Monograph:
Jalel, N.A. and Nicholson, H. (1990) The Application of Neural Networks for Fault Diagnosis in Nuclear Reactors. Research Report. Acse Report 411 . Dept of Automatic Control and System Engineering. University of Sheffield

The Application of Neural Networks for Fault Diagnosis in Nuclear Reactors

by

N. A. Jalel and H. Nicholson

Department of Control Engineering
University of Sheffield
Mappin Street
Sheffield S1 3JD

Research Report No. 411
November 1990

1 Abstract

In recent years considerable work have been done in the field of neural networks due to the recent development of effective learning algorithms, and the results of their applications have suggested that they can provide useful tools for solving practical problems. Artificial neural networks are mathematical models of theorized mind and brain activity. They are aimed to explore and reproduce human information processing tasks such as speech, vision, knowledge processing and control. The possibility of using artificial neural networks for fault and accident diagnosis in the Loss Of Fluid Test (LOFT) reactor, a small scale pressurised water reactor, is examined and explained in the paper.

2 Introduction

Artificial neural networks, which are referred to as neural networks, connections, adaptive networks, neurocomputers and parallel distribution processors, are massively parallel interconnected networks of simple elements intended to interact with the real world in the same way as biological nervous systems.

They have been used to solve a wide variety of science and engineering problems that involve extracting useful information from complex or uncertain data [1,2]. The back propagation algorithm has been used effectively to control a space lander simulation where the space craft should be landed before the fuel runs out and with as little velocity as possible [3]. Another application of the back propagation neural net is for the dynamic modelling and control of chemical process systems where it has been applied to model the dynamic response of pH in a stirred tank reactor [4]. In the field of fault detection and diagnosis, the neural network approach has been used to diagnose faults in a three continuous-stirred tank reactor using a multi-layer feedforward perception model [5]. Neural networks have also been used as a connectionist expert system knowledge base for medical diagnosis [6,7]. In the field of nuclear reactors, neural networks have been used with the back propagation algorithm applied to identify four different abnormal behaviour patterns in the response of a simulated steam generator [8].

In this paper, the concept of artificial neural networks will be explained, two different approaches for their use for fault diagnosis in the LOFT reactor [15] will be illustrated, and finally the two techniques will be examined and discussed through a hypothetical accident.

3 Artificial Neural Network Model

Neural networks use the massively parallel-distributed processing potential of computational hardware and are aimed at developing information processing that is analogous to the biological nervous system [9,2,10].

The main unit of an artificial neural network is the processing element. They are referred to as nodes, neurons or threshold logic units. As shown in figure 1, the processing element is a multi-input single-output unit and consists of:

1. Inputs
   They can come from either sources external to the neural network or outputs of other processing elements, including itself, and form an input vector $I_i, i = 1, 2, \ldots, n$.

2. Weights
   Within each connected pair of processing elements is an adjustable numerical value called a weight, denoted by W_{ij}, which roughly represents the connection strength from the processing element i to the processing element j and determines how much influence an input has on the processing element.

3. Combining function
   This relates the weights W_{ij}, their associated processing element input values I_i and the internal threshold value Θ_j which must be exceeded for there to be any processing element activation. The combining function is performed by taking the dot product of I_i and W_{ij}, adding the threshold and passing the result through a threshold function f ( ). It is defined as:

$$Y = f\left(\sum_{i=1}^{n} W_{ij} I_i + \Theta_j\right)$$

4. Threshold function
   The threshold function, also referred to as an activation function, transfer function or signal function, interprets the result of combining functions and determines the output which could be connected to any other processing element including itself, or can be output to external sources. The most common types of threshold function are the linear, ramp, step and sigmoid functions.

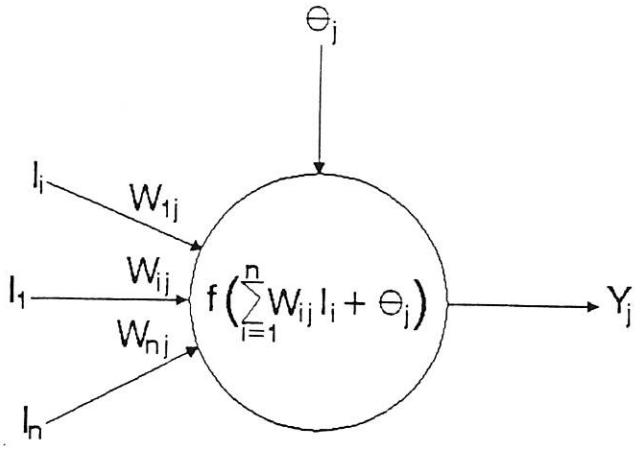

Figure 1: The topology of a processing element

Processing elements within a neural network are grouped together to form a structure called a layer, hence neural networks emerge from the interconnection of one or more layers of processing elements. A typical neural network usually consists of one or more intermediate or hidden layers of processing elements which are responsible for additional information processing with the input and output layer. The input layer receives input from the external world while the processing elements in the output layer have outputs that are taken to be the outputs of the network as a whole and delivers the representation of the input after processing has occurred. The hidden layers usually contain hidden units or processing elements that are not directly connected to both the input and output processing elements.

Networks are classified as either feedforward networks if the output of the processing element in a layer can flow to the processing element in the following layer, lateral feedback if it flows to an element in the same layer or feedback if it flows to itself.

Neural networks perform by receiving inputs from the external world at their input layer, processing the input through its layer and producing a pattern of activation at its output layer, where each individual processing element functions independently and in parallel with other processing elements.

The intelligence of a network is contained in the shape of connections between processing elements and the strength of those connections, which is achieved through the learning laws associated with each layer. In a neural network model, learning deals with the ability of processing elements to modify and find the weights that will produce the desired behaviour where the learning algorithms usually work from the

training examples or experience.

Back propagation and the threshold logic unit are the learning algorithms adopted in this work and are used for fault diagnosis in the LOFT reactor.

4 The Back Propagation Algorithm

The back propagation algorithm is the most common learning algorithm which has been tested with a number of different problems and has been found to perform well in most cases and to find good solutions. Back propagation is a learning algorithm designed to solve the problem of choosing weight values for a three layer artificial neural network with feedforward connections from the input layer to the hidden layer and then to the output layer. The back propagation algorithm performs the input to output mapping by minimizing a cost function using a gradient search technique, where the cost function, which is equal to the mean square difference between the desired and the actual net output, is minimized by making weight connection adjustments according to the error between the computed and desired output processing element values.

The back propagation involves two stages, the forward stage includes initially selecting small random weights and internal threshold for the processing elements, entering the input values, which are propagated forward by computing through-out the network layers to produce the output value for each unit and comparing the calculated output with the given desired output, resulting in an error term for each output. The backward stage involves a backward pass through the network layers during which the error term is computed for each unit in the layers, and finally the weight connections are adjusted using the calculated error term associated with the unit it projects to and the output of the unit it projects from, and the internal threshold of the processing elements are also adjusted.

The procedure continues until the error term for each output unit is either sufficiently low or zero. Table 1 illustrates the back propagation algorithm [11,12,13].

Back Propagation Algorithm
1. Assign small random values for the weights and thresholds for the network
2. Present the input data and the desired output to the network
3. For each non input unit, the net input to the unit, net_j, is computed $$net_j = \sum_j W_{ij} I_i + \Theta_j$$
4. In the case of a sigmoid function, the output of the unit is calculated as $$Y_j = \frac{1}{1+\exp(-net_j)}$$
5. Compare the outputs of the network with the desired outputs to calculate the error $$(d_j - Y_j)$$
6. For each output node compute the error function δ as $$\delta_j = Y_j (1 - Y_j)(d_j - Y_j)$$
7. For each hidden layer, calculate the δ function as $$\delta_j = \acute{X_j}\left(1 - \acute{X_j}\right) \sum_k \delta_k W_{ik}$$ where $\acute{X_j}$ is the output of the hidden layer
8. Adjust all weights in the network $$W_{ij}(t+1) = W_{ij}(t) + \eta \delta_j \acute{X_j} + \alpha (W_{ij}(t) - W_{ij}(t-1))$$
9. Adjust the threshold of the processing elements $$\Theta_j(t+1) = \Theta_j(t) + \eta \delta_j + \alpha (\Theta_j(t) - \Theta_j(t-1))$$ where η is a small positive constant controlling the learning rate and α is a momentum constant where $0 < \alpha < 1$
10. Repeat step 3 to 8 until convergence

Table 1: The description of the back propagation algorithm

5 Network Design and Structure Using the Back Propagation Algorithm

A neural network technique will be used to assist the operator in diagnosing and identifying any abnormal behaviour in the LOFT reactor, by modelling the effects of the different accidents and faults that might occur. The back propagation algorithm is applied to train the network to identify and diagnose any fault by choosing the appropriate connection weights and internal thresholds for the processing elements.

The inputs for the networks are the symptoms for the different faults and accidents, however most of these symptoms are identified through monitoring the state variables of the LOFT reactor model which is simulated using C language on a Sun workstation. The state variables of the LOFT reactor model are analysed for

any deviation from the normal operating limit which could happen due to increase or decrease in the values of the state variables due to abnormal behaviour in the reactor. For each state variable, two limits have been designed to identify the increase or decrease and further increase or decrease from the normal operating limits. Another set of symptoms which involves information about the whole power plant is obtained and entered to the system by asking the user about their occurrence.

The input values for the network are discrete, taking on values of 1 and 0, where 1 refers to the occurrence of the symptoms which could be a deviation in the value of the state variable, and any increase or decrease is assigned a value equal to 1, or it could be a sensor alarm received by the operator in the control room where the user of the system records the occurrence of the symptom. A zero value indicates that the system is in normal operating condition.

For the network with four processing elements in the input layer, as shown in figure 2, the input of the network is a set of four elements, one element for each node, and each element has a value equal to 1 or 0, thus there are a possibility of 16 sets of data which can be entered to the network ranging from (0 0 0 0) which indicates reactor normal operation to (1 1 1 1) which refers to the occurrence of all the symptoms of the accident.

The desired output of the network ranges between 0 and 1 depending on the state of the input set. When all the elements of the input set are equal to 0 then the desired output should be equal to 0 referring to normal operation and indicating that the percentage of faults occurring is 0 %. When all the elements of the input set are equal to 1, the desired output is given as equal to 1, indicating that the fault has occurred with a percentage possibility equal to 100 %. However, when one element of the set is equal to 1 and the rest are 0 the desired output is given as equal to 0.2 indicating a 20 % possibility of the fault occurring. Thus, in the case of two elements of the set equal to 1, the desired output is equal to 0.4 indicating a 40 % possibility of a fault occurring, while in the case of three elements are equal to 1 the desired output is given to be equal to 0.7 recording that the possibility of a fault occurring is 70 %. Table 2 illustrates all the possibilities of the input data to the network with the desired output value for each possible input set.

The input sets	The desired output
(0 0 0 0)	0
(1 0 0 0)	0.2
⋮	⋮
(0 0 0 1)	0.2
(1 1 0 0)	0.4
⋮	⋮
(0 0 1 1)	0.4
(1 1 1 0)	0.7
⋮	⋮
(0 1 1 1)	0.7
(1 1 1 1)	1.0

Table 2: The input data for four processing elements in the input layer

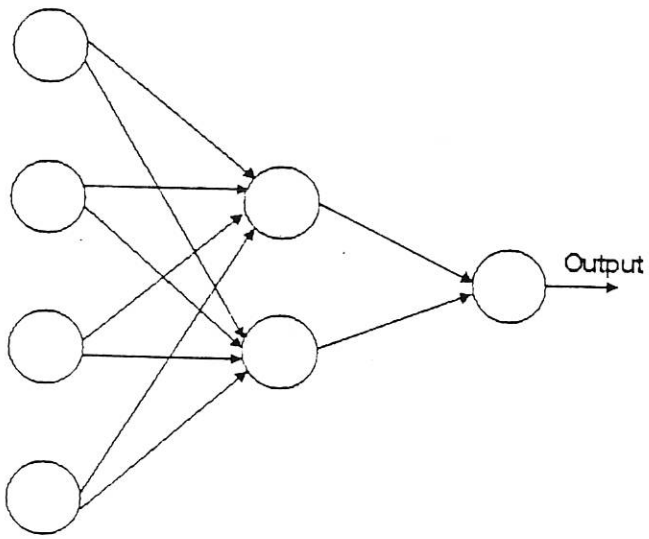

Figure 2: Network with four processing elements in the input layer

6 Applying the Back Propagation Algorithm

The back propagation algorithm illustrated in table 1 is used to train the networks to find the correct output value for each set of input data. The algorithm starts by

assigning small random values for all the weights and the internal thresholds of the processing elements.

After calculating the combining function, the output of the network is calculated by passing it through the sigmoid threshold function which has an output range between 0 and 1. The error between the desired and the calculated output is computed and then the weights and thresholds are updated. The procedure is continued until convergence is achieved. The final set of weights and thresholds for the network with four nodes in the input layer which achieves convergence is given in table 3.

The learning factor is chosen as 0.6 while the momentum constant is equal to 0.9. The back propagation algorithm used to train the networks is coded in C language and run on a Sun workstation.

Weights			Thresholds	
Input to 1st node hidden layer	Input to 2nd node hidden layer	Hidden layer to output layer	Node in hidden layer	Node in output layer
-2.2646	1.4386	-4.283	0.6459	-0.8011
-2.2614	1.4413	7.2953	-5.5387	
-2.2596	1.4432			
-2.2586	1.4434			

Table 3: Weights and thresholds for four processing elements in the input layer

7 Network Performance Using the Back Propagation Algorithm

The computer program simulates the 27th order LOFT reactor model, monitors the state variables to identify any deviation from the normal operating limit, asks the user about the availability of some other information concerning the whole reactor plant and arranges and inputs the complete information to the networks. The output of the program is a list of the percentage possibilities of fault occurrence in the LOFT reactor. The program is designed to simulate the model, analyse the state variables, input the data to the networks and obtain the percentage list for all faults and accidents, at each second interval.

Partial loss of flow due to blockage in a fuel assembly is assumed to occur in

the LOFT reactor to test and examine the ability of the program to diagnose the reactor accident. The hypothetical accident is assumed to be initiated by a drop in the coolant flow from 3.7 (lbm/hr) to 1.6 (lbm/hr) at a time greater than 2.0 sec.

As shown in figure 3, the indication for this accident involves four possible sets of symptoms which are

1. Decrease in coolant flow, change in flow distribution, increase in reactor coolant average temperature and change in temperature distribution.

2. Decrease in coolant flow, change in flow distribution, increase in reactor coolant average temperature and change in power distribution.

3. Decrease in coolant flow, change in flow distribution, increase in temperature difference across reactor core and change in temperature distribution.

4. Decrease in coolant flow, change in flow distribution, increase in temperature difference across reactor core and change in power distribution.

The network with four processing elements in the input layer is used to model this accident where the network is used four times, once for each set of symptoms, resulting in four possible output values in which the highest value will be considered as the network output.

Table 4 illustrates the percentage possibility of all faults and accidents in the reactor at times 0, 3, 4 and 5 sec. At time 0 sec, the maximum network output or fault percentage is 2.8 % revealing that the reactor is in a normal situation. At time 3 sec, the accident has occurred, the faults percentage for the partial loss of flow due to blockage in a fuel assembly has the highest accident percentage which is 19.9 %. At times 4 sec and 5 sec the partial loss of flow also has the highest percentage value where at time 5 sec the percentage value is 96.3 % meaning that the network output is near to the desired output 1, indicating that all the symptoms for the accident have occurred and the network input set is (1 1 1 1).

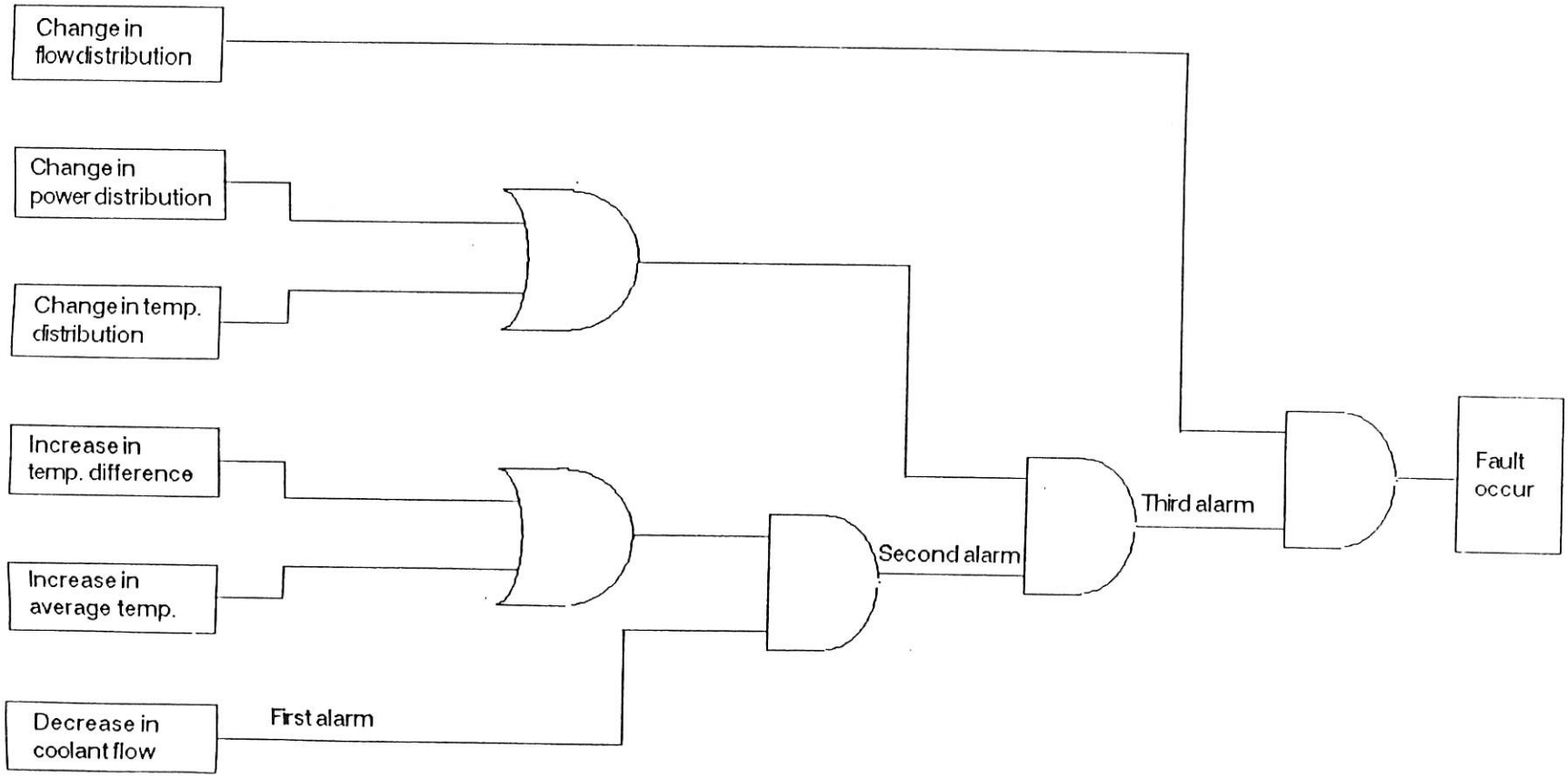

Figure 3: Fault tree for a partial loss of flow due to a blockage in the fuel assembly

Accident type	Percentage failure			
	Time			
	0	3	4	5
Failure in power control system to respond to boron increase	2.8	2.8	31.04	50.07
Failure in power control system to respond to boron decrease	2.8	2.8	2.8	2.8
Failure in the rod drop	2.7	2.7	19.9	19.9
Failure due to misalignment of RCCA (Reactor Control Cluster Assembly)	2.8	2.8	2.8	2.8
Leakage in feedwater piping	2.8	2.8	2.8	2.8
Blockage in feedwater piping	2.8	2.8	9.7	9.7
Excessive feedwater heat removal	1.7	1.7	1.7	29.9
Blockage in fuel assembly	2.7	19.9	70.3	96.3
Loss of coolant	1.7	1.7	1.7	1.7
Failure in the pressuriser heater control	1.7	1.7	1.7	1.7
Failure in the pressuriser spray control	1.7	1.7	1.7	1.7
Failure in the pressuriser water level control	2.7	2.7	2.7	2.7
Steam generator tube rupture	1.7	1.7	1.7	1.7
Steam line break	2.8	2.8	2.8	2.8

Table 4: The fault percentage for the different faults in the reactor at time 0 sec

8 Threshold Logic Unit

A threshold logic unit is referred to a processing element with a step threshold function, so the output of the processing element is discrete, taking on values 1 or 0. These values can be used to solve logic problems where 1 corresponds to a fault condition while 0 corresponds to a normal condition.

The single threshold logic unit first initializes the connection weights and the threshold value to small random values and then computes a weighted sum of the input elements, adds a threshold Θ and passes the result through a step function such that the output Y is either 1 or 0.

The iteration of the network contains several layers of processing elements, which reevaluate each processing element in index order and change their output before the next processing element is reevaluated. One iteration is sufficient to bring the network to the steady state. The threshold logic algorithm is illustrated in table 5 [7,16].

Threshold Logic Algorithm
1. Initialize weight W_i and threshold Θ to small random value
2. Present the input value X_i and the desired output $d(t)$ $d(t) = \begin{cases} 1 & \text{if input in faulty condition} \\ 0 & \text{if input in normal condition} \end{cases}$
3. Calculate the net input to the output unit $net = \sum_i W_i X_i + \Theta$
4. Find the output of the processing element as $Y(t) = \begin{cases} 1 & \text{if } net > 1 \\ 0 & \text{otherwise} \end{cases}$
5. Update all the weights $W_i(t+1) = W_i(t) + \eta [d(t) - Y(t)] X_i(t)$
6. Repeat step 3 to 5 until convergence

Table 5: The description of the threshold logic algorithm

9 Threshold Logic Approach for Fault Diagnosis

New network models will be designed to assist the nuclear reactor operator in identifying the different accidents and alarms that might occur in the reactor. The approach is based on designing a network for each logic tree of the faults and alarms that might happen in the reactor. A threshold logic algorithm is applied to train the

networks where the training algorithm involves calculating the output for the processing elements in the first layer, computing the error by comparing the calculated output with the desired value and adjusting the weight and threshold and changing the output so that the error is 0 before processing to the next layer of processing elements.

The inputs for these networks are discrete values of 1 and 0, where 1 refers to the occurrence of the symptoms of the fault which arises as a result of the deviation in the state variables of the simulated LOFT model, while the 0 refers to the normal operation situation. The inputs for the network are found by monitoring the state variables for any increase or decrease from the normal operating limit and the appearance of any sensor alarms concerning the whole reactor plant.

The output of the network is either 1 or 0 where 1 refers to the occurrence of the fault or the alarm while 0 means normal operation.

10 Applying the Threshold Logic Algorithm

The idea of this approach is based on processing through the network starting from the first layer, where the output for each processing element is calculated until the output of the last processing element in the final layer is found. If this output is 1 then the fault or the alarm has occurred otherwise this type of fault will not arise.

To demonstrate the performance of this approach, an example that deals with the failure of a power control system to respond to boron increase in which the network is shown in figure 4 is considered. The fault tree for this accident is illustrated in figure 5. The symptoms for this type of accident include, an increase in boron concentration, temperature difference across the reactor core and reactor power, and a decrease in the average coolant temperature and pressuriser pressure. Assuming all these symptoms have occurred, the input set to the network is (1 1 1 1 1), where the one's are the input values for the processing elements in the input layer. The network consists of two processing elements in the hidden layer and the output of the first one is calculated according to the threshold logic algorithm as

$$1 \times 1 + 0.5 = 1.5$$

Since $1.5 > 1.0$ then the output of this processing element is 1 indicating the first alarm of failure in the power control system to respond to boron increase. While the output for the second element is

$$1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 1 - 2.5 = 1.5$$

Thus the output is also taken as 1. The output of the processing element in the output layer is then calculated using the same procedure and is found to be equal to 1.5 thus assuring the occurrence of this type of fault.

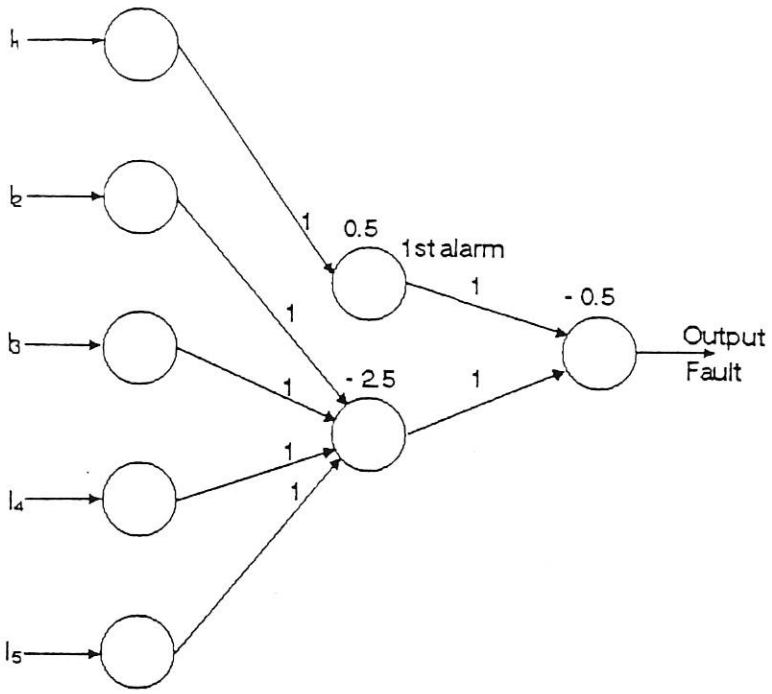

Figure 4: The network for failure in the power control system to respond to boron increase

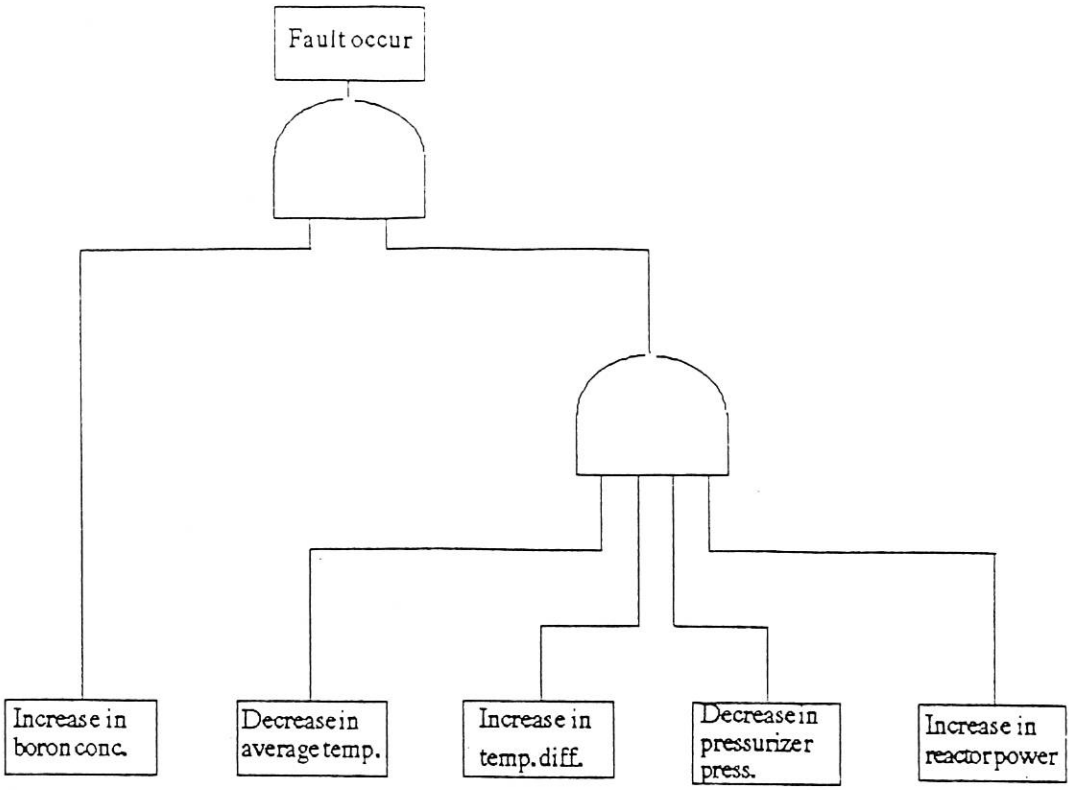

Figure 5: Fault tree for failure in the power control system to respond to boron increase

11 Network Result Using the Threshold Logic Approach

The computer program, again written in C language, simulates the LOFT model, monitors the state variables for any deviation from normal operation, asks the user about the occurrence of other symptoms of different accidents, arranges and inputs the data to the networks and interprets the network output concerning the occurrence of any fault or alarm in the reactor. The program is designed to execute all these procedures every second.

The same hypothetical accidenist used to demonstrate the ability of the back propagation algorithm is used here to examine the ability of this system to assist the nuclear reactor operator in diagnosing accidents. The accident starts by a drop in the value of the coolant flow at a time greater than 2 sec. The neural network model for this type of accident is illustrated in figure 6.

Table 6 illustrates the output of the program where at time 0 sec the system indicates normal operation of the reactor. At time 3 sec it alarms the user for the

partial loss of flow and at time 5 sec the system assures the occurrence of the first, second, and third alarms of partial loss of flow as well as the occurrence of the accident.

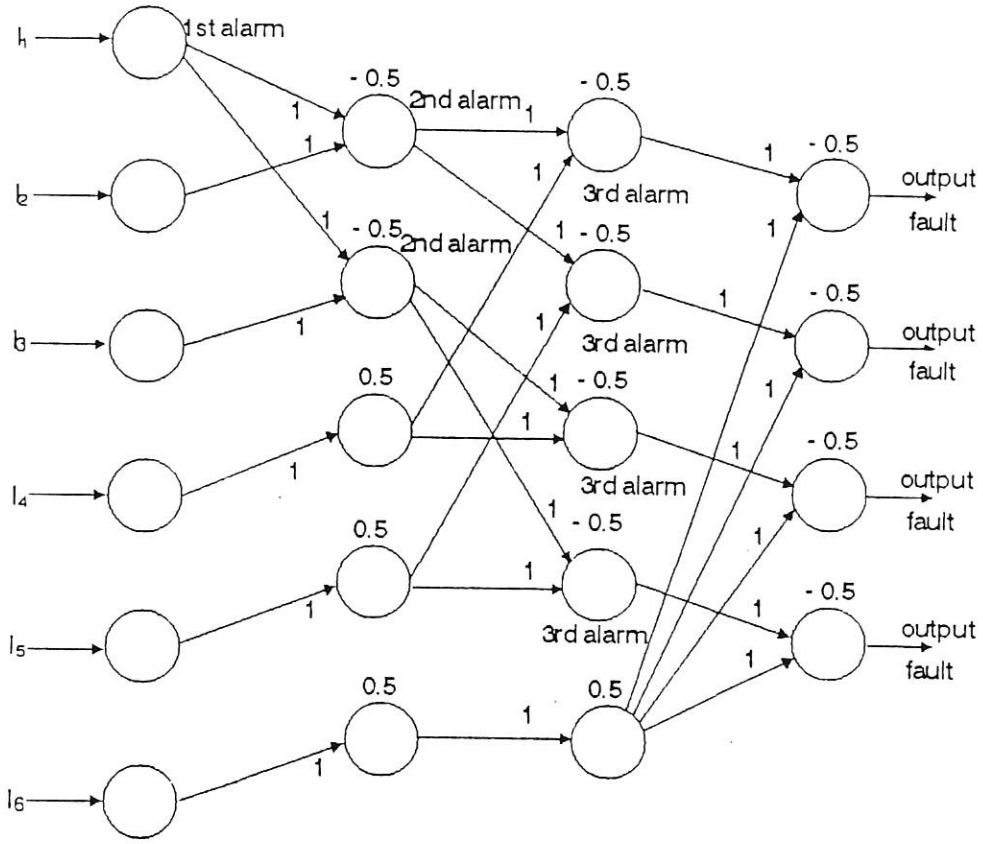

Figure 6: Network for partial loss of flow due to blockage in fuel assembly

Time	Faults and alarms
0.0	Nothing wrong with the reactor
1.0	Nothing wrong with the reactor
2.0	Nothing wrong with the reactor
3.0	First alarm for blockage in fuel assembly
4.0	First alarm for blockage in fuel assembly
5.0	First alarm for blockage in fuel assembly Second alarm for blockage in fuel assembly Third alarm for blockage in fuel assembly Failure due to blockage in fuel assembly

Table 6: The different faults and alarms with respect to time

12 Conclusion

The ability of the neural network to diagnose malfunctioning of the reactor is attractive and encouraging, and the back propagation algorithm worked very well in training the networks to identify the accidents. In all cases, the network output is very near to the desired output indicating that convergence has been achieved using the back propagation algorithm. This approach has been applied to fourteen different accidents and can be easily extended to include different models of faults and accidents that might occur in the reactor plant. The results of the threshold logic approach are also very good, and the approach is simple and easy to apply. The system can be used to represent any type of logic tree and to identify and diagnose any alarm or accident that might occur.

References

[1] Kohonen, T. (1988). *An Introduction to Neural Computing.* Neural Networks, Vol. 1, pp. 3-16.

[2] Roth, M. (1988). *Neural-Networks Technology and It's Applications.* Johns Hopkins APL Technical Digest, Vol. 9, No. 3, pp. 242-253.

[3] Suddarth, S., Sutton, S. and Holden, A. (1988). *A Symbolic Neural Method for Solving Control Problems.* IEEE International Conference on Neural Networks, San Diego, California, Vol. 1, pp. I516-I523.

[4] Bhat, N. and McAvoy, T. (1989). *Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems.* Proceedings of the American Control Conference, Pittsburgh, PA, Vol. 2, pp. 1342-1347.

[5] Hoskins, J. and Himmelblau, P. (1988). *Artificial Neural Network Models of Knowledge Representation in Chemical Engineering.* Comput. Chem. Engng., Vol. 12, No. 9/10, pp. 881-890.

[6] Saito, K. and Nakano, R. (1988). *Medical Diagnostic Expert System Based on PDP Model.* IEEE International Conference on Neural Networks, San Diego, California, Vol. 1, pp. I255-I262.

[7] Gallant, S. (1988). *Connectionist Expert Systems.* Communication of the ACM, Vol. 31, No. 2, pp. 152-169.

[8] Uhrig, R. and Guo, Z. (1989). *Use of Neural Networks to Identify Operating Conditions in Nuclear Power Plants.* Proc. SPIE Int. Soc. Opt. Eng., Vol. 1095, No. 2, pp. 851-856.

[9] Karna, K. and Breen, D. (1989). *An Artificial Neural Networks Tutorial: Part1-Basics.* Neural Networks, Vol. 1, No. 1, pp. 4-22.

[10] Simpson, P. (1990). *Artificial Neural Systems Foundations, Paradigms, Applications and Implementations.* Pergamon Press.

[11] McClelland, J. and Rumelhart, D. (1989). *Explorations in Parallel Distributed Processing. A Handbook of Models, Programs, and Exercises.* MIT Press.

[12] Rumelhart, D. and McClelland, J. (Eds.) (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations.* MIT Press.

[13] Werbos, P. and Park, C. (1988). *Back Propagation: Past and Future.* IEEE International Conference on Neural Networks, San Diego, California, Vol. 1, pp. I343-I353.

[14] Tylee, J. (1980). *Low Order Model of the Loss Of Fluid Test (LOFT) Reactor Plant for Use in Kalman Filter Based Optimal Simulation.* Proceedings of the 4th Power Plant Dynamics, Control and Testing Symposium, Gatlinburg, pp. 1-31.

[15] Tylee, J. (1980). *Low Order Model of the Loss Of Fluid Test (LOFT) Reactor Plant for Use in Kalman Filter Based Optimal Estimators.* EGG-2006, EG & G Idaho Falls, Idaho.

[16] Lippman, R. (1987). *An Introduction to Computing with Neural Nets.* IEEE ASSP Magazine, Vol. 4, No. 2,pp. 4-22.