MODEL BASED FAULT DETECTION IN PROCESS PLANT

Leong Wah Heng , Arshad Ahmad
Faculty of Chemical & Natural Resources Engineering
Universiti Teknologi Malaysia
80990 Johor Bahru, Malaysia
(arshad@fkkksa.utm.my)

Abstract

In this paper, the application of neural network in detecting sensor failures is presented. The study was conducted on the Tennessee Eastman test problem. Faulty conditions were generated by imposing sensor failures in the reactor operation. Both single and multiple fault conditions were investigated. The results revealed that a three layer feedforward network was capable of providing the intended function.

Introduction

In the recent years, process industries are becoming increasingly competitive. Stringent production specification and tight regulations imposed by local governments have led to the needs for high quality production environment as well as more efficient and effective plant control system. As a result of this scenario, majority of today's plant are equipped with computer-controlled system. However, the success of any control system is strongly dependent on the availability of precise process information from the data logging facilities. If incorrect information due to abnormalities in the process is fed to the control system, inappropriate control action will be taken. This may then lead to plant upset.

Process faults can appear in various different forms. Among others, Kanuri and Ventakatasubramaniam (1993) outline the following:

1. gross parameter changes in the actual process such as a change in the feed concentration to the reactor and fouling in a heat exchanger.
2. structural changes such as failure of a control valve and a broken pipe.
3. corrupted measurement such as sensor failure, biased signals and instruments saturation.

These failures may lead to greater flaws if not detected. To alleviate such drawbacks, preventive measure should be taken. One way is to employ fault detection and diagnosis facilities within the data gathering system.

Recently, considerable attentions have been devoted to the development of various fault detection techniques and their application to chemical process industries. Among other techniques, the use of Artificial Neural Networks (ANN) has been found promising. The learning ability of ANN has made it possible to train an ANN model to comprise various fault conditions. Given sufficient training with adequate data, neural network will be able to classify "unseen" fault patterns.

In this paper, a case study involving the application of neural network in detecting sensor malfunction is presented. Various network topologies are tested and compared. Also presented here is the general neural network architecture and backpropagation algorithm.

Neural Network And Backpropagation Algorithm

Artificial Neural Network is a computing tool that is inspired by the model of human brains. An ANN is constructed of interconnected basic elements called neurons or nodes. A schematic diagram of this node is shown in Figure 1. Similar to their biological counterpart, these nodes are capable of processing incoming information and transferring them to other nodes.

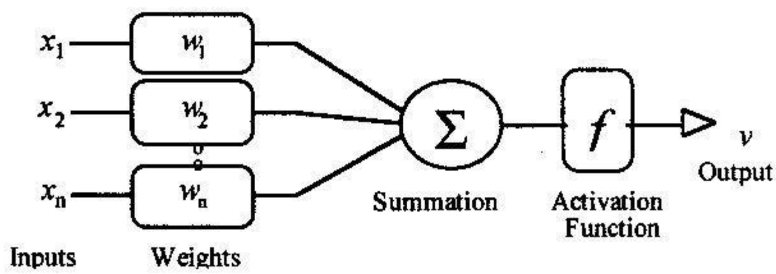


Figure 1 : Schematic Diagram of an Artificial Neuron

The input signals come from either the environment or outputs of other nodes through connections as specified by the network architecture. Within each node, input signals are summed and transformed using an activation function before being sent to other nodes. Transformation of data via activation functions is needed to impart pattern mapping ability to the networks. An example of such function is the sigmoid function given in the following equation:

$$f(z) = (1 + e^{-z})^{-1} \tag{1}$$

Here, z is a weighted sum of all inputs. Associated with each connection is an adjustable value called the network weight. During learning, these weights are adjusted to fulfill the training objective. Effectively, network weights serve as a measure for connection strength that control the influence of each incoming signal on the recipient node.

In process engineering application, the most common network architecture is the multilayer feedforward network. As displayed in Figure 2 below, this network is constructed of nodes arranged in several layers. There is an input layer to receive the incoming data to the network, and an output layer to deliver the processed data from the network. In between these two layers, there could be several layers known as the *hidden layers*. The nodes in the input layer do not perform any processing and is used to distribute all inputs to the next layer. All other nodes carry out information processing as mentioned earlier. In addition to the regular nodes, bias nodes with fixed output of unity are often connected to all the nodes in the hidden and output layers.

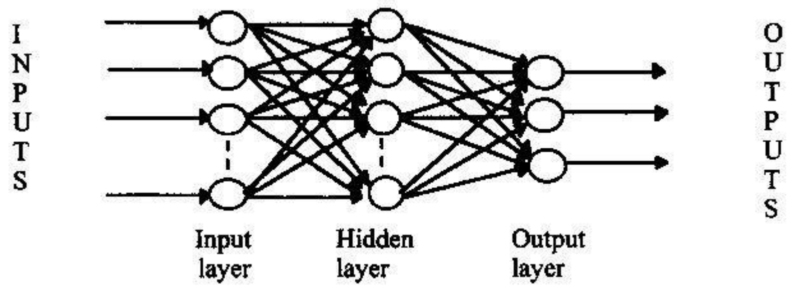


Figure 2 : Multilayer Feedforward Network

For the purpose of fault detection and diagnosis, model development task is divided into 3 main stages: training or learning phase, recall phase and generalization phase. Training refers to process of feeding information to the ANN model. Recall is the process of applying the model to similar fault patterns. Finally in the generalization phase, the model is applied to data sets with different fault patterns.

In this study, backpropagation learning algorithm is employed to adjust network weights to minimize the error between the actual output and the target output. The error function is as follows:

$$E = \sum_{m=1}^{M}\sum_{i=1}^{N}(t_i^{(m)} - y_i^{(m)})^2 \quad (2)$$

Here, M and N denote the number of training patterns presented to the input layer and the number of nodes in the output layer respectively, $t_i^{(m)}$ represents the desired value of the ith output element given the mth pattern, while $y_i^{(m)}$ is the actual output of the same element. Weights are updated in the training according to the following equation:

$$W_{ji}^{(m)} = W_{ji}^{(m-1)} + \Delta W_{ji}^{(m)} \quad (3)$$

Here, $W_{ji}^{(m)}$ denotes the weight of the connection between the jth element of the upper layer and the ith element of the lower layer in the mth learning iteration. The weight change $\Delta W_{ji}^{(m)}$ is calculated as follows:

$$\Delta W_{ji}^{(m)} = \eta \, . \, \delta_j^{(m)} \, . \, O_i^{(m)} + \alpha \, . \, \Delta W_{ji}^{(m-1)} \qquad (4)$$

where η and α denote the learning rate and coefficient of the momentum term, $O_i^{(m)}$ is the output value of the ith element in the previous layer and $\delta_j^{(m)}$ is the gradient descent term.

Case Study : Detection of Sensor Failures in the Tennessee Eastman Plant

Process Description

The application of neural network in fault detection and diagnosis is implemented to the Tennessee Eastman (TE) plant. In this plant, two products (G, H) are produced from four reactants (A, C, D, E). Also present in the process is an inert B and a by-product F. Figure 3 shows the schematic diagram of the process. TE plant can be divided into five major section: (1) feed mixing zone, (2) reactor, (3) vapor-liquid separator, (4) product stripper and (5) recycle compressor .

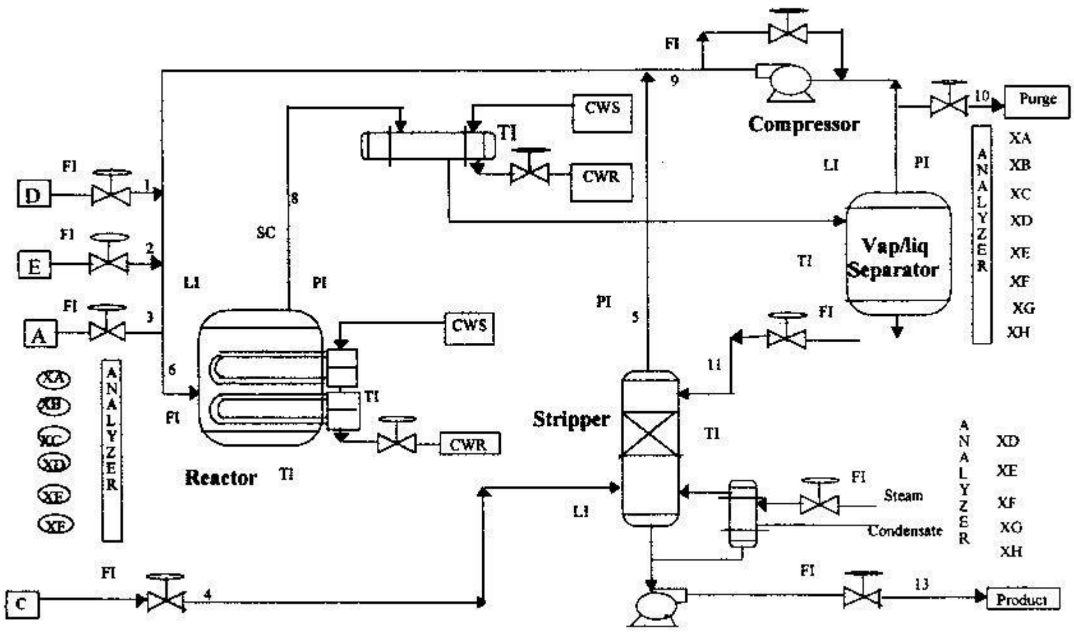


Figure 3 : Schematic Diagram of Tennessee Eastman Plant

Gaseous reactants are fed to the reactor in which they react to form liquid products. The product leaves the reactor as vapors along with unreacted feeds through a stream that passes through a cooler and then to a vapor-liquid separator. Noncondensible components are recycled through a centrifugal compressor to the reactor feed. Condensed components are fed to a product-stripping column to remove remaining reactants by stripping with feed stream number 4. Products G and H leave the stripper base to be separated in a downstream refining section. The inert and by-product are primarily purged from the system as a vapor from the vapor-liquid separator.

Simulation Studies

This investigation focused on the malfunctions of the process caused by the failure of the level, temperature and pressure sensors in the reactor. Faulty conditions are simulated using the Tennessee Eastman Plant model coded in FORTRAN language. Sensor failures are created causing the normal process operation to shift to a faulty operation mode. Effects of these faults are expressed by the composition of components A to H in the purge stream and the composition of components D,E,F,G,H in the product stream leaving the stripper. During implementation, sensor failures are categorized into high and low readings. Thus, there are two possible faults for each sensor. For the level sensor, deviation of ±5.0% from the normal condition is assumed to cause malfunction to the process. Similarly, for temperature and pressure sensors the figures are ±0.5% and ±1.0% respectively.

The neural network models employed in this case study consist of 13 input nodes. Each node corresponds to a variable used to measure the effect of malfunction. There are six output nodes representing six fault conditions caused by the sensor failures. Throughout this implementation, one hidden layer was used. Various combinations of activation functions for hidden and output layers were tested. The best results were obtained when sigmoid function were applied in both layers. Therefore, throughout this investigation sigmoid function was adopted.

For the purpose of training, 12 input-output patterns were generated. The conditions were displayed in Table 1. The desired output for each patterns were discretized between 0 and 1. For example, output nodes representing 5% deviations in level sensor reading were assigned the value of $d_i = 0.333$ and a 15% deviation was given the value of $d_i = 1$. Output nodes not representing any failures were fixed at $d_i = 0$. All input data were normalized prior to the training phase. To enhance the backpropagation training, momentum term was used. A learning rate η and a momentum coefficient α of 0.27 and 0.5 were chosen for this study.

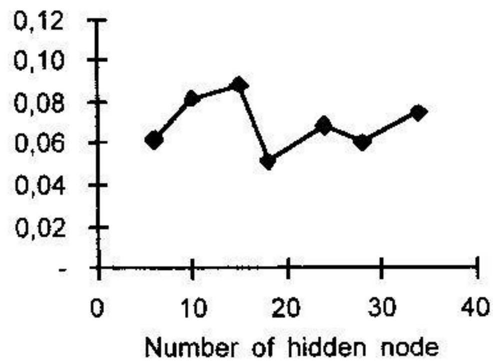
Table 1: Training Pattern

Malfunction	Deviation from normal condition	Targeted output for node giving fault reading
high reading at level sensor, F1	+5.0 %	0.3333
	+15.0 %	1.0
low reading at level sensor, F2	-5.0%	0.4167
	-12.0%	1.0
high reading at temperature sensor, F3	+0.5%	0.3333
	+1.5%	1.0
low reading at temperature sensor, F4	-0.5%	0.3333
	-1.5%	1.0
high reading at pressure sensor, F5	+1.0%	0.3333
	+3.0%	1.0
low reading at pressure sensor, F6	-1.0%	0.3333
	-3.0%	1.0

Result and Discussion

The effect of increasing the number of hidden nodes on the sum squared error (SSE) in fault detection was explored. Figure 4 shows the effect hidden nodes to SSE after the 5000 training iteration. As observed, the network with 18 hidden nodes gives the minimum SSE. However, this is for the training stage. The topology will be considered as "the best" if it is able to generalize well on untrained patterns. Here, SSE only indicates that the network is converged to certain minimum error with the learning algorithm used.

Figure 4 Sum Square Error vs Hidden Nodes



To test the generalization ability of the network, the following faults pattern were generated:

Single faults: 1) level sensor with 10% deviation, G1
2) level sensor with -10% deviation, G2
3) temperature sensor with 1.0% deviation, G3
4) temperature sensor with -1.0% deviation, G4
5) pressure sensor with 1.5% deviation, G5
6) pressure sensor with -1.5% deviation, G6

Double faults : G1+G3 , G1+G4 , G1+G5 , G1+G6 , G2+G4, G2+G6 , G3+G5 , G3+G6 , G4+G5 , G4+G6

Triple faults : G1+G3+G5 , G1+G3+G6 , G1+G4+G5 , G1+G4+G6 , G2+G3+G6, G2+G4+G5 , G2+G4+G6

In order to interpret the output of the network during the generalization of single and multiple faults, the following criteria had been set:

"output is considered as fault only if it is greater than 90% of the targeted outputs for ±5.0% deviation in level sensor, or ± 0.5% deviation in temperature sensor, or ± 1.0% deviation in pressure sensor. Otherwise, the output nodes is considered normal (no fault)."

Network with hidden nodes of 10, 18, 24 and 28 trained for 6000 training iteration were used in the generalization. Based on the criterion stated above, all the networks successfully generalized single-fault patterns. For generalization of double and triple faults, the results were given in the Table 2 and 3 respectively.

Table 2 Generalization of double faults

	G1 + G3	G1 + G4	G1 + G5	G1 + G6	G2 + G4	G2 + G6	G3 + G5	G3 + G6	G4 + G5	G4 + G6
10 nodes	G1(f) G3(i)	a	a	a	G2(f) G4(I)	G2(i) G6(f)	G3(i) G5(f)	a	a	G1(w) G4(I) G6(f)
18 nodes	G1(f) G3(i)	a	a	a	a	G2(i) G6(f)	G3(i) G5(f) G2(w)	a	a	G4(i) G6(f)
24 nodes	a	a	a	a	a	G2(I) G6(f)	a	a	a	a
28 nodes	a	a	a	a	a	G2(I) G6(f)	a	a	a	G4(i) G6(f)

Notes : (i) correctly identify/detect a single fault
(f) fail to identify/detect
(w) wrongly identify/detect
a correctly identify all the faults

Table 3 Generalization of triple faults

	G1+G3 +G5	G1+G3 +G6	G1+G4 +G5	G1+G4 +G6	G2+G3 +G6	G2+G4 +G5	G2+G4 +G6
10 nodes	G1(f) G3(i) G5(f)	G1(f) G3(i) G6(i)	G1(f) G4(i) G5(f)	G1(i) G4(i) G6(f)	G2(f) G3(i) G6(f)	G2(f) G4(f) G5(i)	G2(f) G4(i) G6(f)
18 nodes	G1(f) G3(i) G5(f)	a	G1(f) G4(i) G5(f)	G1(i) G4(i) G6(f)	G2(i) G3(i) G6(f)	a	G2(i) G4(i) G6(f)
24 nodes	G3(i) G1(f) G5(f)	a	a	a	G2(i) G3(i) G6(f)	a	G2(i) G4(i) G6(f)
28 nodes	G3(i) G1(f) G5(f)	a	a	a	G2(i) G3(i) G6(f)	a	G2(i) G4(i) G6(f)

Results as displayed in Table 2 and 3 show that networks with 24 hidden nodes was able to generalize 4 out of 7 untrained patterns of triple faults and almost all the double faults (9 out of 10). It also indicated that increasing number of hidden nodes increases the ability to classify the input patterns correctly.

Conclusion

In this case study, three layer feedforward neural network trained using backpropagation algorithm has been proven to be able to detect and diagnose sensors failure in the process plant. For the particular fault pattern tested, hidden layer with 24 hidden nodes shows excellent performance in detecting "unseen" data of single and multiple faults.

This study was constrained to steady-state analysis. Extension to the dynamic characteristics of faults will be presented in a later communication.

References

Downs J. J., Vogel E. F. (1993). A Plant-Wide Industrial Process Control Problem. *Computer Chem. Eng.*, 17, 245-255

Kavuri S.N. , Venkatasubramanian V. (1993). *Neural Network Decomposition Strategies For Large Scale Fault Diagnosis.* Computer Integ. Process Operation Center, Purdue University.

Venkatasubramaniam V. (1990). Process Fault Detection And Diagnosis Using Neural Networks - I. Steady-State Processes. *Computers chem. Engineering.* 14, 699-712