Conf - 900804--35

NEURAL NETWORKS FOR SENSOR VALIDATION AND PLANT MONITORING

CONF-900804--35

DE93 002127

B.R. Upadhyaya, E. Eryurek and G. Mathai

The University of Tennessee
Knoxville, Tennessee 37996-2300, U.S.A.

ABSTRACT

Sensor and process monitoring in power plants require 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 plant-wide behavior. The backpropagation algorithm has been applied for training feedforward networks. A new and enhanced algorithm for training neural networks (BPN) 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 BPN. Several results of application to the EBR-II are presented.

1. INTRODUCTION

For sensor validation and plant-wide monitoring, neural networks offer several advantages compared to traditional empirical methods[1]. The problem of sensor validation concerns the detection of incipient changes in the sensor behavior. Plant-wide monitoring is useful as a predictor of plant status and for isolating suspect instrumentation channels[2]. Both of these may be used as part of operator assist systems. 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 (ANS) is implicitly nonlinear. The ANS is a parallel distributed network, with information flow fully connected from one processing element to others[3]. Thus it has the property of fault tolerance. The key issue in developing a neural network is training the network to associate one set of information with another. In general a multi-layer 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.

1990 International Fast Reactor Safety Meeting, August 12-16, 1990
Snowbird, Utah.

do FG07-88ER12824

An enhanced version of the BPN algorithm is described in this paper. We want to emphasize the implementational aspects of neural networks for existing processes. Section 2 presents an outline of multi-layer perceptrons, including dynamic networks, and their use for single or multivariate signal prediction. The key features of the adaptive BPN algorithm are described in Sect. 3. Section 4 describes the application to sensor validation using startup data from EBR-II. The application to plant-wide monitoring using multiple signal estimation (autoassociative networks) is presented in Sect. 5. Concluding remarks are given in Sect. 6.

2. MULTI-LAYER PERCEPTRONS FOR SIGNAL ESTIMATION AND PLANT MONITORING

2.1 Multilayer Perceptrons

Traditionally neural networks have been used in pattern classification problems[4]. This is a discrete state estimation problem. 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--input layer, intermediate or hidden layer, and output layer. The processing elements (PEs) are nonlinear operators and the output has the form

$$T(x) = \frac{1}{1+e^{-\beta x}} , \quad \beta>0 \qquad (1)$$

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 PE and has the form

$$x_j^p = T(\sum_i w_{ij}^p x_i^{p-1} + \phi_j^p) \qquad (2)$$

where w^p_{ij} are the connection weights from layer (p-1) to layer p, and from node i to node j. ϕ^p_j is the bias associated with the j-th PE of layer p. T(·) is the sigmoidal function defined in Eq. (2).

The network shown in Fig. 1 is considered to be a dynamic network because the output $y_i(t)$ is a function of $\{x_1, x_2, ..., x_n\}$ at time instants (t, t-Δt, t-2Δt). Only two time lags are indicated for the sake of simplicity. Thus the model represented by the ANS in Fig. 1 has the form

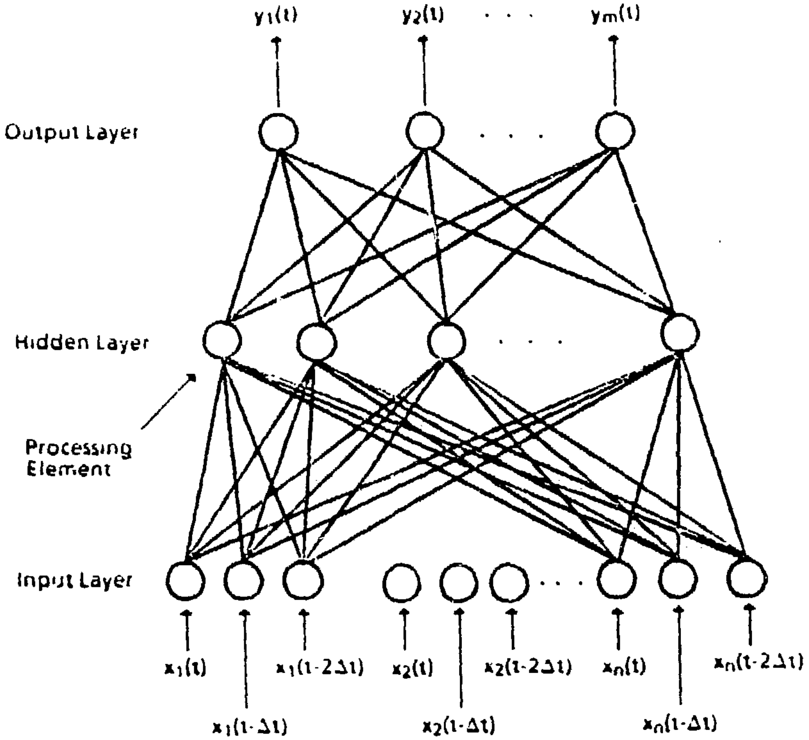

Figure 1. A three-layer dynamic network with n inputs and m outputs.

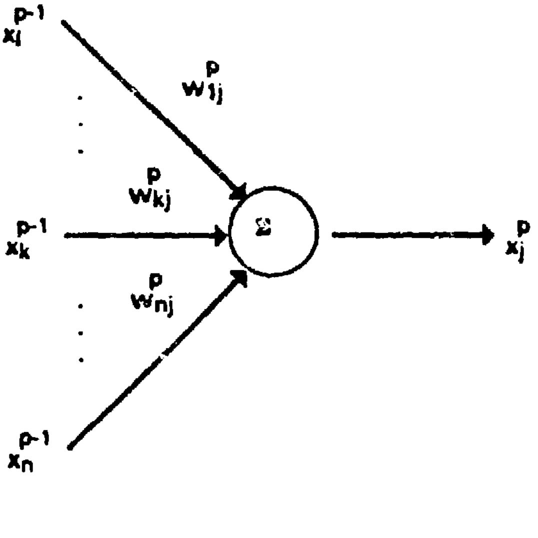

Figure 2. A typical processing element in the backpropagation algorithm.

$$Y_i(t) = f_i[x_1(t),\ldots,x_1(t-l\Delta t);x_2(t),\ldots x_2(t-l\Delta t);\ldots,x_n(t) \ldots x_n(t-l\Delta t)]. \quad i = 1,2,\ \ldots,m. \tag{3}$$

where l is the total number of lags of this regression model. Note that the model does not require the assumption of a linear regression or a functional form for f_i. A static form of the network has the form

$$Y_i(t) = g_i(x_1(t),x_2(t),\ldots,x_n(t)) \tag{4}$$

State variables, autoregression terms, and control variables may be included in the above formulations, so that ANS models can be used for control applications[5]. Note that the complexity of training the network increases with more dynamic terms.

2.2 Network Training and Backpropagation 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 backpropagation (BPN) algorithm[3] have been incorporated in the current implementation. The backpropagation algorithm is an iterative gradient algorithm that minimizes the global error between the output of a multi-layer feedforward network and the actual output.
The BPN algorithm requires continuous differential nonlinearities such as the sigmoidal function. The algorithm is fully described in Ref. 2 and uses the generalized delta rule[3].

The connection weights are updated using the recursion formula (k is the iteration step)

$$w^p_{ij}(k+1) = w^p_{ij}(k) + \alpha\delta^p_j x^{p-1}_i + \mu(w^p_{ij}(k) - w^p_{ij}(k-1)), \qquad \alpha>0,\ 0<\mu<1. \tag{5}$$

δ^p_j is referred to as the generalized delta value and is a function of the output error at the upper level node. The last term in Eq. (5) is called the momentum term and is a function of the change in the connection weight during the two previous iterations. The momentum term helps to eliminate stagnancy at local minima.

The network training requires presentation of data corresponding to different time sequences after each iteration step. The iteration will be terminated when the global error reaches an acceptable level. The backpropagation algorithm is the workhorse of predictive modeling networks.

3. FEATURES OF THE ADAPTIVE NETWORK

The new algorithm, which is an improvement over the existing backpropagation training algorithm, incorporates features that improve the speed and accuracy of network training in the applications described in Sects. 4 and 5, and are enumerated here.

3.1 Adaptive Threshold Shaping

The algorithm allows the changes in the sigmoidal shaping parameter β (Eq.(1)). The problem of saturation of local PEs 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.

3.2 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 are illustrated in Ref. 2.

3.3 Weight Updating and Momentum Parameters

The weight updating parameter α (see Eq. (5)) 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 (α and β) are varied in the range (0.1, 1.0).

3.4 Hidden Layer Nodes

One of the important areas of on-going research is to establish an appropriate number of PEs in the intermediate layers. In pure pattern classification problems the hidden-later nodes are set 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's information theoretic approach[2].

4. 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 1. Figure 3 shows a comparison of the measured and network predicted values of reactor power from about 45% to 100% level. Good network convergence was achieved within 5000 iterations (% error of 0.802). The training was continued until the error decreased to about 0.5%. The training was performed off-line. The number of nodes in the hidden layer was selected using Shannon's information criterion. An initial phase consisted of choosing the training parameters listed in Table 1.

Table 1. Network for Predicting the Power(%)
Level in EBR-II

Input signals:	Control rod position (inch)
	Core-exit temperature (F)
	IHX secondary sodium outlet temperature (F)

Number of training patterns = 150
Overall prediction error standard deviation = 0.40
Overall modeling error, % = 0.53
Threshold shaping parameter, β = 0.3
Weight update coefficient, α = 0.7
Momentum term coefficient, μ = 0.9
Number of hidden layer nodes = 21

The network shows a high fidelity of signal prediction for a wide range of power variation (45-100%).

5. PLANT-WIDE MONITORING OF EBR-II PROCESS VARIABLES

The application of neural networks presented here is concerned with the prediction of several plant variables as network output. 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 2 and 3 provide details of the two networks used for plant-wide monitoring. Figure 4 shows a comparison of the measured and predicted values of primary variables (Table 2). 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 3) was developed using 150 data points between 45% and 100% power. Figure 5 illustrates the usefulness of the network for detecting the deviation in a signal (IHX secondary sodium outlet temperature) 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.

Table 2. Network for Monitoring Primary Variables in EBR-II

Power level (%)
Core-exit temperature (F)
Control rod position (inch)
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)

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the

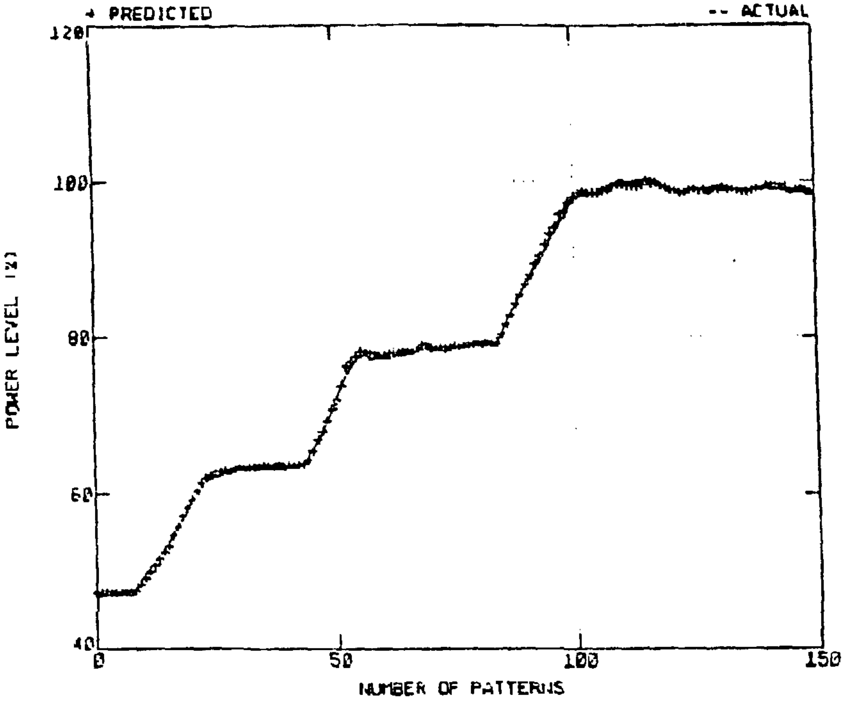

Figure 3. Comparison of measured and network-predicted values of reactor power (%) in EBR-II.

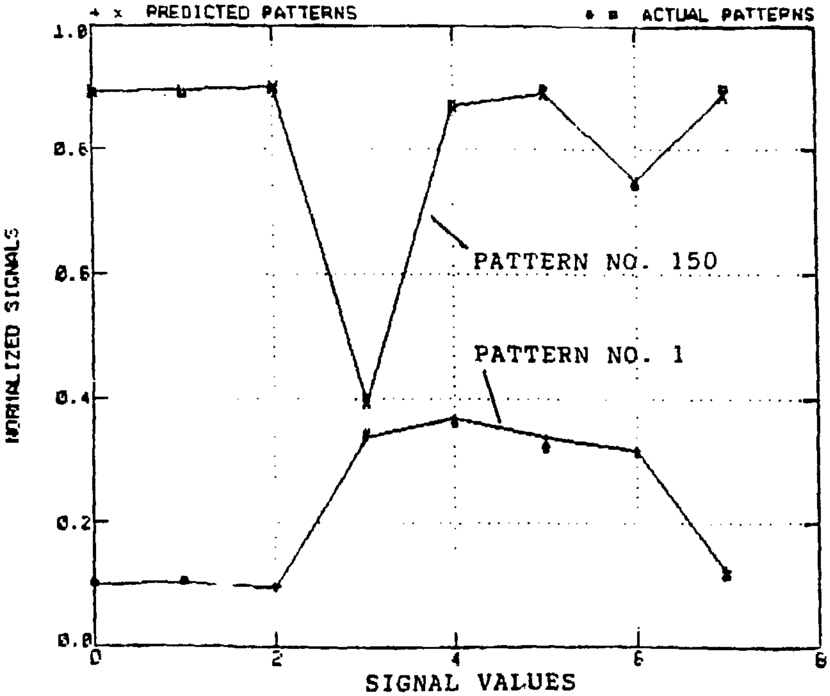

Figure 4. System-wide monitoring of primary side signals in EBR-II (Table 2). Comparison of measured and predicted values at two time instants.

Table 3. 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 (inch)
Steam drum pressure (lb/in^2)
Steam drum feedwater flow rate (lbm/hr)

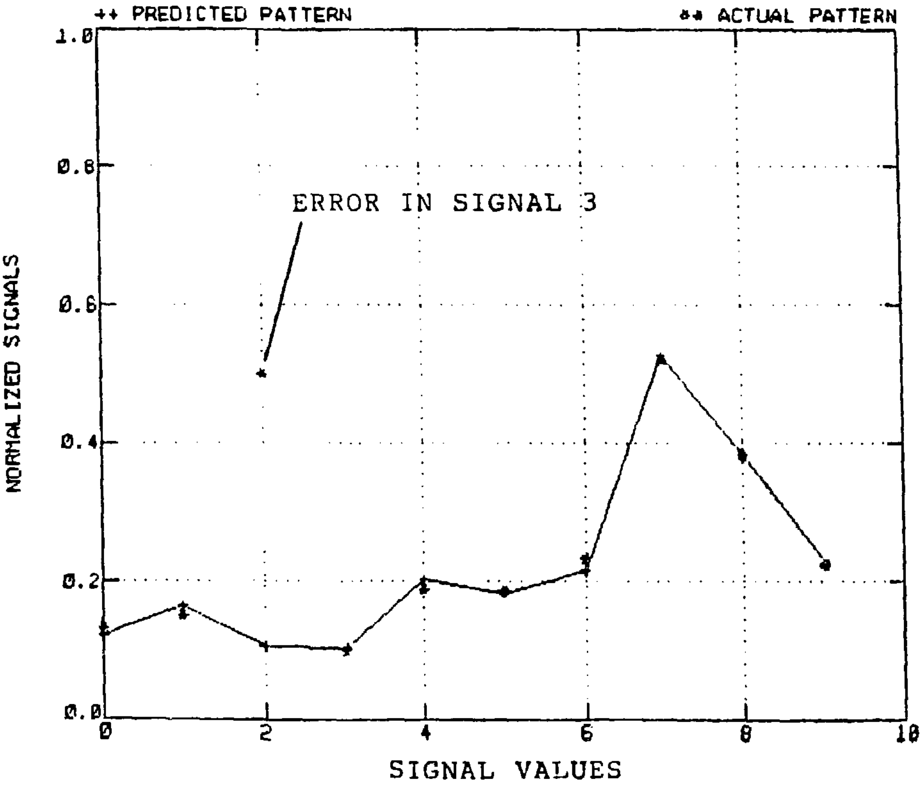

Figure 5. System-wide monitoring of secondary side signals in EBR-II (Table 3). Comparison of measured and predicted values of ten signals. An error introduced in signal 3 is correctly detected as indicated by its estimated value.

6. 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 plant-wide monitoring of signals in EBR-II. An adaptive backpropagation algorithm has been used for the training of a three-layer network. 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 in establishing guidelines for optimal network generation for a given problem.

ACKNOWLEDGMENT

The research reported here is sponsored by a grant from the U.S. Department of Energy under contract number DE-FG07-88ER12824.

REFERENCES

1. B.R. UPADHYAYA et al., "Development and Testing of an Integrating Signal Validation System for Nuclear Power Plants," Vols. 1, 2, 3; Final Report prepared for the U.S. Department of Energy, by the University of Tennessee, DOE/NE/37959-34, 35, 36, September 1989.

2. E. ERYUREK and B.R. UPADHYAYA, "Sensor Validation for Power Plants Using Adaptive Backpropagation Neural Network," Proc. 1989 IEEE Nuclear Sciences Symposium, San Francisco.

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

4. R.P. LIPPMANN, "An Introduction to Computing With Neural Nets," IEEE ASSP Magazine, Vol. 4, No. 2, pp 4-22, April 1987.

5. K.S. NARENDRA and K. PARTHASARATHY, "Identification and Control of Dynamical Systems Using Neural Networks," IEEE Trans. on Neural Networks, Vol. 1, No. 1, pp 4-27, March 1990.