

CHEMICAL PLANT FAULT DIAGNOSIS BY NEURAL NETWORKS

An analysis of learning and generalization characteristics of neural networks for diagnosing process failures was presented. Various feedforward neural network topologies were tested and compared. The single fault assumption was relaxed to include multiple causal origins of the symptoms. A chemical plant composed of a reactor and a distillation column was used as a case study. The performance during recall improves at first with an increase in the number of hidden units and with the amount of training, and then attains convergence. The algorithm of the Generalized Delta Rule (GDR) was used to train the networks by minimizing the sum of squares of residual according to the given convergence criterion. The obtained results show the applicability of the neural networks structure with hidden layers for process fault diagnosing. These results illustrate the feasibility of using neural networks for fault recognition and location.

Neural computing is one of the fastest growing areas of artificial intelligence. A neural net consists of a number of connected processing units of neurodes, each of which performs a non-linear transformation on its inputs. Often the neurodes are organized into distinct layers, input, hidden or associative, and an output layer [1-3].

Since a non-linear governing relationship can be handled by neural nets, the nets may offer a cost effective approach to modeling chemical process systems [4-7]. Modern chemical plants are highly complex. This, coupled with quick diagnosis, rules out exclusive reliance on human operation for diagnosis. Only a few knowledge based expert system - KBES approaches have been proposed in the literature for automated fault diagnosis.

However, the rapid deployment of these systems has been difficult to achieve due to certain inherent limitations associated with current KBES. These limitations include the tedious nature of knowledge acquisition, the inability of the system to learn or dynamically improve its performance and the unpredictability of the system outside its domain of expertise. A potential solution to these problems is the use of neural networks as demonstrated in recent papers [8-10].

The application of artificial neural networks to process engineering problems, notably malfunction diagnosis, has been discussed in papers [10-13].

The purpose of this paper is to illustrate an artificial neural network application for fault detection and diagnosis. The effect on incomplete and uncertain process symptom data such as sensor faults was. A reactor distillation column for the reaction esterification

of acetic acid with ethanol was investigated as a model chemical plant.

NEURAL NETWORK TRAINING

Artificial neural networks are composed of many simple computational elements; nodes, locally interacting across very low bandwidth channels, connections. The architecture of these models is specified by the node characteristics, network topology and learning algorithm.

Neural architectures are based on a common computing or threshold logic unit, TLU, shown in Fig. 1, with models associated with concepts and are weights representing similarity. The implementing element, TLU consists of a single concept node, all of the inputs signals, and an output signal. The TLU computes the $W-X-T$ function for a series of X imputes and the unit threshold characteristic T .

Because these units implement a linear surface within some multiple measurement spaces, they can be corrected to form multiple surfaces that portion space into different decision regions.

Training, or modification of the decision surface so that it maintains consistency with some know phenomenon, is realized by adjusting the weights and threshold to ensure that the appropriate unit activates for representative samples of a given population.

Although a network system readily supports automated learning, the link between learning and

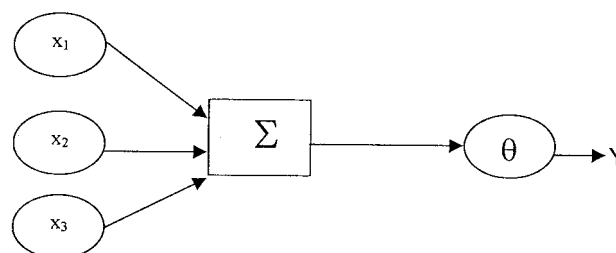


Figure 1. Threshold logic unit

parsimony is the development of a structure for characterizing an active goal hierarchy that represents a logical and coherent cognitive theory within some world model. This neural architecture (Figure 1) consists of three unit types: input units, internal units and output units. Input units are either facts of state conditions. Within the context of the simulation, inputs are the data base and are derived from the outcome of previous events. Internal units are located between the set of input units and the set of output units. Within the context of a production system, these units constitute the knowledge base. The output units are events to be scheduled.

The neural network structure is denoted as $(I \times H \times O)$, where H , O , and I represent the number of units, respectively. Although there has been some research on the design of optimal neural network of NN structure, it is still largely an art to determine the number of hidden layers and number of units in each hidden layer.

The backpropagation training algorithm has been used successfully in training the neural network for wide applications. Backpropagation of the algorithm adjusts the weights in a feedforward neural network consisting of several layers, and an output layer. The goal is to teach the network to associate specific output states, called target states, to each of several input states. Having learned the fundamental relationships between the inputs and outputs, the neural network can produce the correct output for a new, previously unseen input. The back propagation learning algorithm does not need too much computation time to obtain the correct weights, if the training data is small in size. In other words, if the relationship between the inputs and outputs is simple, a set of weights is easy to obtain. Most often for complex problems the relationship between the inputs and outputs cannot be completely represented with a small number of training data sets. Therefore, the computation time required by backpropagation is very large.

Training a Feedforward Neural Network

A popular configuration of networks for backpropagation is a totally feedforward net [10, 15]. In feedforward nets inputs feed up through layers to an output layer. Each neuron forms a weighed sum of the inputs from a previous layer to which it is connected, adds a threshold value and produces a non-linear function of this sum as its output value (Figure 1). The output value serves as the input to the next layer to which the neuron is connected and the process is repeated until values are obtained for the neurons in the output layer. Thus, each neuron performs

$$Y_j^p = f(\sum w_{ij}x_i^p - \theta_j) \quad (1)$$

where w_{ij} is the weight from neuron i to neuron j , w_{ij} can be a positive or negative real number, and θ_j is the threshold of the j -th neuron, p means the p -th pattern.

The $f(x)$ is a non-linear activation function that is often chosen to be a sigmoidal form:

$$f(x) = (0.5) (1 + \tanh(x)) \quad (2)$$

is used in this analysis where \tanh is the hyperbolic tangent. If d_i^p are the desired outputs and y_i^p are the outputs obtained from the output layer for the p -th pattern, neural nets are trained by minimizing the error function

$$E = \sum \sum (d_i^p - y_i^p)^2 \quad (3)$$

where i indexes the number of neurons in the output layer, and p means the p -th input pattern of the presented training set of the input layer. Backpropagation by the generalized delta rule, a kind of gradient descent method is one popular method [2-5, 10, 12, 13]. The gradient descents method is described by the following eqns (4) - (9).

The commonly used steepest descents procedure in minimizing E is to change w_{ij} and θ_i by Δw_{ij} and $\Delta \theta_i$ where

$$\Delta w_{ij} = \frac{\partial E}{\partial w_{ij}} \eta \quad (4)$$

$$\Delta \theta_j = \frac{\partial E}{\partial \theta_j} \eta \quad (5)$$

where η is the learning rate.

After simplification Δw_{ij} and $\Delta \theta_j$ can be expressed as

$$\Delta w_{ij} = -\eta \sum \delta_j^p y_j^p \quad (6)$$

$$\Delta \theta_j = -\eta \sum \delta_j^p \quad (7)$$

where

$$\delta_j^p = (d_j^p - y_j^p) y_j^p (1 - y_j^p) \quad (8)$$

if the j -th neuron is the output layer, and

$$\delta_j^p = y_j^p (1 - y_j^p) \sum \delta_k^p w_{jk} \quad (9)$$

if the j -th neuron is the hidden layer and k is the number of neurons in the layer above neuron j (Figure 2).

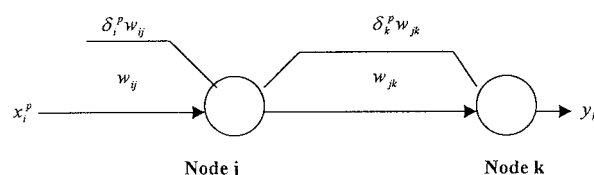


Figure 2. Backpropagation at the error signal

THE BACKPROPAGATION ALGORITHM WITH THE GENERALIZED DELTA RULE

Network training aims at achieving the least of errors, the errors measured as the difference between the calculated output and the desired output. In this

investigation the back propagation algorithm by the generalized delta rule – GDR method (Figure 3) was used to calculate the weights in the neural networks.

Step 1. Assign all neuron offsets (thresholds) to small random values: θ_j
Step 2. Assign all weights to small random values: w_{ij}
Step 3. Repeat
 for $p=1$ to TP (TP is total number of training patterns)
 for $j=1$ to n_2 (n_2 is the number of neurons in the hidden layer)
 calculate neuron outputs in the hidden layer: y_j endfor
 for $k=1$ to n_3 (n_3 is the number of neurons in the output layer)
 calculate neuron outputs in the output layer: y_k calculate δ_k^p
 $\theta_k^{(p)} = \theta_k^{(p-1)} + \Delta\theta_k^{(p)}$
 end for
 for $j=1$ to n_2 and $k=1$ to n_3
 $w_{jk}^{(p)} = w_{jk}^{(p-1)} + \Delta w_{jk}^{(p)}$
 end for
 for $j=1$ to n_2 calculate δ_j^p
 $\theta_j^{(p)} = \theta_j^{(p-1)} + \Delta\theta_j^{(p)}$
 end for
 for $i=1$ to n_1 and $j=1$ to n_2 (n_1 is the number of neurons in the input layer)
 $w_{ij}^{(p)} = w_{ij}^{(p-1)} + \Delta w_{ij}^{(p)}$
 end for
 end for
 until $w < \xi$ (ξ is the convergence criterion)

Figure 3. The back propagation GDR algorithm

FAULT DETECTION AND DIAGNOSIS EXAMPLE

Modern chemical plants are extremely complex which makes them susceptible to equipment malfunction and operator error. The complexity hampers the operator's ability to diagnose and eliminate potential process upsets or equipment failures before they can occur. Hence, a continuing problem is how to use the process state vector at each time increment to make or aid decisions about possible action or control. The study

of fault detection and diagnostics is concerned with designing systems that can assist the human operator in detecting and diagnosing equipment faults in order to prevent accidents.

In order to analyze the learning and generalization characteristics of neural networks for process fault diagnosis in a steady state process, the following example was investigated (Figure 4). The test problem consisted of diagnosing faults for a model chemical plant composed of a reactor and distillation column for the reaction esterification of acetic acid with ethanol. The reactor effluent stream containing a mixture of acetic acid, ethanol, ester and water is fed into the distillation column where it is separated into a distillate stream containing 78% of the ester and a bottoms stream containing 22% of the ester [16]. The column is provided with controllers that control the overhead and bottoms product compositions by manipulating the reflux rate R and the vapor boilup rate V , respectively. Table 1 presents the selected faults that can occur in the augmented plant of Figure 4.

Table 1. Selected malfunctions for a model chemical plant

Malfunction (fault)	Symbol
High flowrate of ethanol at the reactor inlet	F1
Low flowrate of ethanol at the reactor inlet	F2
High flowrate of acetic acid	F3
Low flowrate of acetic acid	F4
High concentrate of ethanol at the reactor inlet	F5
Low concentrate of ethanol at the reactor inlet	F6
High concentrate of acetic acid	F7
Low concentrate of acetic acid	F8
High temperature of the reactor inlet	F9
Low temperature of the reactor inlet	F10
Bottom composition controller failure	F11
Distillate composition controller failure	F12

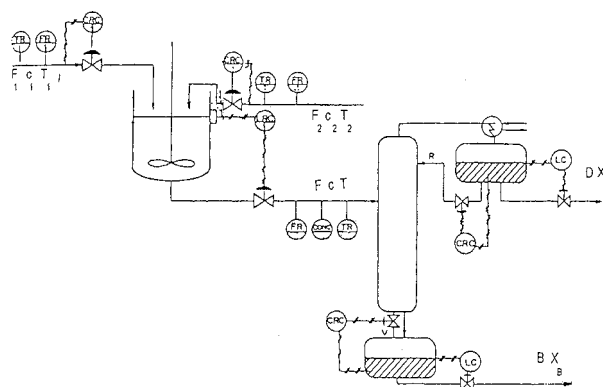


Figure 4. The chemical plant of a reactor-distillation column

The measurement patterns required for training and testing the neural network were obtained from simulation programs that model the behavior of the plants. Various malfunctions were simulated and the new steady state measurement patterns obtained were used to train the network.

The variables causing malfunction are the inlet flowrate, inlet temperature and the inlet concentration of reactants. Here also, deviations beyond $\pm 5\%$ of the normal value of these variables are assumed to result in F2, a malfunction of the plant. Deviations within the $\pm 5\%$ threshold are assumed not to perturb normal plant operation. In addition, the bottoms and overhead product composition controllers can fail.

The neural network employed in this case has 12 output nodes corresponding to the 12 possible malfunctions. The state variables characterizing this plant are the concentration of ethanol at the reactor inlet c_1 , the concentration of acetic acid at the reactor inlet c_2 , the reactor effluent temperature T , the reactor holdup V , the reactor outlet flowrate of feed rate to distillation column F , the reflux rate R , the vapor boilup rate V_G , the bottoms flow rate B , The distillate flow rate D , the bottoms product composition X_B , and the distillate product composition X_D is expressed in terms of the mole fraction of ester.

Initially a network with 11 input nodes corresponding to the state variables c_1 , c_2 , V , X_D , X_B , F , R , V_G , B and D was chosen for training. The NN with 11 input units, 12 output units and 6 hidden units is shown in Figure 5. Table 2 indicates the 13 unnormalized measurement patterns were used to train the network. This network failed to recall the first 10 malfunctions correctly after 15000 iterations.

Table 2. Measurements patterns used for training networks

No	Fault	c_1	c_2	T	V	F	R	V_G	X_D	X_B	D	B
1	F1(+15%)	0.575	0.500	350.5	3.55	3.25	4.10	5.99	0.780	0.2200	1.75	1.50
2	F2(-15%)	0.515	0.525	349.0	3.50	3.00	3.95	5.50	0.780	0.2200	1.55	1.45
3	F3(+15%)	0.577	0.575	351.3	3.53	3.15	4.00	5.65	0.780	0.2200	1.65	1.50
4	F4(-15%)	0.525	0.457	347.0	3.52	3.10	3.95	5.55	0.780	0.2200	1.60	1.50
5	F5(+15%)	0.585	0.515	352.5	3.54	3.13	4.50	5.60	0.780	0.2200	1.61	1.52
6	F6(-15%)	0.485	0.535	346.5	3.54	3.17	3.93	5.63	0.780	0.2200	1.70	1.47
7	F7(+15%)	0.530	0.580	350.0	3.56	3.05	3.74	5.52	0.780	0.2200	1.78	1.27
8	F8(-15%)	0.570	0.495	349.8	3.55	3.10	3.94	5.59	0.780	0.2200	1.65	1.45
9	F9(+15%)	0.574	0.537	350.1	3.54	3.23	4.21	5.95	0.790	0.2150	1.74	1.49
10	F10(-15%)	0.482	0.499	349.0	3.53	3.03	3.97	5.60	0.770	0.2000	1.63	1.40
11	F4 and F8	0.525	0.485	349.5	3.54	3.16	4.09	5.75	0.788	0.2100	1.66	1.50
12	F5 and F9	0.560	0.510	350.0	3.49	3.08	4.08	5.80	0.785	0.2199	1.72	1.36
13	F1 and F11	0.550	0.505	348.1	3.52	3.19	4.10	5.89	0.787	0.2050	1.79	1.40

V, F, R, V_G, B, D , (mol s^{-1}); $T(K)$; c_1, c_2 (mol/cm³); X_D, X_B (mol/mol).

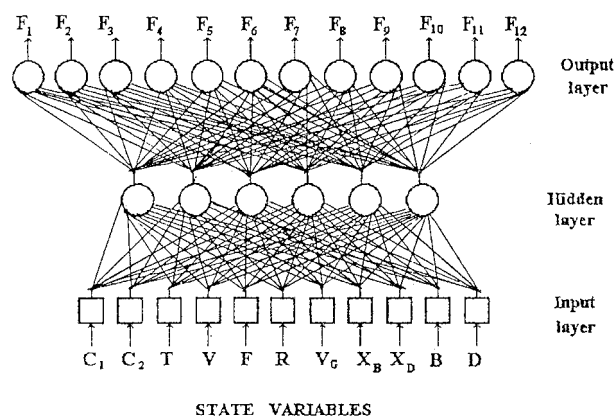


Figure 5. The neural network NN (11x6x12)

Finally, three multiple faults were simulated, as well as the ability of the network to correctly identify these malfunctions.

Networks with various numbers of hidden units were explored. The root mean square error during training was between 0.005 and 0.358. The learning rate $\eta = 0.9$ was used.

CONCLUSIONS

In this paper the usefulness of neural network application in automated fault diagnosis was demonstrated. It shows how neural networks are successful in diagnosing single faults in the case study of distillation with reaction esterification of acetic acid and ethanol. Performance during recall improves at first with an increase in the number of hidden units and with the amount of training, and then attains convergence.

In order to extend the applicability of the model, the ability of the network trained on single faults was to diagnose multiple faults.

Changes in plant configuration necessitate retraining of the neural network, which is simpler than revising the complex knowledge base of an expert system. Another advantage of neural networks is their robustness and resistance to input noise.

NOTATION

B	– bottoms flow rate, mol s ⁻¹
c ₁	– concentration of ethanol at the reactor inlet, mol/cm ³
c ₂	– concentration of acetic acid at the reactor inlet, mol/cm ³
d	– desired output
E	– minimizing error function
F	– malfunction
R	– reflux rate, mol s ⁻¹
T	– temperature, K
V	reactor holdup, mol/s
V _G	– vapor boilup rate, mol s ⁻¹
X _B	– bottoms product composition, mol/mol
X _D	– distillate product composition, mol/mol
x	– input
y	– calculated output
w	– weight of each branch

Greek Symbols

δ	– parameter
η	– learning rate
θ	– threshold
ξ	– convergence criteria

REFERENCES

- [1] J.D. Morison, *Simulation*, **58** (1992) 152.
- [2] C.J. Wang and C.H. Wu, *Simulation*, **57** (1991) 223.
- [3] G.A. Korn, *Simulation*, **56** (1990) 214.
- [4] J. Savković–Stevanović, *Computers chem Engng.*, **17**, (1993) 411.
- [5] J. Savković–Stevanović, M. Vico–Stevanović, S. Jorgačević and N. Tica, An artificial neural network for variable estimation and state identification in process fermentation, *Proceedings of the Summer Computer Simulation Conference SCSC*, Boston Massachusetts, July 19–21 (1993) 323–327.
- [6] N. Bhat and T.J. McAvoy, *Computers Chem Engng.*, **14** (1990) 573.
- [7] M.L. Padget and T.A. Roppel, *Simulation*, **58** (1992) 295
- [8] J.C. Hoskins and D.M. Himmelblau, *Computers chem Engng.*, **12** (1998) 881.
- [9] V. Venkatasubramanian and R. Chan, *AIChEJ.*, **35** (1989) 1993.
- [10] V. Venkatasubramanian, V. Vaidyanathan and Y. Yamamoto, *Computers Chem Engng.*, **14** (1990) 699.
- [11] M.A. Kramer and J.A. Leonard, *Computers Chem. Engng.*, **14** (1990) 1323.
- [12] J. Savković–Stevanović, Use of neural nets for process fault diagnosis, *Proc. of the CHEMCON'94 – Intern. Symp. on Artificial Intelligence and Expert Systems in the Process Industry*, Dec. 15, IIT, Karghapur, India, (1994) 116–128.
- [13] J. Savković–Stevanović, Use of neural nets for chemical plant fault diagnosing, *Proc. of the IES'95 – The Ninth Sem. and Symp. on Information and Expert Systems in the Process Industries*, Belgrade, Oct. 4–5 (1995) 61–71.
- [14] S. Haybin, *Neural Networks*, McMillan, Publish. Co., New York (1994)
- [15] W.P. Jones and J. Hoskins, *Byte*, Oct. (1987) 155.
- [16] J. Savković–Stevanović, M. Mišić–Vuković, G. Bončić–Caričić, B. Trišović, S. Jezdić, *Sep. Sci. and Tech.*, **27** (1992) 613.

IZVOD

DIJAGNOSTIKA KVARA HEMIJSKOG POSTROJENJA POMOĆU NEURONSKIH MREŽA

(Naučni rad)

Jelenka B. Savković–Stevanović,
Tehnološko–metalurški fakultet, Univerziteta u Beogradu, Beograd

Data je analiza procesa učenja i karakterizacije neuronskih mreža za dijagnostiku hemijskih procesa, testirane su i poredene različite topološke strukture neuronskih mreža. Pretpostavljajući jedan kvar uvode se i analiziraju višestruki uzroci simptoma. Kao primer za ispitivanje posmatrano je hemijsko postrojenje koje se sastoji od reaktora i destilacione kolone. Performanse neuronske mreže se poboljšavaju, pre svega povećanjem broja međučvorova i treninga, da bi se na kraju postigla konvergencija. GDR algoritam opšteg delta pravila je korišten za treniranje mreže prema datom konvergentnom kriterijumu.

Ključne reči: Analiza rizika • dijagnostička neuronska mreža • algoritam opšteg delta pravila • analiza simptoma i uzroka poremećaja
Key words: Risk analysis • Neural networks • Fault diagnosis by neural networks • Fault locator.

