Conf-920471--5

DISCLAIMER

CONF-920471--5

DE93 003551

USING MODULAR NEURAL NETWORKS TO MONITOR ACCIDENT CONDITIONS IN NUCLEAR POWER PLANTS

Zhichao Guo*

Department of Nuclear Engineering, University of Tennessee, Knoxville, TN 37996-2300

and

Robert E. Uhrig

Department of Nuclear Engineering, University of Tennessee, Knoxville, TN 37996-2300 USA
and
Instrumentation and Control Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6005

*Now with Corporate Advanced Systems, Carrier Corporation, Syracuse, NY 13221

Proceedings
of the
SPIE Technical Symposium on
Intelligent Information Systems
Application of Artificial Neural Networks, III
Orlando, Florida
April 20-24, 1992

FG07-88ER12824

Using modular neural networks to monitor accident conditions in nuclear power plants

Zhichao Guo[1] and Robert E. Uhrig

Nuclear Engineering Department, The University of Tennessee
315 Pasqua Engineering Building, Knoxville TN 37996-2300

ABSTRACT

Nuclear power plants are very complex systems. The diagnoses of transients or accident conditions is very difficult because a large amount of information, which is often noisy, or intermittent, or even incomplete, need to be processed in real time. To demonstrate their potential application to nuclear power plants , neural networks are used to monitor the accident scenarios simulated by the training simulator of *TVA's Watts Bar Nuclear Power Plant.* A *self-organization* network is used to compress original data to reduce the total number of training patterns. Different accident scenarios are closely related to different key parameters which distinguish one accident scenario from another. Therefore, the accident scenarios can be monitored by a set of small size neural networks, called modular networks, each one of which monitors only one assigned accident scenario, to obtain fast training and recall. Sensitivity analysis is applied to select proper input variables for modular networks.

1. INTRODUCTION

A nuclear power plant is a very complex system with hundreds of subsystems integrated together to perform various functions. The operating conditions of a nuclear power plant need to be monitored to insure that the plant is operated safely and efficiently. Diagnoses of potentially unsafe plant conditions should be quick and accurate and the corrective actions must be taken as soon as possible. The diagnoses of transient or accident conditions in a nuclear power plant are sometimes difficult because the information obtained from many different sensors in the plant may be noisy, intermittent, or even incomplete.

Neural networks have some remarkable features, such as error tolerance and non-algorithmic simulation and may be used to improve the diagnostic systems. The error tolerance makes the networks suitable for processing noisy and incomplete data, and the non-algorithmic simulation makes it possible to model some complex systems, where only the data of system inputs and outputs are available. In addition, the neural computation when implemented in hardware is essentially parallel, which gives neural networks the power to process a large number of signals in real time.

Uhrig[1] reviewed the various applications of neural networks to nuclear power plants, including fault diagnostics, reactor control, sensor validation, plant status monitoring, design of nuclear fuel cycle reload, licensee event report data base access, and analysis of vibrations. All these applications show that neural

[1]Current Address: Corporate Advanced Systems, Carrier Corporation P. O. Box 4808 Carrier Parkway, Syracuse NY 13221

networks offer an interesting and productive means of addressing many of the problems that occur in nuclear power plants and have the potential to enhance the performance and safety of nuclear power plants in a cost-effective way.

Because many fault conditions which need to be recognized have never actually occurred, the data from the training simulator for *TVA's Watts Bar Nuclear Power Plant* are used in this study.

2. SIMULATED ACCIDENT SCENARIOS

The data obtained from *TVA's Watts Bar* simulator covers seven accident scenarios listed in Table I.

Table I: Accident Scenarios Simulated on Watts Bar Simulator

1	Total loss of offsite power (ED1)
2	Main feedwater line break (F23)
3	Main steam line break (MS1)
4	Control rod ejection (RD6)
5	Hot leg loss of coolant accident (TH1)
6	Cold leg loss of coolant accident (TH2)
7	Steam generator tube leak (TH5)

The data set for each scenario contains 81 plant process variables sampled at intervals of 0.5 or 1.0 second and the total sampling time for each scenario is 200 to 500 seconds. Within each data set the accident condition is preceded by a period of normal full power operation. The data is normalized so that a value of one is equivalent to 100% of a variable's meter reading in the control room. Thus, all variables have values between zero and one.

3. USING SELF-ORGANIZATION TO COMPRESS DATA

Each raw data file has about 500 data points for each of 81 variables. The training patterns are formed by the data values of selected variables at each time step. Therefore, there will be about 500 patterns for each accident scenario, and about 3,500 patterns for the total of 7 accident scenarios. It is difficult to train a neural network with such a large number of training patterns. In the previous study[2], the patterns were averaged over some time intervals to reduce the total number of training patterns. A disadvantage of this approach is that the temporal information is lost, especially for the rapid transients. The *self-organization* network[3] can be used to compress the data and reduce the number of training patterns without substantial loss of information. If data patterns are compressed by the *self-organization* network through clustering, patterns gathered in one cluster will have the same or very similar characteristics defined by the Euclidean distance, and this group of patterns can be represented by its centroid. The advantage of using *self-organization* is that the temporal information is retained after the data compression. One cluster may contain tens of or even hundreds of data patterns if they are similar, such as the data patterns in the time interval of full power operating condition and in the equilibrium stage of the transients. On the other hand, one cluster may only contain one pattern as in the case of rapid transients. By using the centroids of the formed clusters as the training patterns for the *lateral feedback* neural network, the total number of data patterns has been greatly reduced from 3,354 to 138.

4. LATERAL-FEEDBACK NEURAL NETWORK

The network used in this study is a modified version of *backpropagation*[4]. The modification is made by introducing intra-layer connections to neurons in the hidden layer(s). Therefore, it is called *lateral feedback neural network*. Figure 1 gives the schematic drawing of a three layer *lateral feedback* network with intra-layer feedback only in the hidden layer. The intra-layer connections are introduced only to a hidden neuron's closest neighbors. The computer code for implementation of *lateral feedback* network is designed in a 2-D form, therefore, a neuron's neighborhood is defined as the closet neurons surrounding it. It has been shown that the intra-layer or lateral connections can provide contrast enhancement of signals during encoding and recall[5]. It may also increase the stability and convergence of the system if we consider it as a feedback system. The learning algorithms are developed for the *lateral-feedback* network by applying the *Generalized Delta Rule* to the network's special configuration[6]. All connections including intra-layer connections and bias terms can be adapted during the learning. The network architecture, its learning algorithms, and benchmark tests are discussed in details in a Ph.D dissertation[6].

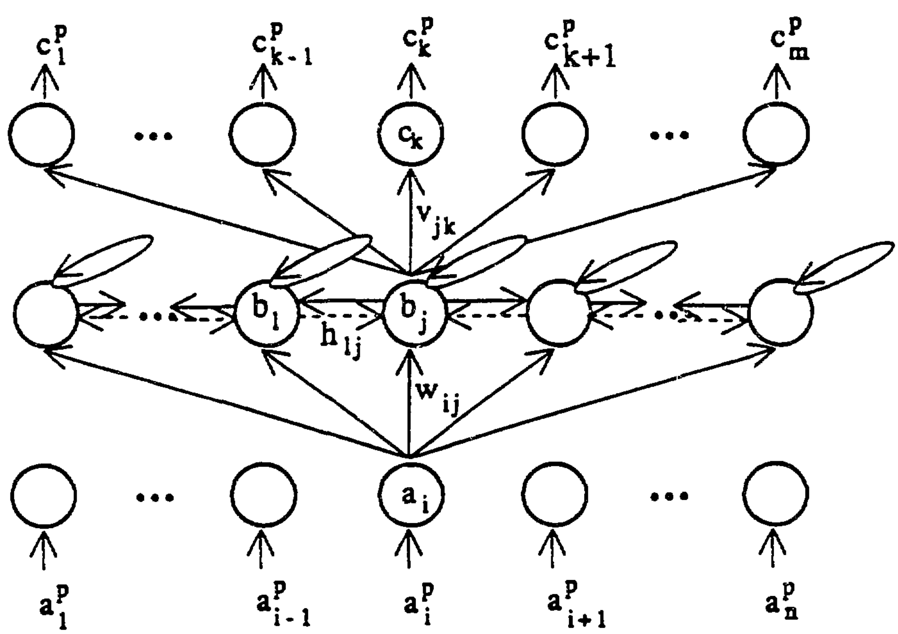

Figure 1: Three layer network architecture with intra-layer connections in the hidden layer.

5. SENSITIVITY ANALYSIS

For a continuous function, its derivatives provide very useful information to characterize the function itself. Sensitivity is usually provided by the first derivative of a function, because it is the ratio of a change in the function's value caused by a change in an independent variable. The larger the absolute value of the first derivative is , the more sensitive the function is with respect to the variable. The sign of the first derivative indicates whether these changes are in the same or opposite directions. The sensitivity analysis of a trained network is the analog of that for an ordinary function. A successfully trained neural network maps an input vector $\vec{X}$ from a n-dimensional space to an output vector $\vec{Y}$ in a m-dimensional space. It can be expressed as:

$$\vec{Y} = f(\vec{X}) \tag{1}$$

where $\vec{Y} = (y_1, y_2, \ldots, y_m)$, and $\vec{X} = (x_1, x_2, \ldots, x_n)$.

The first derivative, $\partial y_j / \partial x_i$, measures the change in y_j while x_i is changing. Hence, it can provide information about how sensitive the output y_j is with respect to the input x_i. The derivative, $\partial y_j / \partial x_i$, can be found through the structure of a trained neural network. The sensitivity analysis has been developed for the *lateral-feedback* neural network, which derives the equation of $\partial y_j / \partial x_i$ through the network's architecture and learning algorithms. The final equation for sensitivity analysis developed for *lateral feedback* network is given below and discussed in more detail in the Ph.D dissertation[6]. Equation 2 shows that the partial derivative, $\partial y_k / \partial x_i$, depends not only on the information learned by the network, which is stored distributively in the connections of w_{ij}, v_{jk}, and h_{jl}, but also on the activation of neurons in both the hidden layer and the output layer, which, in turn, depends on the inputs of the network. For a trained network, all inter- and intra-connections, w_{ij}, v_{jk}, and h_{jl}, are fixed, therefore, the derivative is a function of input patterns only. If the absolute value of $\partial y_k / \partial x_i$ is averaged over all input patterns, it will give the measure of the sensitivity of the output variable y_k with respect to the input variable x_i on a global or average sense. Therefore, all input variables can be ranked in the order of sensitivity (or importance) according the values of the first derivative.

$$\frac{\partial y_k}{\partial x_i} = \beta^3 y_k(1 - y_k) \sum_j v_{kj} b_j (1 - b_j) [\frac{w_{ji}}{\beta} - \sum_{l, l \neq j} h_{jl} b_l^* (1 - b_l^*) w_{li} + h_{jj} b_j^* (1 - b_j^*) w_{ji}] \tag{2}$$

where β is the sigmoid slope; y_k and b_j are the activations of kth neuron in the output layer and jth neuron in the hidden layer, respectively; b_j^* is the intermittent activation of jth neuron in the hidden layer used to generate intra-layer feedback signals; w_{ij} and v_{jk} are the inter-layer connections between the input layer and hidden layer and between hidden layer and output layer, respectively; h_{jl} is the intra-layer connection from lth neuron to jth neuron within the hidden layer; and l is the label for all nearest neighbors of the jth hidden neuron.

6. MODULAR NEURAL NETWORKS

Different variables may play different roles in different accident scenarios and only a few key variables are necessary for a network to distinguish one accident from another. If a single network is designed to monitor all possible transients or accident scenarios, it must have a large input layer because the important variables of each and all accident scenarios have to be considered as the input variables. One or more large hidden layers and a large output layer is also required to make it possible for the network to distinguish all scenarios. It will be very difficult to train some large neural networks because the limitations of learning ability, convergence , and stability in many frequently used neural networks. One alternative is to build a set of small-scale neural networks, called *modular networks*, each one of which responds to only one accident scenario. The size of the modular networks can be very small because only the key variables for that particular accident are used as the input variables for the corresponding modular network. There will be only one neuron in the output layer whose output will be 1 or 0. 1 indicates that the accident has occurred and 0 indicates that it has not occurred. The concept of modular network will be very useful in the application to nuclear power plants where a large number of parameters and potential accidents need to be monitored.

7. SELECTING INPUTS FOR MODULAR NETWORKS BY SENSITIVITY ANALYSIS

How to select proper input variables for different modular networks is crucially important for building up modular neural networks. We need to select only these variables which are necessary in providing important information to modular networks for monitoring and distinguishing different accident scenarios. The sensitivity analysis can be used to guide the selection of input variables for modular networks because it can tell if a variable is important for a particular accident scenario. Before the sensitivity analysis can be applied, we need to have a trained neural network which monitors all accident scenarios and uses all provided variables as its inputs. Because all modular networks are built based on the information extracted from this trained network through sensitivity analysis, we call it motherhood neural network. If we can train a motherhood neural network to monitor all accident scenarios why bother with modular networks? We may answer this question through different aspects. First, a modular network uses only the key variables as its inputs to monitor an assigned accident scenario and has only one neuron in the output layer and a small hidden layer, therefore, it needs much less computation time for recall, hence, can respond faster than a big size motherhood network. Second, a set of modular networks can be operated in parallel, hence, to improve the fault tolerance of the neural network diagnostic system because one damaged modular network may only affect its own performance. For the motherhood network, its diagnostic performance will degrade to all monitored accident scenarios if there is some damage to the network. Finally, it will be easier for hardware implementation of a modular network because of its simple structure and small size. Therefore, a set of modular networks can be used to replace a large motherhood network to improve the neural network diagnostic system.

8. PREPARATION OF DATA FOR NEURAL NETWORKS

The data of each simulated accident scenario is stored in one data file called raw data file, such as ed1.dat which is the data file for the accident scenario of *Total Loss Of Offsite Power*. 22 variables were selected as the input variables for the motherhood network and are listed in Table II. The input variables for all modular networks would also be selected from those 22 variables by sensitivity analysis.

Table II: Variables Selected as Inputs for the Network

Code	Plant Process Variable
0	Nuclear Power Level
1	Unit Megawattage Output
2	Volume Control Tank Level
3	RB FL & Equip Drain Level
4	Containment Pressure
5	P. R. % Full Power AO Array
6	SG #1 Steam Flow Channel 1 Meter
7	SG #1 Main Steam HDR Pressure
8	SG #1 Feedwater Inlet Flow
9	SG #1 Feedwater Supply Pressure
10	SG #1 Aux. Feedwater Inlet Flow
11	SG #1 Level Indicator
12	SG #1 Wide Level Indicator
13	RCS Pressurizer Level
14	RCS Pressurizer Pressure
15	RCS Pressurizer Surge Line Temp.
16	RCS Loop 1 Spray Temp.
17	RCS Loop 1 Hot-leg Pressure
18	RCS Loop 1 Cold-leg Temp.
19	RCS Loop 1 Hot-leg Temp.
20	RCS Loop 1 Average Temp.
21	RCS Loop 1 Coolant Flow

The data extracted from each raw data file are stored as two separate files. One contains the data under normal full power operation, and the other contains the data under transient condition, because the full power data and the transient data need to be compressed by *self-organization* separately. All normal full power data files are concatenated together to form one data file. Therefore, the total data files are eight including one for normal full power operation and seven for the accident scenarios. These 8 data files will be used as the training data files for *self-organization* to create clusters and as the recall data files for the *lateral-feedback* to insure the training success. The data files listed in Table III are used for the training of *self-organization* and the results of the training are listed in Table IV. It can be seen from Table III that each accident scenario has been assigned a binary code and only one bit is turned on as 1 and the rest are turned off as 0. The design of this binary coding is to let each neuron in the output layer of the motherhood network respond to only one accident scenario, hence, the sensitivity analysis can extract information about the important variables for each accident scenario.

The centroids of all clusters listed in Table IV are used as the training patterns for the *lateral feedback* network. The total training patterns are 138. It can be seen that by using the *self-organization* the training data patterns can be greatly reduced for the further supervised training. In this application the training patterns are reduced from the original 3,354 to 138 through training by the *self-organization* neural network.

Table III: Binary Representation of 8 Accident Scenarios

Plant Condition	Binary Code	Data File
Normal Full Power Operation	0 0 0 0 0 0 0	full.pat
Total loss of offsite power	1 0 0 0 0 0 0	ed1t.pat
Main feedwater line break	0 1 0 0 0 0 0	f23t.pat
Main steam line break	0 0 1 0 0 0 0	ms1t.pat
Control rod ejection	0 0 0 1 0 0 0	rd6t.pat
Hot leg loss of coolant	0 0 0 0 1 0 0	th1t.pat
Cold leg loss of coolant	0 0 0 0 0 1 0	th2t.pat
Steam Generator tube leak	0 0 0 0 0 0 1	th5t.pat

Table IV: Clusters Formed for Each Data File

Data File	Patterns	Clusters	Eu-Distance
full.pat	255	2	0.05
ed1t.pat	459	17	0.10
f23t.pat	396	21	0.10
ms1t.pat	396	19	0.10
rd6t.pat	440	18	0.10
th1t.pat	469	25	0.10
th2t.pat	471	23	0.10
th5t.pat	468	13	0.10
Total #	3354	138	

9. TRAINING AND RECALL OF MOTHERHOOD NETWORK

The motherhood network set up for this application had 3 layers. The input layer had 22 neurons to match the 22 selected variables. The hidden layer had 16 neurons. And the output layer contained 7 neurons to have each neuron response only one of the seven accident scenarios. The network was trained for 4,500 cycles and converged with maximum error less than 0.05%, which is the relative error of a network output with respect to the corresponding target value. All 8 data files with extension *pat* listed in Table IV were used as the data files for recall. Figure 2 shows the network response of the motherhood network to the accident scenario of *Total Loss Of Offsite Power*.

The network responses for all other accident scenarios are listed in Table V. It can be seen from the table that the network can determine all the scenarios at the very beginning of the transients. For slow transients, control rod ejection and steam generator tube break, it takes longer for the network to determine the transients but the time for the diagnoses is still much shorter than the corresponding reactor scram time even though the data used for recall was not used directly for training of the motherhood network but for training of the *self-organization* to form the clusters listed in Table IV. Therefore, the training of the motherhood network is successful. The quick and accurate responses to all the scenarios indicates that the combination of *self-organization* and a supervised learning network, i.e. *lateral-feedback*, is suitable for the training tasks where a large amount of data need to be processed.

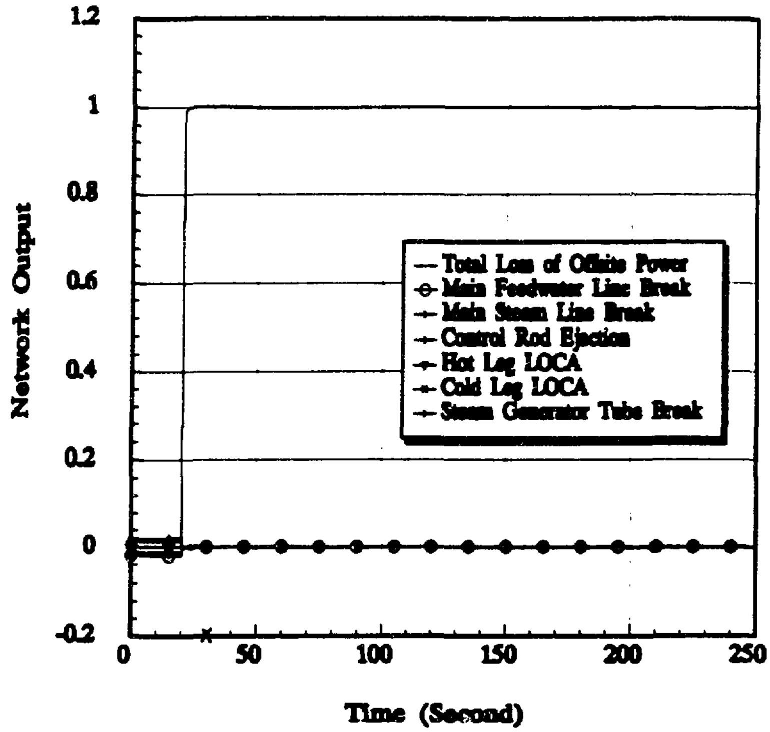

Figure 2: Outputs From Node 1 of Motherhood Network Which Responses to Total Loss of Offsite Power

Table V: Time to for the Network to Diagnose Transients

Plant Condition	Diag. time(sec)	Scram Time (sec)
Total loss of offsite power	0.5	9.0
Main feedwater line break	0.5	15.5
Main steam line break	0.5	11.0
Control rod ejection	10.0	44.0
Hot leg loss of coolant	0.5	4.0
Cold leg loss of coolant	0.5	4.5
Steam Generator tube leak	90.0	379.5

10. INPUTS SELECTED FOR MODULAR NETWORKS

To selected the key variables as inputs for each modular network, the sensitivity analysis was applied to the successfully trained motherhood network. The accurate and fast response to all 7 accident scenarios insured that the previously trained motherhood network could be considered as the mapping function, which could, then, be studied further by the sensitivity analysis. The importance of an input variable is measured by the sum of the absolute values of $\partial y_k/\partial x_i$ over all input patterns, which gives the measure of importance of an input x_i with respect to an output y_k on a global or average sense. The results of the sensitivity study ranked the 22 input variables in the order of importance for each output node which monitors one of the seven accident scenarios, therefore, the input variables for each modular network could be selected. The selection of important variables for an accident scenario was based on the relative values of $\|\sum_p \partial y_k/\partial x_i\|$ over all patterns for each i and a fixed k. If there was a sharp decrease in the value of $\|\sum \partial y_k/\partial x_i\|$, i.e. at jth position of ranked variable list, then, the variables ranked higher than jth position were selected as the important variables for the accident. The variables selected for all seven modular networks are listed in Table VI.

Table VI: Variables Selected for Each Modular Network

Modular Network for	Input Variables Selected	# Variables
Total loss of offsite power	3 1 21 14 9	5
Main steam line break	6 7 12 9 13 8 1 14 21	9
Main feedwater line break	13 14 21 11 9 3 6	7
Control rod ejection	13 21 3 14 1 15 4 9	8
Hot leg loss of coolant	21 4 13 11 17 7 9	7
Cold leg loss of coolant	13 21 4 11 14 17 9	7
Steam Generator tube leak	13 21 14 3 9	5

The physical meaning of each selected variable can be retrieved from Table II. For *Hot-leg LOCA* and *Cold-leg LOCA*, for example, the first three important variables are selected as 4, 13, and 21, which are, in turn, *Containment Pressure, RCS Pressurizer Level*, and *RCS Coolant Flow.* These three variables respond rapidly during the accidents and can be considered as the key parameters for the networks to determine what actually happened in the system. Because the plant process variables give similar responses for *Hot-leg LOCA* and *Cold-leg LOCA*, the key variables used to detect the accidents should be identical or similar. The sensitivity study has drawn the same conclusion and selected the virtually the same variables as the important variables for both *Hot-leg LOCA* and *Cold-leg LOCA*. It shows that the sensitivity analysis can provide useful information about the important variables for determining the system status.

11. CONFIGURATION, TRAINING AND RECALL DATA OF MODULAR NETWORKS

The paradigm of neural network used for all modular networks was *lateral feedback*. All modular networks had three layers. The neurons in the input layer were determined by the number of input variables selected by the sensitivity analysis that are listed in Table VI. There were 4 neurons in the hidden layer for all modular networks, and 1 neuron in the output layer to respond the assigned accident scenario. The output value of 1.0 (or close to 1.0) indicates that the assigned accident has been detected and a value of 0.0 (or close to 0.0) indicates that an accident has not occur. The training data file for all modular networks was the data file formed by the clusters listed in Table IV. The target value of 1.0 was assigned to the patterns (centroids of clusters) associated with the accident scenario monitored by the modular network, and 0.0 to the rest of the patterns. One pattern could be assigned a target value of 1.0 or 0.0 depending on which scenario is monitored by the modular networks. The recall data files used for all modular networks are all files (with pat as extension name) listed in Table IV. These data files have not been used for training of any modular network but for training of the *self-organization.*

12. WEIGHING INPUTS ACCORDING TO THEIR RELATIVE IMPORTANCE

Variables were selected for each modular network according to their importance to the accident scenario. It is logical to think that a variable with higher rank should play more important role in affecting a network performance. Therefore, the selected variables were weighed according to their rank in importance determined by $\sum \|dy/dx\|$. The approach used for weighing the inputs for this study is the method of amplitude modulation which changes the low and upper bounds of normalization for an input variable to be proportional to its $\sum \|dy/dx\|$ values, or importance. The most important variable, for example, will be normalized into range of $[-2,2]$ while the least important variable will be normalized into range of $[-1,1]$. This weighing method is to differentiate values between the minimum and maximum values of an input. For more important variables this method makes a small difference large, i.e. increasing the relative amplitude of a small peak in the signal, and making it easy for the network to detect. This weighing method was applied to all modular networks for their training.

13. PERFORMANCE OF DIAGNOSTICS OF MODULAR NETWORKS

All modular networks were trained for 1,000 training cycles with the training file formed by centroids of clusters. The recall was performed with all seven modular networks by using all 8 data files, listed in Table IV, corresponding to 7 accident scenarios and one full power operation. There is only one neuron in the output layer for all modular networks. The correct response for a modular network is that its output is close to 1.0 for the assigned accident and to 0.0 for the rest. All seven modular networks can give quick and correct responses to their own assigned accident scenarios. The diagnosing time required for the modular networks to detect the corresponding transients are summarized in Table VII.

Table VII: Time required for modular networks to diagnose the transients

Code	Plant Condition	Diag. Time	Scram Time
ED1	Total loss of offsite power	0.5	9.0
MS1	Main steam line break	0.5	11.0
F23	Main feedwater line break	1.0	15.5
RD6	Control rod ejection	10.0	44.0
TH1	Hot leg loss of coolant	0.5	4.0
TH2	Cold leg loss of coolant	0.5	4.5
TH5	Steam Generator tube leak	80.0	379.5

It can be seen from Table VII that the diagnosing time is shorter than the corresponding scram time for all 7 scenarios. Figure 3 shows the recall results of the modular network which monitors the accident scenario of *Total Loss Of Offsite Power*.

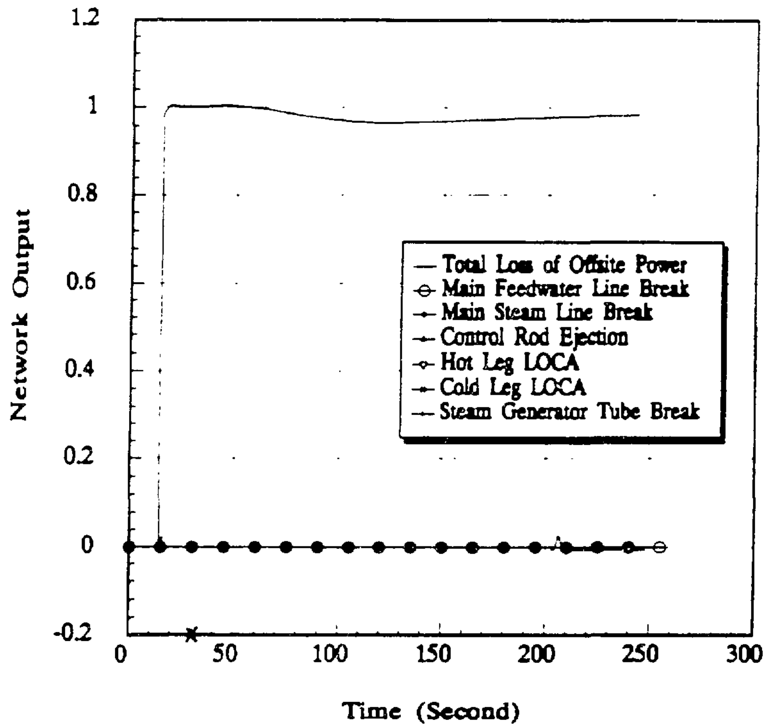

Figure 3: Response to All 7 Scenarios From Modular Network of Total Loss of Offsite Power (5 Weighted Inputs Selected for the Network)

14. CONCLUSION

The diagnostics of nuclear power plant transients was performed using modular networks. Because there are usually a large number of plant process variables and accident scenarios that need to be monitored to ensure the plant safety, it is very difficult to use a single neural network to monitor an entire plant system. Therefore, building up modular networks to monitor the plant system is necessary. Besides, modular networks have simple configuration and small size, and can be operated in parallel, hence, they can be used to build up a more reliable and efficient neural network diagnostic system. The crucial point to success in building up the modular networks is to choose the proper inputs for different modular networks to reach simple network structure, fast training, and accurate recall. The sensitivity analysis can be used to guide the selection of input variables for modular networks and to emphasize the relative importance among the selected variables by weighing them through modulating boundaries of normalization. The methodology of using *self-organization* to compress data was also demonstrated in this study. The total training patterns can be greatly reduced if the centroids of clusters formed by *self-organization* are used as the training data for a following network, such as a *backpropagation* network.

15. REFERENCES

1. Uhrig, R. E.,"Potential Application of Neural Networks to the Operation of Nuclear Power Plants," Nuclear Safety: vol. 32 no. 1, pp.68-79, 1991.

2. Bartlett E. B. and Uhrig, R. E.,"Nuclear Power Plant Status Diagnostics Using An Artificial Neural Network," Nuclear Technology, vol. 97, pp. 272-281, March 1992.

3. Kohonen, T., *Self-Organization and Associative Memory*, New York: Springer-Verlag, 1984.

4. Rumelhart D., McClelland J. L., and PDP Research Group, *Parallel Distributed Process*, MIT Press, Cambridge, Massachusetts, 1989.

5. Choen, M. and Grossberg, S.,"Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," IEEE Transactions on System, Man, and Cybernetics, SMC-13, pp. 815-825, 1983.

6. Guo, Z.,"Nuclear Power Plant Fault Diagnostics and Thermal Performance Studies Using Neural Networks and Genetic Algorithms," A Ph.D Dissertation, Department of Nuclear Engineering, The University of Tennseess, 1992.