,
i = normalized input vector (values are from 0.0 to 1.0),
W1 = weighting matrix for the connection between the input and the hidden layer,
thresh1 = bias vector, and
$F(x)$ = $1/(1 + e^{-x})$.

C. Compute the outputs:

$$\mathbf{o} = F(\mathbf{h} \cdot \mathbf{W2} + \mathbf{thresh2}) \tag{2}$$

where

o = output vector (values are from 0.0 to 1.0),
W2 = weighting matrix for the connection between the hidden layer and outputs, and
thresh2 = bias vector.

D. Compute the output error, which is the difference between the teacher's (known) outputs and calculated outputs:

$$\mathbf{d} = \mathbf{o}(\mathbf{1} - \mathbf{o})(\mathbf{o} - \mathbf{t}) \tag{3}$$

where

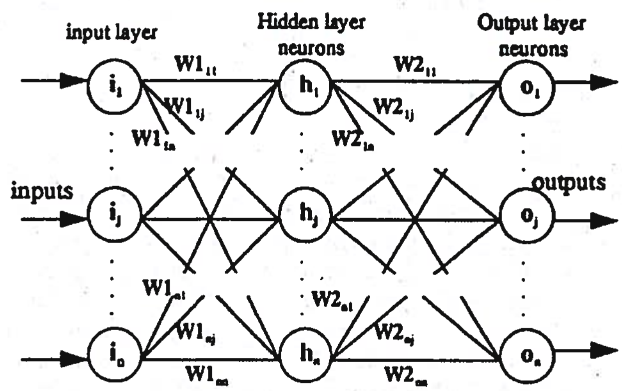

Figure 3 *Diagram of backpropagation model.*

d = output error vector,
o = calculated output vector, and
t = teacher's output vector (normalized 0.0 to 1.0 data).

E. Propagate this error backward to the hidden layer and compute the hidden-layer error:

$$\mathbf{e} = \mathbf{h}(\mathbf{1} - \mathbf{h})\,\mathbf{W2}\,\mathbf{d} \tag{4}$$

where

e = hidden-layer error vector.

F. Adjust the weighting matrix and threshold vector for the output layer:

$$\mathbf{W2} = \mathbf{W2} + \Delta\mathbf{W2}_t, \tag{5}$$

$$\Delta\mathbf{W2}_t = \alpha\mathbf{hd} + \Theta\Delta\mathbf{W2}_{t-1}, \tag{6}$$

and

$$\mathbf{thresh2} = \mathbf{thresh2} + \alpha\mathbf{d} \tag{7}$$

where

$\Delta\mathbf{W2}_t$ = changing value matrix for matrix **W2** at time t,
α = learning rate (value from 0.0 to 1.0), and
Θ = momentum factor (value from 0.0 to 1.0).

G. Adjust the weighting matrix and threshold vector for the hidden layer:

$$\mathbf{W1} = \mathbf{W1} + \Delta\mathbf{W1}_t, \tag{1}$$

$$\Delta\mathbf{W1}_t = \alpha \mathbf{ie} + \Theta\Delta\mathbf{W1}_{t-1}, \tag{2}$$

and

$$\mathbf{thresh1} = \mathbf{thresh1} + \alpha\mathbf{e}. \tag{3}$$

Repeat processes B through G using all of the training data until the sum-of-the-square error for the entire training data set becomes a minimum. The learning rate (α) can be gradually reduced during the training to accelerate the process.

PREDICTION USING TRAINED ANN

Since the trained ANN has adjusted weighting matrices, it can estimate the outputs for specified inputs:

$$\mathbf{h} = F(\mathbf{i} \cdot \mathbf{W1} + \mathbf{thresh1}) \tag{4}$$

and

$$\mathbf{o} = F(\mathbf{h} \cdot \mathbf{W2} + \mathbf{thresh2}). \tag{5}$$

The **o** is denormalized by the minimum and the maximum to get its original units.

THREE-PHASE ANNEALING

Three-phase annealing is an empirical way to gradually reduce the learning rate (α) during training in order to get the lowest sum-of-the-square error in a relatively short period.

The learning rate starts from a number such as 0.5 and is gradually reduced using Equation 13. During the training, the training data set is used again and again. The period when all the training data are used for the backpropagation process (steps B through G) is called an *epoch*. The learning rate is reduced at every epoch throughout the training:

$$\alpha = \frac{C}{\log(1 + N)} \tag{13}$$

where

α = learning rate,
C = constant value, and
N = epoch number.

Figure 4 shows the learning rate as a function of time and the frequency for adjusting the weighting matrices. In phase 1, the weighting matrices are adjusted at each step until the magnitude of the error reverses. In phase 2, the constant (C) in Equation 13 is reduced when the magnitude of the error reverses. The weighting matrices are adjusted at every step. If the learning rate becomes less than 0.0001, the learning rate annealing goes to phase 3. In phase 3, the weighting matrices are adjusted at every epoch until the sum-of-the-square error reaches a minimum.

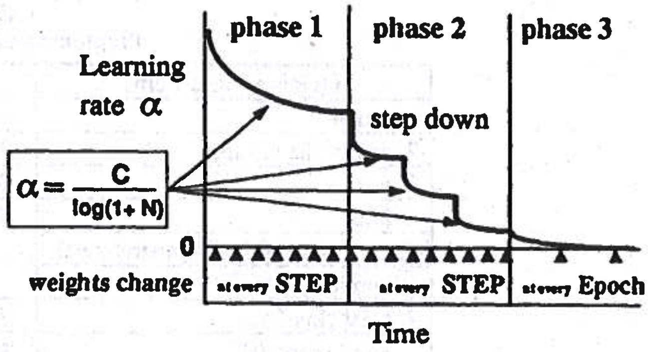

Figure 4 Three-phase annealing method for learning rate.

PREPROCESSING THE COMPETITION DATA

Preprocessing the data is one of the key steps in obtaining better accuracy in the ANN prediction. Tables 1 and 2 show the original and processed items for data sets A and B of the competition. For data set A, the electricity consumption, chilled-water load, and hot-water load were predicted. For data set B, the beam solar radiation was the predicted output. The variable t in Table 1 means current time and $t - 1$ means the current time minus one step (one hour before t). Items 8 through 13 in Table 1 show cross-product items that are commonly used in linear regression. Time-related information (including month, day of the month, year, and hour) in Table 1 was omitted and a weekend indicator (0 or 1) was created. The weekend indicator allows the ANN to identify the significant difference that occurs between weekday and weekend data. The decimal date in Table 2 was transformed into two cosine curves that represent yearly and daily waves.

The items for the ANN model need to be normalized from 0.0 to 1.0 using Equation 14:

$$X_i = \frac{(X_o - X_{min})}{(X_{max} - X_{min})} \tag{14}$$

where

X_i = input data for the ANN,
X_o = original input data,
X_{min} = minimum of the original data, and
X_{max} = maximum of the original data.

RESULTS OF THE PREDICTION

Table 3 shows the results of criteria for the competition. CV, RCV, and MBE are defined by Equations 15 through 17:

coefficient of variation (CV):

TABLE 1
Preprocessing of Data Set A

Original data items	Input items for ANN model
1. month	1. week end indicator (0/1)
2. day of the month	2. dry-bulb ambient temperature: Tdb(t)
3. year	3. Tdb(t-1)
4. hour	4. humidity ratio: Hum(t)
5. dry-bulb ambient temperature(t)	5. solar flux: Sol(t)
6. humidity ratio (t)	6. Sol(t-1)
7. solar flux (t)	7. wind speed: Wind(t)
8. wind speed (t)	8. Tdb(t)×Hum(t)
	9. Tdb(t)×Sol(t)
	10. Tdb(t)×Wind(t)
	11. Hum(t)×Sol(t)
	12. Hum(t)×Wind(t)
	13. Sol(t)×Wind(t)

TABLE 2
Preprocessing of Data Set B

Original data items	Input items for ANN model
1. decimal date	1. $\cos(2\pi(n-172)/365)$
2. horizontal solar flux	2. $\cos(2\pi(t-12)/24)+1$
3. south east solar flux	3. horizontal solar flux
4. south solar flux	4. south east solar flux
5. south west solar flux	5. south solar flux
	6. south west solar flux

TABLE 3
Results of the Prediction Competition

	A data				B data	overall (average)
	Electric Consump.	Cold Water Consump.	Hot Water Consump.	Average	Beam Solar Insolation	
CV(%)	12.79	12.78	30.98	18.85	9.78	16.58
RCV(%)	12.14	11.42	23.07	15.54	2.88	12.38
MBE(%)	7.33	-5.31	-27.1	13.24	-0.31	10.01

$$CV = \frac{\sqrt{\frac{\sum_{i=1}^{n} (y_{pred,i} - y_{data,i})^2}{n}}}{\bar{y}_{data}} \times 100 \; ; \quad (15)$$

robust CV:

$$RCV = \frac{\sqrt{\frac{\sum_{i=1}^{n} \left\{ \begin{array}{c} \textit{omitting the 10\% worst} \\ \textit{values of} (y_{pred,i} - y_{data,i})^2 \end{array} \right\}}{n}}}{\left\{ \begin{array}{c} \textit{distance between 5th and 95th} \\ \textit{percentile of the entire data} \end{array} \right\}} \times 100 \quad (16)$$

mean bias error (MBE):

$$MBE = \frac{\frac{\sum_{i=1}^{n} (y_{pred,i} - y_{data,i})}{n}}{\bar{y}_{data}} \times 100 \quad (17)$$

where

$y_{data,i}$	=	measured data at time i,
$y_{pred,i}$	=	predicted data at time i,
$\bar{y}_{data}$	=	mean value of the measured data, and
n	=	number of data sets in the testing data.

The prediction results for data set A are shown in Figures 5 through 7 for electricity consumption, chilled-water consumption, and hot-water consumption, respectively. The figures show the predicted and measured data and their residuals. The residuals during the first two weeks of January (shown in Figure 5) are relatively larger than in the other period. If there were additional inputs available, such as number of occupying people, the prediction might be closer to the measured data. The residuals in Figure 7 are almost always less than 0. All contestants have negative mean bias errors on this prediction. This bias indicates an underlying difference between the training data period and the testing data period. Actually, we heard a change in building usage after the competition from the coordinator.

Figure 8 shows the results for predicted beam solar insolation for data set B. The disagreement at high radiation levels suggests that the network should have been trained longer with a smaller learning rate.

GUIDELINES OF ANN MODELING FOR IMPROVED ACCURACY

Guidelines for ANN model development based on the author's experience can be divided into two categories. The first regards the network structure and the second category regards processing of the inputs and outputs. Figure 9 shows a typical structure of the artificial neural network to illustrate the guidelines.

ANN Structure

1. One hidden layer is enough for a load prediction.
2. The number of hidden-layer neurons (p) should be more than $2n+1$, where n is the number of inputs.
3. The number of output neurons should be one (even though the multi-output model can be specified).

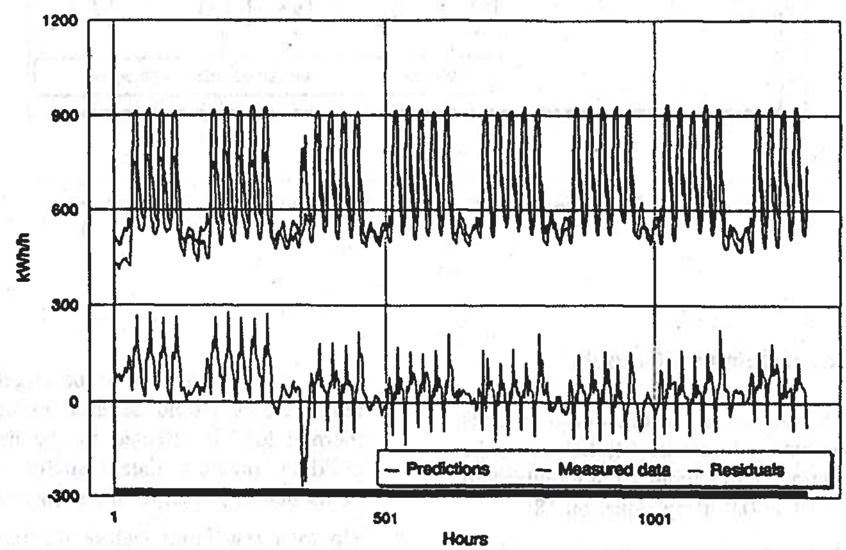

Figure 5 Prediction results on data set A (electricity consumption).

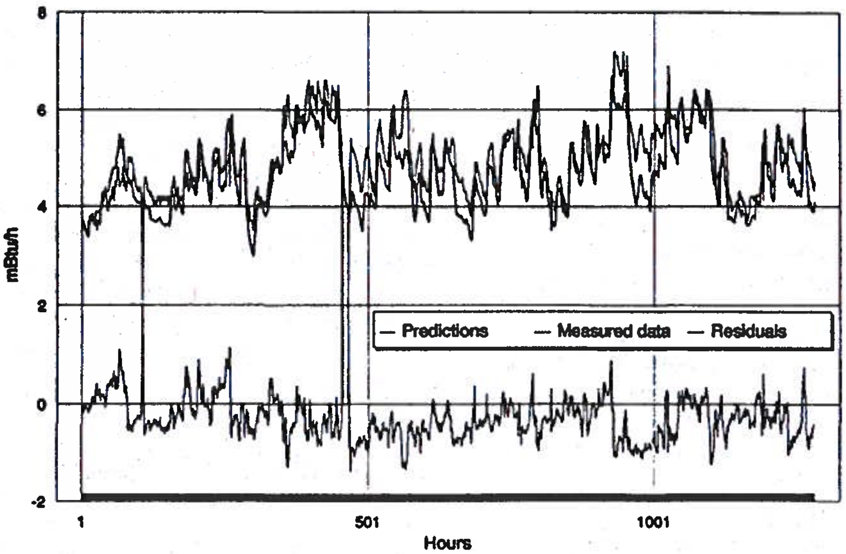

Figure 6 *Prediction results on data set A (cold water consumption).*

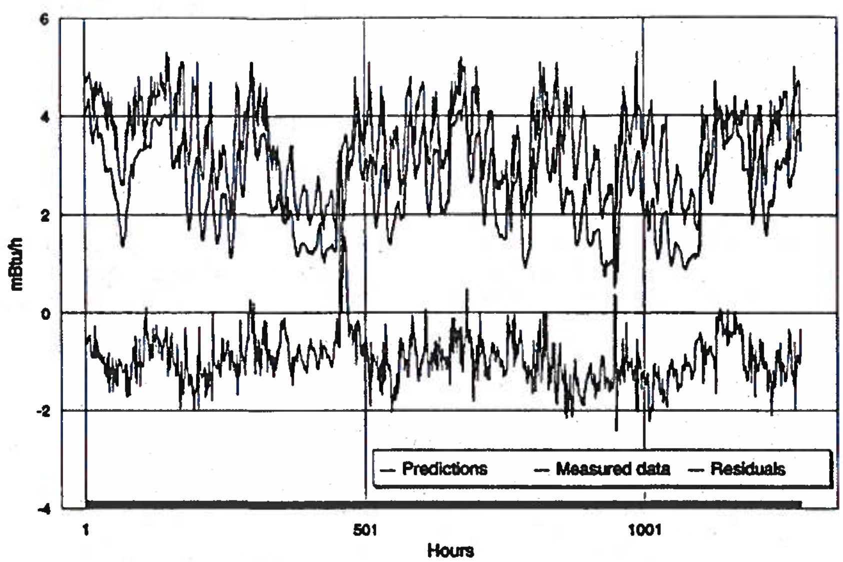

Figure 7 *Prediction results on data set A (hot water consumption).*

Preprocessing of the Inputs and Outputs

1. Every input must be normalized from its original units to 0.0 to 1.0 data using Equation 14; however, the output for the training data should be normalized to data ranging from 0.1 to 0.9 using Equation 18:

$$X_i = \frac{\{X_o - X_{min} - 0.1 \cdot (X_{max} - X_{min})\}}{1.2 \cdot (X_{max} - X_{min})} . \qquad (18)$$

3. Up to a few hours before the target time, the ambient temperatures should be used as the input. Since the thermal load is affected by the thermal mass of the building, previous data (usually up to two or three hours before the target time) should be used.

4. Up to a few hours before the target time, the solar insolation should be used as the input for the same reason.

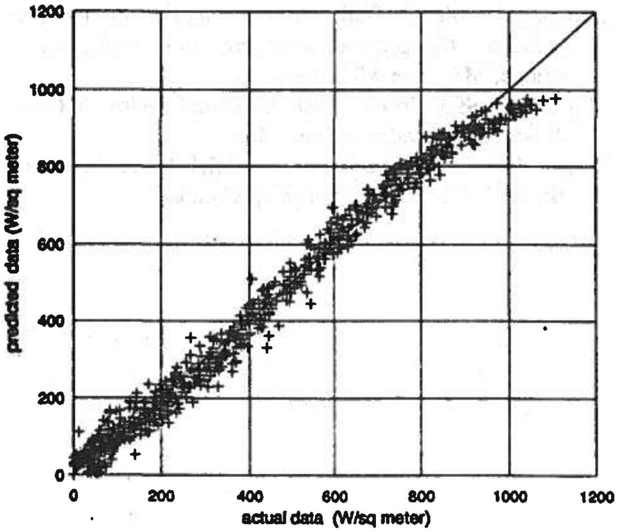

Figure 8 *Prediction results on data set B (beam insolation).*

5. The year, month, and day data can be omitted if the prediction period is not included in the training data period. For example, it is not appropriate to use month data item "1" (January) for prediction if the training data period is December (month data item "12").
6. The time data (0-24) can be transformed to a sine or cosine product using Equation 19:

$$T_i = \sin\left(2\pi\,\frac{T_o}{24}\right) \quad \text{or} \quad T_i = \cos\left(2\pi\,\frac{T_o}{24}\right) \tag{19}$$

where

T_i = input data for the ANN, and
T_o = original time data (0-24).

7. Cross-products of inputs, such as ambient temperature times solar insolation, may be used if the magnitude of error in the training is not satisfactory.

CONCLUSIONS

Many different ANN model configurations were examined for the four types of data prediction in the competition. Through a trial-and-error process, the guidelines listed in this paper became clear in order to get better prediction accuracy. The ANN does not require a physical model; however, a way of thinking oriented toward the physical model is recommended. In other words, previous temperature or solar insolation data up to few hours before the target time should also be used as input if the output is considered to have a time delay factor because of the thermal capacitance of the object.

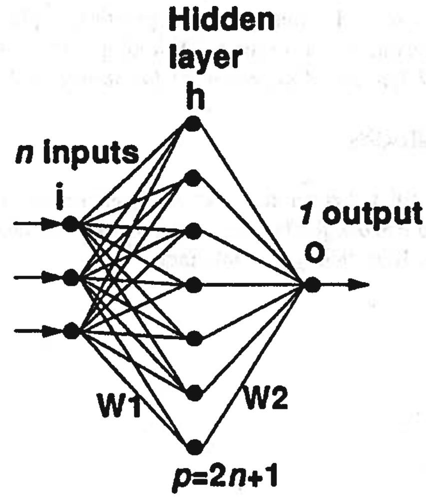

Figure 9 *Typical structure of an ANN model.*

With respect to the learning rate, smaller initial values almost always brought good results; however, it took longer to train. The three-phase annealing method reduces the training time without losing prediction accuracy. It took about 6 to 12 hours for one case with three-phase annealing and two to three days without three-phase annealing on a personal computer equipped with a 33-MHz CPU.

ACKNOWLEDGMENTS

The author is grateful to the Thermal Storage Applications Research Center and the Solar Energy Laboratory at the University of Wisconsin-Madison for technical support.

REFERENCES

Forrester, J.R., and W.J. Wepfer. 1984. Formulation of a load prediction algorithm for a large commercial building. *ASHRAE Transactions* 90(2B): 536-551.

Kreider, J.F., and X.A. Wang. 1991. Artificial neural network demonstration for automated generation of energy use predictors for commercial buildings. *ASHRAE Transactions* 97(1).

Kreider, J.F., X.A. Wang, D. Anderson, and J. Dow. 1991. Expert systems, neural networks and artificial intelligence applications in commercial building HVAC operations. *Proceedings of the Building Systems Automation-Integration*, pp. 4.3.1-4.3.24. Madison: University of Wisconsin.

Nakahara, N., and S. Hachisuka. 1977. Air conditioning facilities of Osaka Ohbayashi building and its on-line computer control operating systems of optimization and prediction features. *Proceedings of the Second International Symposium on the Use of Computers for Environmental Engineering Related to Buildings*, Banff, Canada.

Tobi, T., and T. Hanafusa. 1991. A practical application of fuzzy control for an air-conditioning system. *International Journal of Approximate Reasoning* 5(3).

BIBLIOGRAPHY

Blum, A. 1993. *Neural networks in C++: An object-oriented framework for building connectionist systems.* New York: John Wiley & Sons, Inc.

Caudill, M., and C. Butler. 1992. *Understanding neural networks: Computer explorations,* vols. 1 and 2. Cambridge, MA: The MIT Press.

Masters, T. 1993. *Practical neural network recipes in C++.* New York: Academic Press, Inc.

Zurada, J.M. 1992. *Introduction to artificial neural systems.* St. Paul, MN: West Publishing Company.