APPLICATION OF NEURAL NETWORKS FOR SENSOR VALIDATION AND PLANT MONITORING

FISSION REACTORS

KEYWORDS: *neural networks, sensor validation, reactor monitoring*

BELLE R. UPADHYAYA and EVREN ERYUREK
The University of Tennessee, Knoxville, Tennessee 37996-2300

Received January 17, 1991
Accepted for Publication June 27, 1991

Sensor and process monitoring in power plants requires the estimation of one or more process variables. Neural network paradigms are suitable for establishing general nonlinear relationships among a set of plant variables. Multiple-input/multiple-output autoassociative networks can follow changes in plantwide behavior. The backpropagation (BPN) algorithm has been applied for training multilayer feedforward networks. A new and enhanced BPN algorithm for training neural networks has been developed and implemented in a VAX workstation. Operational data from the Experimental Breeder Reactor II (EBR-II) have been used to study the performance of the BPN algorithm. Several results of application to the EBR-II are presented.

I. INTRODUCTION

For sensor validation and plantwide monitoring, neural networks offer several advantages over traditional empirical methods.[1] The problem of sensor validation concerns the detection of incipient changes in sensor behavior. Plantwide monitoring is useful as a predictor of plant status and for isolating suspect instrumentation channels.[2] Both of these may be incorporated in an operator-assistance system. Neural network paradigms do not require the definition of a functional form relating a set of process variables. The functional form created by an artificial neural system is implicitly nonlinear. An artificial neural system is a parallel distributed network, with information flow fully connected from one processing element to the others.[3] This may render it more fault tolerant if it is carefully designed. The key issue in developing a neural network is training the network to associate one set of information with another. In general, a multilayer perceptron is capable of performing arbitrary mapping from data to data. Both steady-state and transient behavior can be incorporated into the network during training.

Neural networks have usually been applied to problems that require classification of patterns and detection of discrete anomaly types. This study uses neural networks to estimate continuous variables, similar to an interpolating polynomial. The applications have been extended beyond nuclear systems and include diagnostics in process industries (rolling mills) and chemometric data analysis for chemical composition estimation using Raman spectroscopy.[4]

A detailed discussion of the establishment of optimal network architecture and the choice of intermediate layer nodes using Shannon and Weaver's information theory[5] is presented.

An enhanced version of the backpropagation (BPN) algorithm is described in this paper. The implementational aspects of neural networks are emphasized. Section II presents an outline of multilayer perceptrons, including dynamic networks, and their use for single or multivariate signal prediction. The key features of the adaptive BPN algorithm are described in Sec. III. Section IV describes the application to sensor validation using startup data from Experimental Breeder Reactor II (EBR-II). The application to plantwide monitoring using multiple signal estimation (autoassociative networks) is presented in Sec. V. Concluding remarks are given in Sec. VI.

II. MULTILAYER PERCEPTRONS FOR SIGNAL ESTIMATION AND PLANT MONITORING

II.A. Multilayer Perceptrons

Artificial neural networks have been used very successfully in pattern classification problems.[6] These are discrete state estimation problems. By increasing the complexity of the processing elements and using input and output data samples of a continuous process, the network takes the form of an interpolation and extrapolation filter. Figure 1 is the schematic of a three-layer perceptron showing the input layer, the intermediate or hidden layer, and the output layer. The processing elements are nonlinear operators, and the output has the form

$$T(x) = \frac{1}{1 + \exp(-\beta x)} \quad , \quad \beta > 0 \ . \tag{1}$$

The threshold function $T(x)$ plays a very important role in network training and data association. Figure 2 shows the output of the j'th processing element and has the form

$$x_{jp}^{p} = T\left(\sum_i w_{ij}^{l} x_{ip}^{l-1} + \theta_j^{l} \right) \quad , \tag{2}$$

where

w_{ij}^{l} = connection weights from layer $(l-1)$ to layer l and from node i to node j

θ_j^{l} = bias associated with the j'th processing element of layer l

$T(\cdot)$ = sigmoidal function defined in Eq. (1).

The behavior of the sigmoidal threshold function is shown in Fig. 3 for various values of parameter β.

When only snapshots of the process signals are used for estimating one or more variables, the network may be considered as a steady-state form of a more general dynamic (or time-dependent) network. The networks developed in this study have the general mathematical form

$$y_i(t) = g_i[x_1(t), x_2(t), \ldots, x_n(t)] \quad , \tag{3}$$

where the output $y_i(t)$ is a function of the network input vector $\{x_1(t), x_2(t), \ldots, x_n(t)\}$.

State variables, autoregression terms, and control variables may be included in the above formulations, so that artificial neural system models can be used for control applications.[7] Note that the complexity of training the network increases with more input terms.

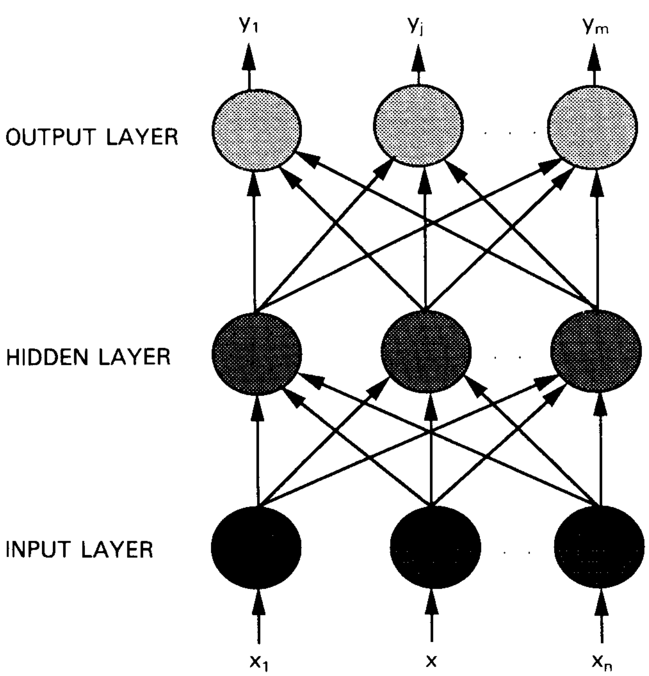

Fig. 1. A three-layer feedforward network with n inputs and m outputs.

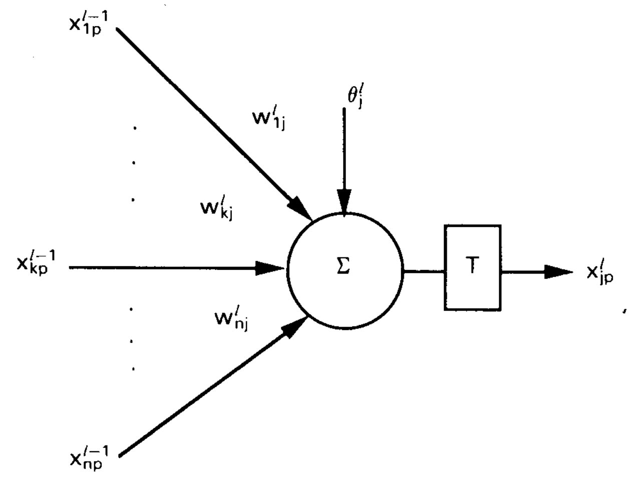

Fig. 2. A typical processing element used in the BPN algorithm.

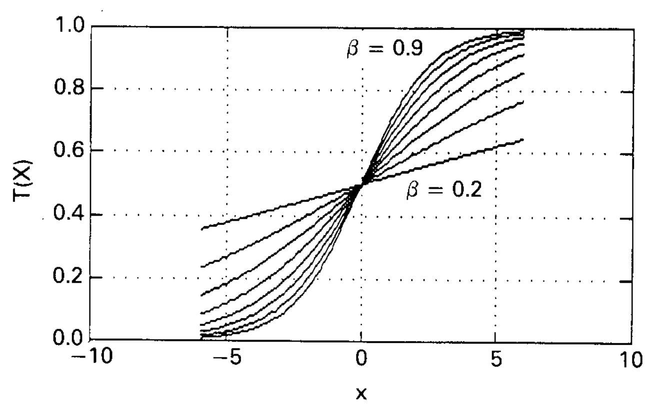

Fig. 3. Typical sigmoidal threshold functions used in the study of signal modeling.

II.B. Network Training and BPN Algorithm

The key to successful application of neural networks lies in training the network that would represent the signal association as accurately as possible. Several modifications of the BPN algorithm[3] have been incorporated in the current implementation. The BPN algorithm is an iterative gradient descent algorithm that minimizes the global error between the output of a multilayer feedforward network and the actual output. The BPN algorithm requires continuously differentiable nonlinearities such as the sigmoidal function. The algorithm is described in Ref. 2 and uses the generalized delta rule.[3] The connection weights and node biases are updated recursively as

$$w_{ij}^{l}(n+1) = w_{ij}^{l}(n) + \alpha \Delta w_{ij}^{l}(n+1) \tag{4}$$

and

$$\theta_{j}^{l}(n+1) = \theta_{j}^{l}(n) + \Delta \theta_{j}^{l}(n+1) \ . \tag{5}$$

The network training requires presentation of data corresponding to different time sequences at each iteration step. The iteration is terminated when the global error reaches a predefined level.

III. FEATURES OF THE ADAPTIVE ALGORITHM

The new algorithm, which is an improvement over the existing BPN training algorithm, incorporates features that improve the speed and accuracy of network training in the applications described in Secs. IV and V; they are enumerated here.

III.A. Adaptive Threshold Shaping

The algorithm allows on-line changes in the sigmoidal shaping parameter β [Eq. (1)]. The problem of the saturation of local processing elements can be minimized by forming a more linear threshold function during the early stages of learning and then progressively increasing its nonlinearity as convergence is achieved. The training can be stopped temporarily to check the convergence properties.

III.B. Automatic Scaling of Signals

All the input and output signals are scaled to be in a range such as [0.1, 0.9]. This range allows any signal value that may exceed the trained domain and thus facilitates in extrapolating variable estimates. All scaling is performed automatically by the algorithm. The use of scaling and adaptive thresholding is illustrated in Ref. 2.

III.C. Weight Updating and Momentum Parameters

The weight updating parameter α [see Eq. (4)] is progressively increased after the initial learning period. This increases the convergence rate. The momentum term helps in avoiding stagnancy at local minima. Both parameters are varied in the range [0.1, 1.0].

III.D. Hidden Layer Nodes

One of the important areas of ongoing research is to establish an appropriate number of processing elements in the intermediate layers. In pure pattern classification problems, the hidden layer nodes are set to be equal to the number of independent patterns. Too many nodes will result in overspecification of connection weights and increased weight error. An attempt is being made to establish the number of nodes based on Shannon and Weaver's information theory.[5]

In multilayer network applications, one of the most important issues is the number and size of the intermediate (hidden) layers. There have been various studies related to this topic, but there is no definite solution to this important problem.[8,9] Since the hidden layer represents the nonlinear properties of the data, it has a significant effect on the convergence and accuracy of the network models. Generally, multilayer networks have more than one hidden layer. The results of this study indicate that using only one hidden layer is sufficient to solve the problems in the area of signal processing, plant monitoring, parameter estimation, and sensor validation.[2] This observation makes it easy to determine the number of hidden layer nodes. Other important factors in deciding the optimum size of the hidden layer are the number of training patterns and the size of the input vector.

A formula has been derived for estimating the optimum hidden layer size using Shannon and Weaver's information theory.[5] This formula has been derived from the following facts:

1. Three-layer networks are sufficient to solve estimation problems.
2. The minimum number of required hidden units is $\log_2 N$ (Ref. 3), where N is the number of training patterns.

In the BPN algorithm, communication between the input vector and the output vector is made through the hidden layers; therefore, the size of the input vector also has an important effect that must be incorporated into the formula.

Depending on these facts, the empirical formula for estimating the optimum hidden layer size is given by

$$H = I \times \log_2 N \ , \tag{6}$$

where I is the size of the input vector and N is the number of training patterns. The formula has been successfully applied to the problems analyzed in this study.

Consider a problem in which a set of orthogonal input patterns are mapped to a set of orthogonal output patterns. Suppose N input patterns are to be mapped onto N output patterns. If a hidden layer with $\log_2 N$ nodes is selected, the system will learn to use the hidden nodes to form a binary code with distinct binary patterns for each of the N input patterns. Suppose each of the patterns appears with an *a priori* probability of $p_i = 1/N$. Then a measure of choice or information is given by Shannon and Weaver's information theorem as

$$H = -K \sum_{i=1}^{N} p_i \log_2 p_i \ , \quad K > 0 \tag{7}$$

$$= -K \sum_{i=1}^{N} \frac{1}{N} \log_2 \left(\frac{1}{N}\right)$$

and

$$H = K \log_2 N \ . \tag{8}$$

It was shown by Shannon and Weaver[5] that the information H defined in Eq. (8) is the only function that satisfies the following properties:

1. Information H should be a continuous function of p_i.

2. If all the p_i are equal, $p_i = 1/N$, then H should be a monotonically increasing function of N. That is, when there are more events, there is more choice.

3. If a choice is broken down into successive choices, the original H should be the weighted sum of the individual values of H.

In communication theory, Eq. (8) is often referred to as Shannon's theorem. In this study, an analogy has been developed between the number of training patterns (choices) and the number of intermediate layer nodes as the number of binary coded bits required to represent the pattern classes. It is important to note that this number matches closely with Shannon and Weaver's information H, thus providing effective information classification. However, the patterns considered in the present application need not necessarily be orthogonal vectors. Thus, the determination of the number of intermediate layer nodes using this approach provides a conservative estimate (an upper bound) of this parameter.

IV. APPLICATION TO SENSOR VALIDATION

This application considers the prediction of a selected variable as a function of other process variables. Startup data from EBR-II were used to illustrate the application to sensor validation. Reactor power was used as the output of a three-layer network with control rod position, core exit temperature, and intermediate heat exchanger (IHX) secondary sodium outlet temperature as inputs to the network. A summary of network and training parameters is given in Table I. Figure 4 shows a comparison of the measured and network-predicted values of reactor power from ~0 to 100%. The data points shown were not used in training the network. Good network convergence was

TABLE I

Network for Predicting the Power (%) Level in EBR-II

Input signals	
Control rod position (in.)	
Core exit temperature (°F)	
IHX secondary sodium outlet temperature (°F)	
Number of training patterns	150
Overall prediction error standard deviation	0.40
Threshold shaping parameter, β	0.3
Weight update coefficient, α	0.7
Momentum term coefficient, μ	0.9
Number of hidden layer nodes	21

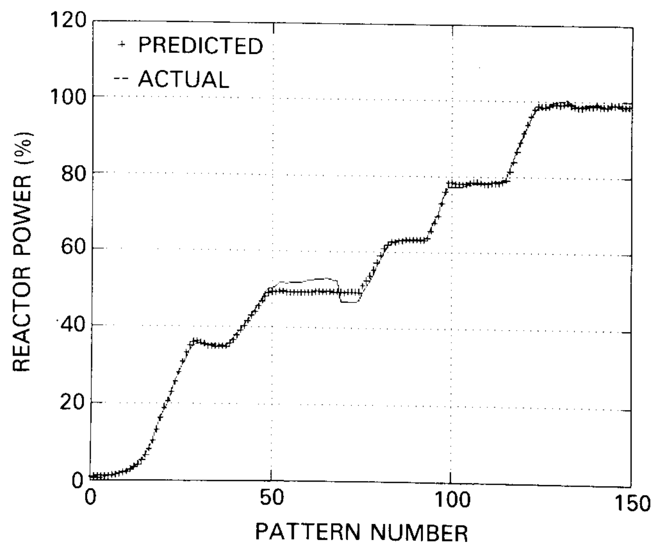

Fig. 4. Comparison of measured and network-predicted values of reactor power in EBR-II.

achieved within 5000 iterations. Each iteration updates the weights for all patterns presented. The training was continued until the error decreased to ~0.5%. The number of nodes in the hidden layer was selected using Shannon and Weaver's information criterion. The network shows a high fidelity of signal prediction for a wide range of power variation (0 to 100%).

An example of estimating the IHX secondary sodium outlet temperature is summarized in Table II. The comparison of measured and network-predicted temperature variation during startup is shown in Fig. 5. In many applications, it is also important to validate the control strategies as designed by a control algorithm. This is often referred to as command validation.[10] The control rod position was predicted by developing a network with reactor sodium core exit temperature, IHX orifice plate temperature, and bulk sodium pool temperature as input variables. The details

TABLE II

Network for Predicting the IHX Secondary Outlet Temperature (°F) in EBR-II

Input signals	
Core exit temperature (°F)	
Upper plenum temperature (°F)	
Sodium inlet header temperature (°F)	
Number of training patterns	150
Overall prediction error standard deviation	0.531
Threshold shaping parameter, β	0.3
Weight update coefficient, α	0.8
Momentum term coefficient, μ	0.9
Number of hidden layer nodes	21

TABLE III

Network for Predicting the Control Rod Position (in.) in EBR-II

Input signals	
Core exit temperature (°F)	
IHX orifice plate temperature (°F)	
Bulk sodium pool temperature (°F)	
Number of training patterns	150
Overall prediction error standard deviation	0.044
Threshold shaping parameter, β	0.3
Weight update coefficient, α	0.7
Momentum term coefficient, μ	0.9
Number of hidden layer nodes	21

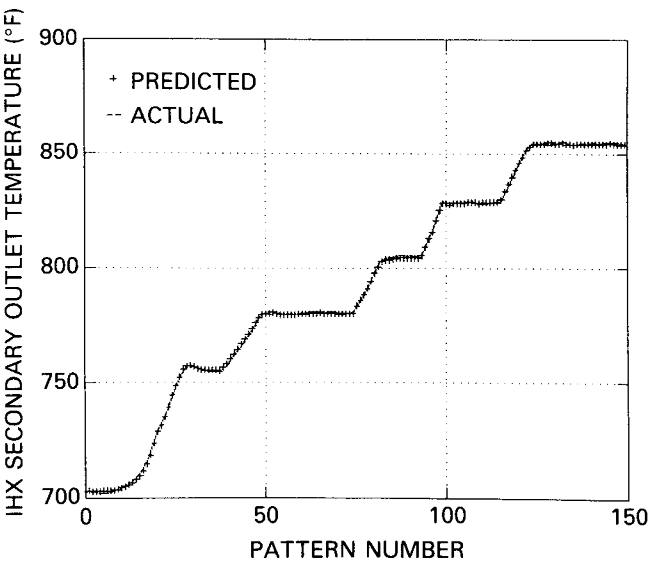

Fig. 5. Comparison of measured and network-predicted values of IHX secondary outlet temperature (°F).

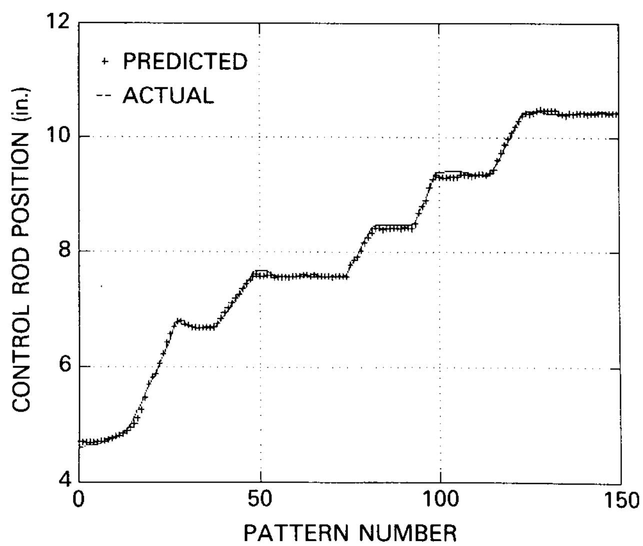

Fig. 6. Comparison of measured and network-predicted values of control rod position (in.).

of the network are given in Table III. The network prediction performance is shown in Fig. 6. The data points shown in Figs. 5 and 6 were not used in training the networks.

V. PLANTWIDE MONITORING OF EBR-II PROCESS VARIABLES

The application of neural networks presented here is concerned with the prediction of several plant variables. The three-layer network was trained in the autoassociative mode (input and output variables are the same). Two separate networks were generated, one for the primary-side variables and the other for the secondary-side variables. Tables IV and V provide details of the two networks used for plantwide monitoring. Figure 7 shows a comparison of the measured and predicted values of primary variables (Table IV). All the signals are normalized in the range [0.1, 0.9], and two patterns of the eight signals are shown.

A similar network for the ten secondary variables (see Table V) was developed using 150 data points between 60 and 80% power. Figure 8 illustrates the usefulness of the network for detecting the deviation in a signal from the nominal value at a given power level. Thus, autoassociative networks may be used to track the entire plant behavior or to detect deviations in a small set of signals caused by sensor degradation.

An error was introduced in the third variable (IHX secondary sodium outlet temperature). The resulting pattern of signals was presented to the network. The

TABLE IV

Network for Monitoring Primary Variables in EBR-II

Power level (%)
Core exit temperature (°F)
Control rod position (in.)
Primary pump flow rate (%)
High-pressure plenum sodium temperature (°F)
Low-pressure plenum sodium temperature (°F)
IHX primary outlet sodium temperature (°F)
Core upper plenum temperature (°F)

TABLE V

Network for Monitoring Secondary Variables in EBR-II

Secondary sodium flow rate (%)
IHX secondary sodium inlet temperature (°F)
IHX secondary sodium outlet temperature (°F)
Superheater sodium inlet temperature (°F)
Superheater sodium outlet temperature (°F)
Evaporator sodium inlet temperature (°F)
Evaporator sodium outlet temperature (°F)
Steam drum level (in.)
Steam drum pressure (lb/in.^{2})
Steam drum feedwater flow rate (lb_m/h)

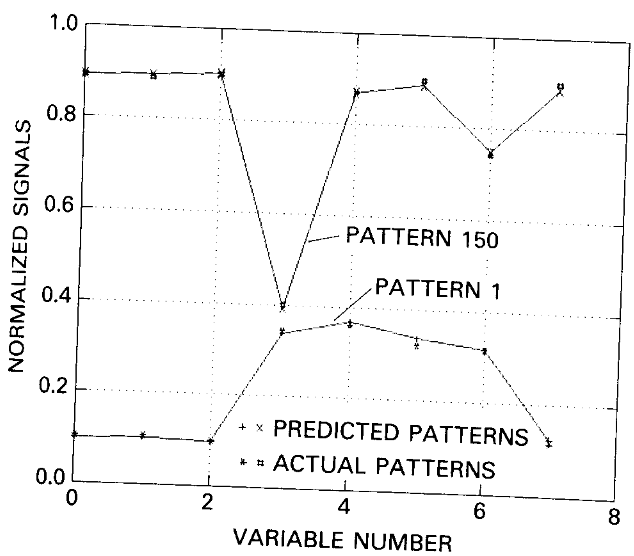

Fig. 7. Plantwide monitoring of primary-side signals in EBR-II (Table IV): comparison of measured and predicted values at two time instants.

network output matched with all the variables, except the one with the anomaly. Thus, the deviation from a normal signal pattern can be identified, and the anomalous signal can be isolated.

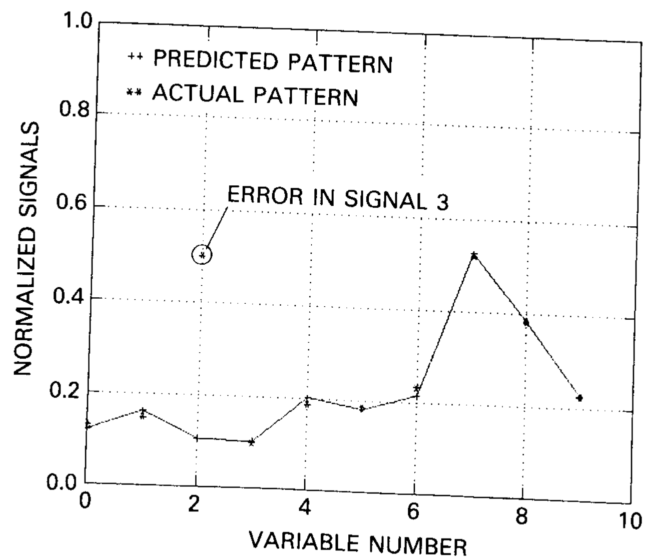

Fig. 8. Plantwide monitoring of secondary-side signals in EBR-II (Table V): comparison of measured and predicted values of ten signals. An error introduced in signal 3 is correctly detected as indicated by its estimated value (normal pattern).

VI. CONCLUDING REMARKS

An approach for continuous estimation of process variables using neural networks has been developed. The technique is illustrated with application to sensor validation and plantwide monitoring of signals in EBR-II. An adaptive BPN algorithm has been used for the training of three-layer networks. The high fidelity with which process variables can be predicted indicates that neural networks can be used as estimators in place of physical or empirical models. Research is continuing at the University of Tennessee to establish guidelines for optimal network generation for problems in the power and process industries.

ACKNOWLEDGMENTS

The research reported here was sponsored by a grant from the U.S. Department of Energy under contract DE-FG07-88ER12824. The EBR-II data were acquired from Argonne National Laboratory-West. This assistance is gratefully acknowledged.

REFERENCES

1. B. R. UPADHYAYA et al., "Development and Testing of an Integrated Signal Validation System for Nuclear Power Plants," Vols. 1, 2, and 3, DOE/NE/37959-34, 35, 36, U.S. Department of Energy (Sep. 1989).

2. E. ERYUREK and B. R. UPADHYAYA, "Sensor Validation for Power Plants Using Adaptive Backpropagation

Neural Network," *Proc. IEEE Nuclear Science Symp.*, San Francisco, California (1989).

3. D. E. RUMELHART and J. L. McCLELLAND, *Parallel Distributed Processing*, Vol. 2, The MIT Press, Cambridge, Massachusetts (1987).

4. B. R. UPADHYAYA, G. MATHAI, and E. ERYUREK, "Development and Application of Neural Network Algorithms for Process Diagnostics," *Proc. 29th IEEE Conf. Decision and Control*, Honolulu, Hawaii, December 1990.

5. C. E. SHANNON and W. WEAVER, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, Illinois (1971).

6. R. P. LIPPMANN, "An Introduction to Computing with Neural Nets," *IEEE ASSP*, **4**, *2*, 4 (1987).

7. K. S. NARENDRA and K. PARTHASARATHY, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks*, **1**, *1*, 4 (1990).

8. S. Y. KUNG and J. N. HWANG, "An Algebraic Projection Analysis for Optimal Hidden Units Size and Learning Rates in Back-Propagation Learning," *Proc. 1st Int. Conf. Neural Networks*, San Diego, California, June 1987.

9. E. B. BAUM and D. HAUSSLER, "What Size Net Gives Valid Generalization?" *Neural Computation*, **1**, 151 (1989).

10. R. L. BYWATER, B. R. UPADHYAYA, R. C. BERKAN, and R. A. KISNER, "Command Validation of Secondary EM Pump System in the EBR-II," *Trans. Am. Nucl. Soc.*, **62**, 412 (1990).

Belle R. Upadhyaya (PhD, systems science, University of California–San Diego, 1975) is a professor of nuclear engineering at the University of Tennessee–Knoxville, director of the Preventive Maintenance Engineering Laboratory, and an adjunct research associate at Oak Ridge National Laboratory. His research interests include digital signal processing, instrumentation and controls, plant surveillance and diagnostics, and neural networks.

Evren Eryurek (BS, physics, Hacettepe University, Turkey, 1986; MS, nuclear engineering, University of Tennessee–Knoxville, 1991) has had 2 years of industrial working experience as an engineer. He is currently pursuing a PhD degree in the nuclear engineering department at the University of Tennessee.