Proceedings of ESDA2006
8th Biennial ASME Conference on Engineering Systems Design and Analysis
July 4-7, 2006, Torino, Italy

ESDA2006-95173

FAULT DIAGNOSIS COMPETITIVE NEURAL NETWORK TRAINING THROUGH CONDITION MONITORING OF INDUSTRIAL MACHINES AND STOCK EXCHANGE PREDICTION

Sohrab Khanmohammadi
Faculty of ECE, University of Tabriz, Iran
khan@tabrizu.ac.ir

Sayyed Mahdi Hosseini
Control and Intelligent Processing Center of Excellence
Faculty of ECE, University of Tehran, P.O. Box 14395/515, Tehran, Iran
sm.hosseini@ece.ut.ac.ir

ABSTRACT

In this paper a new way for neural network training is introduced where the output of middle (hidden) layer of neural network is used to update weights in a competition procedure. Output layer's weights are modified with multi layer perceptron (MLP) policy. This learning method is applied to two systems as case studies. First one is the monitoring of industrial machine where the results are compared with other training methods such as MLP or Radial Basis Function (RBF). Oil analysis data is used for condition monitoring. The data is gathered by using ten stages technique. The second one is the Stock prediction where the data are highly nonlinear and normally unpredictable especially when the markets are affected by political facts. The simulation results are analyzed and compared with other methods.

INTRODUCTION

The fault detection and diagnosis is one of the most expensive tasks of the maintenance programs of plants [2].

In application of neural network to fault diagnosis, the neurons symbolize the different subsystems of a plant or organization, classified in different levels, represented by network's layers [1]. This NNT can be applied for N input data consisting F features from Q class, M neurons can be chosen as input neurons for middle layer. The outputs of middle layer neurons have relative values for each class. In competition procedure, the neuron with maximum value will win and identify relative class and the others will lose [7] and show the fault amount. Winner neuron's output shows true classified amount and in ideal condition its value has to be 1. Figure 1 shows the structure of the competitive neural network.

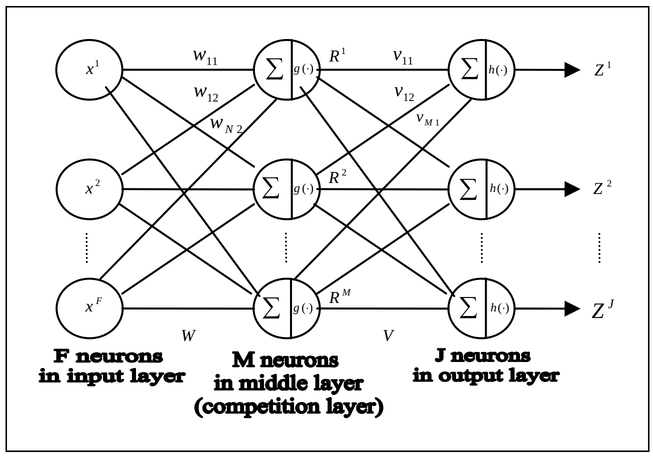


Fig.1 - *Schematic of competitive neural network*

Mathematical relation of designed network in figure 1 is:

$$\begin{aligned} net^m &= \sum_{n=1}^{N} w_{mn}^T x^f \\ R^m &= g(net^m) \\ net^j &= \sum_{m=1}^{M} v_{mj}^T R^m \\ Z^q &= h(net^j) \end{aligned} \tag{1}$$

Where net^j represents the net value of the j'th neuron and R^m, Z^q are output values of middle and output layers respectively. Input data are normalized in the range [0, 1] and the threshold values of all neurons are considered to be zero.

This type of NNT is applied to predict the data of condition monitoring for industrial machines and the stock market's data for stock prediction as case studies. The simulation results are studied and analyzed.

NOMENCLATURE

MLP	Multi Layer Perceptron
RBF	Radial Basis Function
NNT	Neural Network
FD-SLP	Fault Diagnosis - Single Layer Perceptron
IDP	Iran Data Process
IMI	Iran Marin Industry
PCA	Principle Component Analysis
PDF	Probability density function (pdf)

1. Fault diagnosis neural network

In a fault diagnosis NNT the maximum value of outputs of middle layer has to be found because in ideal condition for specific class of input, the output of layer has to be clean from fault to determine the class. It means that the output of activation function of winner neuron has to be 1 and the others have to get to 0. This structure can be extended to k winner neurons in competition layer as candidates for different classifications. This methodology forms reliable network and increases the degree of freedom as winner neurons. k is an arbitrary number and equation 2 indicates limitation which has to be considered for choosing k.

$$\binom{M}{k} \geq J \qquad (2)$$

Where $\binom{M}{k} = \dfrac{M!}{(M-k)!k!}$ are k commitments of M neurons. M is the number of neurons of middle layer, k is arbitrary value to determine the number of winner neurons in middle layer and J is the number of classified data.

By the explanation, relative weights from input layer to winner neurons of competition layer must train rewardingly and rest of the weights should be punished by generated fault in order to reduce error in next iteration of training. Activation functions of neurons are sigmoid and can be defined by equation (3) [2].

$$a_m^{(layer)} = \frac{1}{1+\exp(-\alpha \cdot net_m)} \qquad (3)$$

The value of $a_m^{(layer)}$ in equation (3) represents whether neuron 'm' is the source of faults.

1.1. Learning procedures in fault diagnosis neural network

Here we introduce a new way of training for fault diagnosis neural network in 6 steps:

STEP1) *Consider N input data with F features and consider weights of W and V for neural network to be random variables in the range of [0, 1].*

STEP2) *Calculate all outputs of neurons* $a_m^{(q)}$ *for competition layer. Here q is considered as middle layer.*

STEP3) *Find k winners of m neurons from maximum order value of* $a_m^{(q)}$ *in competition layer.*

STEP4) *As mentioned before we need to calculate source of faults (* δ *). Winners should be rewarded:*

$$\delta_{1:i,m} = 1 - f(w_{mi}^{(q)}) \quad \text{where } i = 1,...,F \;\&\; m = k \text{ neurons of } m \text{ as Max} \qquad (4)$$

And Losers should be punished:

$$\delta_{1:i,m} = -f(w_{mi}^{(q)}) \quad \text{where } i = 1,...,F \;\&\; m = (M-k) \text{ neurons as Not} - Max \qquad (5)$$

Here δ *is the source of fault.*

STEP5) *Calculate new relative weights (W) between layer q-1 & q by released equation [1]:*

$$W(new) = W(old) + \mu \cdot \delta \cdot f'(W) \qquad (6)$$

Where M defines the number of middle layer's neurons, F is number of input layer's neurons, μ *is normalizing factor and* $f'(\cdot)$ *is first derivation of activation function.*

STEP6) *Continue step3 through step5 until the predetermined condition is satisfied.*

2. Second layer of NNT needed factor for design

By using the procedure which presented in section 1.1, the output of fault diagnosis neural network forces us to consider k neurons as winner neurons. As we will see in our two test plants (Condition monitoring data for industrial machine and stock market's data) we have to train network to learn our required output. So we need to add second layer of NNT to satisfy our goal. Here we consider Multi Layer Perceptron (MLP) for implied network. Training equation for weights V in fig.1 is [3]:

$$V_{jm}(new) = V_{jm}(old) - \mu \cdot \frac{\partial E}{\partial V_{jm}} \qquad (7)$$

$$\begin{aligned} \text{Where: } \frac{\partial E}{\partial V_{jm}} &= 2 \cdot \sum_{j=1}^{J} (Z^j - T^j) \cdot \frac{\partial Z^j}{\partial V_{jm}} \\ \frac{\partial Z^j}{\partial V_{jm}} &= h'(net^j) \cdot \frac{\partial net^j}{\partial V_{jm}} \\ \frac{\partial net^j}{\partial V_{jm}} &= R^m \\ \Rightarrow \frac{\partial E}{\partial V_{jm}} &= 2 \cdot \sum_{j=1}^{J} (Z^j - T^j) \cdot h'(net^j) \cdot R^m \end{aligned} \qquad (8)$$

Where T is the desired output of system.

3. Case studies

3.1. Condition Monitoring of Industrial Machines

Condition monitoring is one of the important ways of maintenance in industrial machines. Oil analyzing is applied to condition monitoring. Oil cycle in industrial machines is like a blood cycle in human being's body. Because oil is in contact with mechanical surfaces of machines, we can get sample of the oil to get information of systems interior.

3.1.1. Oil analysis technique's stages

Stages of oil analysis technique are:

1) Getting sample of oil from different sides of mechanical unit.
2) Putting information on state of unit and type of oil used
3) Sending oil sample to laboratory
4) Doing critical experiments on sampled oil
5) Registering knew information
6) Primary analyzing of experiments by experts
7) Approving final analysis of experiments
8) Registering approved results
9) Sending back rules of experiments to sender
10) Looking forward to notations of laboratory

As it is seen, condition monitoring is based on Tacit Knowledge of experts in laboratory. Some logics and equalities are in expert's mind but for some times there exist difficult conditions and expert act only by his own experiences. This is why there are no strict rules to monitor the conditions. There exists three ways for maintenance where third is more economical [5]:

1) Breakdown Maintenance
2) Regular Preventive Maintenance
3) Condition Based Maintenance

3.1.2. Oil analysis data formation

Experts in laboratory classify each data to 5 states. Table 1 shows these states:

Table.1 - Five classes of oil

Code	Category	Output Class
5	Strict Condition	A
4	Boundary-Fast Condition	B
3	Boundary Condition	C
2	Boundary-Accepted Condition	D
1	Normal Condition	E

And every data has fifteen features:

Table.2 - Features of Data

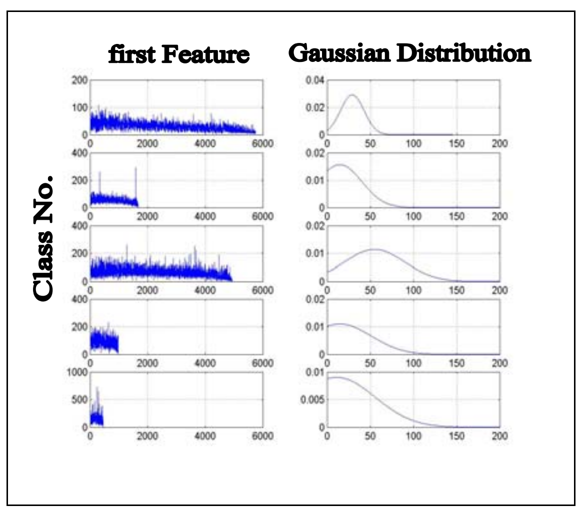
Feature number	Name	Type
1	Fe	Erosive Elements
2	Cr	
3	Al	
4	Cu	
5	Pb	
6	Sn	
7	Ni	
8	Ti	
9	Mo	
10	Ag	
11	Si	Pollutant
12	Na	
13	B	
14	V	
15	PQ	

Number of data used to train the network is about 19000 where 14300 are for training and 4700 for test procedure [5].

3.1.3 Oil analysis data

Principle Component Analysis (PCA) finds a mapping between the original feature spaces to a lower dimensional feature space. In some applications it might be desired to pick a subset of the original features rather then finding a mapping that uses all of the original features. The benefits of finding this subset of features could be in cost of computations of unnecessary features, cost of sensors (in physical measurement systems) [8].

Studying which feature of data is more dominant to the system is important. By using PCA analysis we found that first feature from fifteen features has up to 92.5% of information to classify [4]. Figure 2 shows first feature's of Normal Distribution for five classes:



a	b
c	d
e	f
g	h
i	j

Fig.2 - a) Data of first class **b)** First's class data pdf **c)** Data of second class **d)** Data of second class **e)** Data of third class **f)** Data of third class **g)** Data of forth class **h)** Data of forth class **i)** Data of fifth class **j)** Data of fifth class

The Gaussian distribution is used to compare correlations between 5 classes in first feature.
Correlation is defined by common areas in PDF distribution. As it can be clearly seen in figure 2 there are correlations between all of five classes. Two classes, one and three have the most amounts of data in average. Table 3 shows the number of data gathered for different classes.

Table.3 - Number of data in each class

	Class 1	Class 2	Class 3	Class 4	Class 5
N	5800	1800	5000	1000	700

Before simulation we can guess that these two classes (1&3) will get more training time (iterations) and will saturate the weights of network. We can solve this problem by two methods. First we can delete some data from class 1 and 3, but it is not wise because we risk deleting some useful information which occurs most of the time in this plant. Second we can change the learning rate for each class. By considering equation 8 we see that (Z-T) is directly related to equation. This means each class of data reacts in weights directly. Hence more data in a class causes more effect in weights. We can control this learning procedure by considering relative amount for ideal output of T for each class in place of amount 1.

Table.4 - Represents ideal output to reach every output of output layer neurons

		Class Number				
Ideal Output		1	2	3	4	5
	T^1	0.5	0	0	0	0
	T^2	0	2	0	0	0
	T^3	0	0	0.7	0	0
	T^4	0	0	0	2	0
	T^5	0	0	0	0	2

3.1.4 Simulation of Fault Diagnosis-Single Layer Perceptron (FD-SLP) network

We consider the network mentioned in introduction where the first layer is Fault Diagnosis and the second layer is Single Layer Perceptron. The training algorithm for this network is mentioned in sections 1&2. The learning rate of first layer is $\mu_1 = 0.001$ and that of the second layer is $\mu_2 = 0.06$. In these work only ten iterations for training was good enough to learn the NNT.

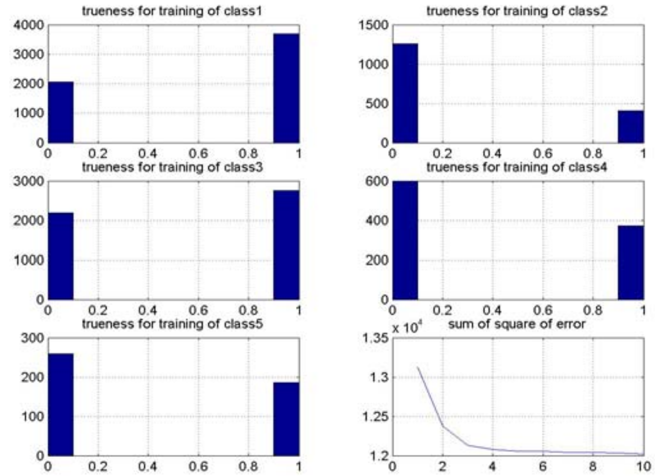


Fig.3 - Simulation result for FD-SLP

Figure 3 shows the result for training each class, it is seen that sum of square errors is decreasing faster than MLP and RBF Methods released in section 3.1.5. All of 5 classes are trained, where comparing to section 3.1.5, here three more classes are trained.

0.6380 →	Class 1
0.2509 →	Class 2
0.5145 →	Class 3
0.3940 →	Class 4
0.3124 →	Class 5

Fig.4 - Truth values of training for each of 5 classes

W&V weights of network are learned. 3D graph of weights before and after learning is good field to compare:

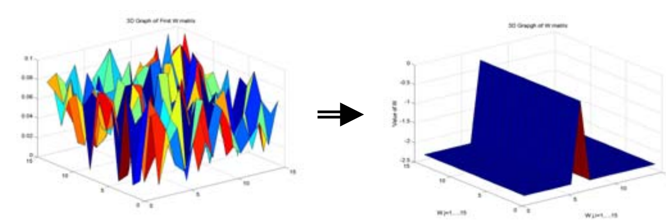


Fig.5 - 3D graph of W weights before and after learning procedure

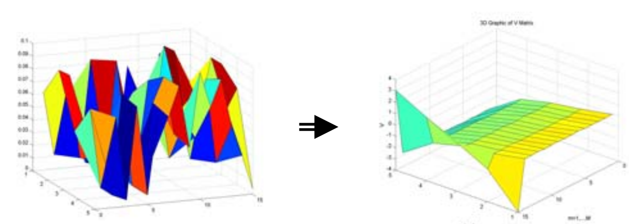


Fig.6 - 3D graph of V weights before and after learning procedure

After training we have to test system by test data.

Figure 7 shows near 61% truthness of testing and calculated confusion matrix which shows amount of true and false data classified in every class:

$$\mathit{Confusion\ Matrix}: \begin{bmatrix} 1560 & 280 & 375 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 346 & 275 & 1057 & 167 & 22 \\ 5 & 8 & 197 & 139 & 84 \\ 1 & 0 & 10 & 12 & 33 \end{bmatrix}$$
$$\mathit{Average\ Correct\ Classification\ Rate}: 60.82\%$$

Fig.7 - Confusion matrix and truth ness of testing procedure for FD-SLP

Confusion matrix is actually an out looking for matrix which detects the truth values of classification in test period. As it is seen in figure 7, for 1560 data of first class has been detected correctly and 280 data has classified in class 2. For class 2 we have no classification but the network has learned class 2 about 25%. In comparison by MLP&RBF, classes 2&3 are more classified.

3.1.5 MLP and RBF network's result

Here we present multi layer perceptron (MLP) and radial basis function (RBF) results and compare them with FD-SLP.
For MLP training period we have:

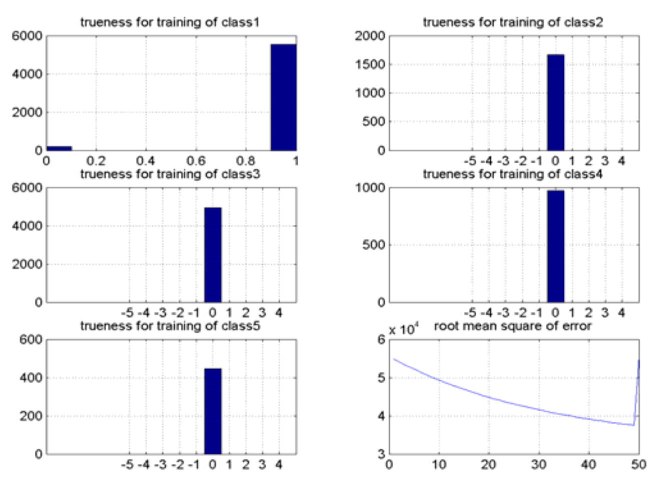


Fig.8 - MLP Training period

As it seen MLP only trains the first class and error is reduced to 39000 in 50'Th iteration. By testing, only the first class has been classified by a truth value of 42%:

$$\mathit{Confusion\ Matrix}: \begin{bmatrix} 1912 & 563 & 1639 & 333 & 139 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\mathit{Average\ Correct\ Classification\ Rate}: 41.69\%$$

Fig.9 - Confusion matrix and truthness of testing procedure for MLP

And in RBF network training period is:

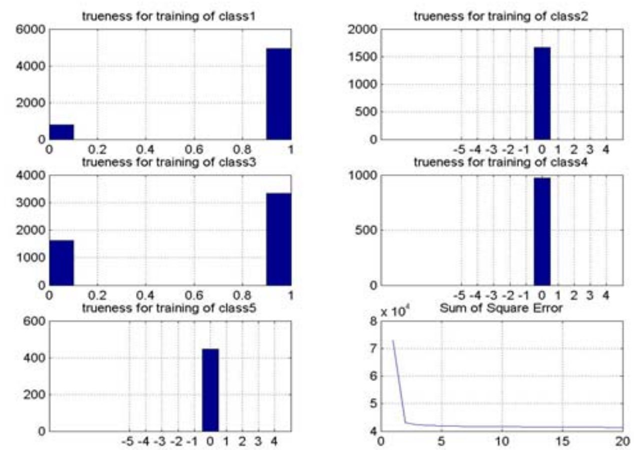


Fig.10 - RBF Training period

As it is seem in figure 10 error has been reduced to 41000 in 20 iterations and only first and third class are trained. By testing we get a truth value of about 60%:

$$\mathit{Confusion\ Matrix}: \begin{bmatrix} 1660 & 344 & 527 & 26 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 252 & 219 & 1112 & 307 & 139 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\mathit{Average\ Correct\ Classification\ Rate}: 60.44\%$$

Fig.11 - Confusion matrix and truthness of testing procedure for RBF

3.1.6 Comparing FD-SLP with MLP and RBF

As it is seen classification error in MLP and RBF was 39000 and 41000 where in FD-SLP it is reduced to 12000.
In the next step MLP has trained only for first class and RBF is trained for first and third classes but FD-SLP has been

trained for all classes having large correlation in 5 classes. The reason that FD-SLP couldn't classify second class because of its large correlation with first class, as shown in figure 2.

4. Stock predictions

Stock prediction is one of the attractive fields. All predictors know the difficulties of plan. It is nonlinear and unpredictable in normal condition because it is very sensitive to all stock sellers and buyers and even it is very sensitive to politic and policies of different societies.

4.1 Stock's data properties

Each stock has 12 features of registered properties [6]:

1) Volume of stock
2) Value of stock
3) Amount of stock
4) Maximum price
5) Minimum price
6) Average price
7) First price
8) Closed price
9) Changed price
10) Best buy
11) Best sell
12) Number of Purchasers

For example Iran's Data Processes (it's a stock name) stock has 232 working days through September 25, 2005. Closed price (feature 8) of the stock is shown in figure 12:

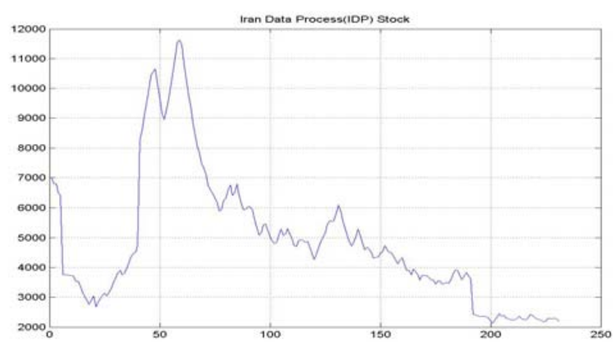


Fig.12 - IDP (Iran Data Process) Stock

As it is seen, two "break down" prices happened in 40th and 193rd working day. This is because of increasing amount of stock. This is important for balancing data before break and after break. An equation must be applied to prices before simulation which is to balance the Volume of stock and Value of stock. Fortunately it's enough to process on 3 months data because actually on every 100 of working days, rules of stock exchange are changing some how. We find this from processing different numbers of working days. This reasoning is definable by knowing that most of the people change their type of stock in about every three month and even other polices come to the game by interring knew dealers.

4.2 Identify the type of output to predict

Goal of designing this system is to predict future features of stock. The important aim of the goal is to predict 8'Th feature of stock which is the "Closed price". Long life view is important key to solve the problem. It means that if we train our network to learn some future days it's better to learn only for next day. So our ideal output of network will be some future working days of "Closed price" and input data of network is about 100 working days with 12 features as mentioned in section 4.1.

4.3 Designing FD-SLP network for Stock exchange market and results

Here we designed FD-SLP network to train to predict future prices. We used two types of stocks to show the powerfulness of FD-SLP:

I) Iran Data Process (IDP) stock
II) Iran Marine Industry (IMI) stock

We used error back propagation method and considered a soft learning rate about $\mu = 0.01$, with 2000 iterations of training, N=12 input neurons for 12 features, M=10 middle neurons and J=20 output neurons which means we want to predict 20 future date of work. Figure 13 shows train result of IDP stock and figure 14 is for IMI:

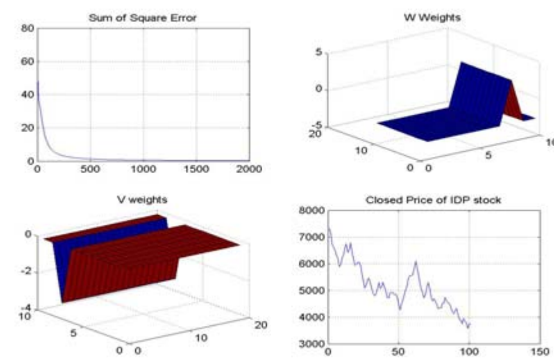


Fig.13 - Training results for IDP

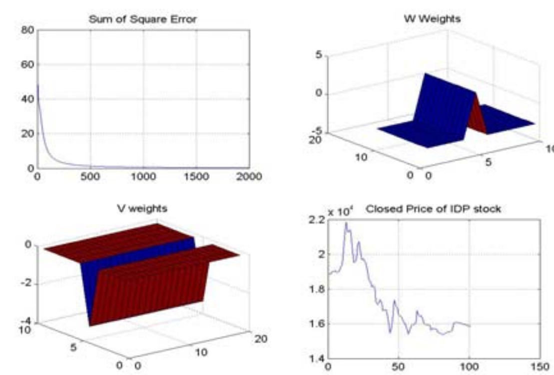


Fig.14 - Training results for IMI

In IDP training period the sum of square error is reduced from 60.0143 to 0.2900 and in IMI reduced from 60.0146 to 0.2895. Now we test the trained neural network for 20 future working days, results are shown in table 5. As it is mentioned previously, long time prediction of system is an important key to determine future's prices. We considered 20 days for future and as it is seen in table 5 the prices predicted for 20th day of two stocks are very near to real values. Price difference for IDP is 24.2 and for IMI is 280. Error of prediction is 0.65% and 1.76% for IDP and IMI respectively.

Table.5 - Results of test period

Future Days	IDP Real Closed Price	IDP Predict Closed Price	IMI Real Closed Price	IMI Predict Closed Price
1	4521	3704.8	15411	15594
2	4739	3704.8	15445	15594
3	4605	3704.8	15487	15594
4	4523	3704.8	15535	15594
5	4507	3704.8	15545	15594
6	4368	3704.8	15574	15594
7	4227	3704.8	15612	15594
8	4124	3704.8	16035	15594
9	4255	3704.8	16082	15594
10	4323	3704.8	16086	15594
11	4107	3704.8	16078	15594
12	3902	3704.8	16044	15594
13	3901	3704.8	16059	15594
14	3753	3704.8	15982	15594
15	3940	3704.8	15966	15594
16	3861	3704.8	15942	15594
17	3759	3704.8	15951	15594
18	3575	3704.8	15866	15594
19	3744	3704.8	15874	15594
20	**3729**	**3704.8**	**15874**	**15594**

5. Conclusion

FD-SLP (Fault Diagnosis-Single Layer Perceptron), introduced in this paper, is a suitable tool (network) to classify Oil analyzed data. FD-SLP is simple and fast and can be trained in only about 10 iterations (see figure 3).

FD-SLP can classify data having high correlations. FD-SLP is also powerful neural network to apply to stock exchange market because of predicting stock price in hard predicting conditions. The network is sensitive for output values of middle layer's neurons and it has to be tried several times to get best answer. Because of non-linearity of stock data, network should train itself with datum for many iterations, here 1000 iterations is good enough to proceed the learning (see figure 13).

REFERENCES

[1] S. Khanmohammadi *, I. Hassanzadeh and H. R. Zarei Poor. Fault diagnosis competitive neural network prioritized modification rule of connection weights. ELSEVIER Artificial Intelligence in Engineering 14 (2000) 127-132.
(*: Corresponding Author)

[2] S. Khanmohammadi, I. Hassanzadeh and H. R. Zarei Poor & R. M. Mathur. Fault Diagnosis Probabilistic Neural Network with β distributed Weights of Connection.

[3] "Neural Network for Pattern Recognition". Christopher M.Bishop. Clarendon Press Oxford. 1995. Chapter 4, Multi Layer Perceptron.

[4] "Patter Recognition". Sergios Theodoridis & Konstantinos Koutroumbas. Academic Press. 1999. Chapter 5, Feature Selection.

[5] Iranian Laboratory for Condition Monitoring By Oil Analyses Press.

[6] http://www.tsesc.com as a stock market Data Reference.

[7] Joachim Buhmann. Hans Kuhnel. Unsupervised and Supervised Data Clustering with Competitive Neural Networks. IEEE 1992.

[8] Ira Cohen, Qi Tian, Xiang Sean Zhou, Thomas S. Huang. Feature Selection Using Principal Feature Analysis, IEEE ICIP 2002.