Application of Artificial Neural Network In Cathodic Protection of Carbon Steel Pipe In Sea Water

Sami Abulnoun Ajeel* **Ghalib A. Ali****

Received On :7/3/2007
Accepted On:7/10/2007

Abstract

The intelligent techniques are used successfully in a broad band of applications one of these applications is the cathodic protection system. Examples of these techniques used in cathodic protection are fuzzy logic and genetic algorithms. The present work aims to use the neural network to predict the minimum current density required in impressed current cathodic protection to protect low carbon steel pipe which have been related previously.[1].

This work deals with choosing the best network architecture for cathodic protection system. This step used multilayer feed forward network four environment variables (concentration C%, temperature T, distance D and pH) as input to identify the minimum current density as output in a feed forward network structure with one hidden layer using the practical results data for the learning process. The best number of neurons in the hidden layer is chosen by trial and error and it is found to be 25 neurons. the decision function used is the tan training algorithm with one variable learning rate. Then, neural network training is done using 25 data samples from the experimental data for the current density in the above four variables conditions. The stopping criterion for training was to obtain a sum square error of 0.001 or read 10000 Epochs. An (SSE) than 0.001 were obtained after 5226 Epochs.

Generalization test used 5 data samples taken from the experimental results other than those data samples used in the learning process to check the performance of the neural network on these data. The SSE for these samples was 0.0053 and it shows a good generalization results for our application. The comparison between the actual experimental output and the neural network out put after the learning process are almost identical which indicates that good learning process was achieved.

The practical results indicate that neural network system can be used successfully to obtain minimum cathodic protection current density to protect low carbon steel pipes.

تطبيق الشبكات العصبية الاصطناعية في الحماية الكاثودية لانبوب من الصلب الكاربو

الخلاصة

استخدمت التقنيات الذكية في نطاق واسع وبنجاح وأحدى تطبيقاتها هي فيمجـالالحمايـة الحماية الكاثودية بتقنية الخوارزمية الوراثية.

ان هدف البحث الحالي هو استخدام الكاثودية و ل ل الشبكة العصبية لتحديد أوطئكثافـةتيـار مطلـوب للحماية الكاثودية بطريقة التيار المسلط من خلال استخدام بيانات التيار المسلط بالحماية الكاثودية لأنبوب من الفولاذ المنخفض الكاربون التي تم الحصول عليها من بحثسـابقلنـا فـيهـذا المجال.

يتناول هذا البحث اختيار أفضل شبكة بناء لنظام حماية كاثودية. تستخدم في هذهالخطـوة شبكة تغذية أمامية متعددة الطبقات لاربعة متغيرات بالوسط (تركيز c% ، درجةالحـرارة T ، المسافة D ودرجة الحامضية pH) كمعلومات داخلة لتحديد أوطئ كثافة تيار كنيجة منالنظـام

https://doi.org/10.30684/ etj.26.8.2
University of Technology-Iraq, Baghdad, Iraq/2412-0758

في شبكة مبنية بتغذية أمامية مع طبقة مخفية واحدة مستخدمة نتائج عملية لعملية التعليم للشبكة. افضل عدد من الاعصاب المستخدمة في الطبقة المخفية المختارة بطريقة المحاولة والخطأ ووجدت مساوية الى 25 عصيبة. دالة القرار المستخدمة هي لوغارتيم تدريب الظل مع متغير واحد كمعدل تعليم.

بعدها عمل تدريب للشبكة العصبية باستخدام بيانات 25 نموذج من البيانات العملية لكثافة التيار في الظروف المتغيرات الاربعة اعلاه. ان مقياس التوقف للتدريب كان للحصول على مجموع مربع الخطأ(0.001)او الوصول10000 أيكو ومجموع مربع الخطأ هو اقل بالمقارنة مع (0.001) تم الحصول عليه بعد 5226 ايكو .

اختبار التعميم استخدم 5 نماذج اخذت من نتائج عملية غير تلك النماذج المستخدمة في عملية التعلم للتاكد من اداء الشبكة العصبية على هذه البيانات ومجموع مربع الخطأ لهذه النماذج كان 500053 حيث أعطت نتائج لقيم جيدة في تطبيقنا هذا.

المقارنة بين النتائج الفعلية الناتجة والنتائج من الشبكة العصبية بعد عملية التعليم كان على الاغلب متماثل حيث يشير الى تحقيق عملية تعليم جيدة وناجحة. تشير النتائج العملية بان استخدام نظام الشبكات العصبية ناجح للحصول على اوطئ كثافة تيار حماية كاثوديه لحماية انابيب من الصلب الكربوني المنخفض.

1. Introduction

Corrosion is an electrochemical reaction based on universal laws of nature. All metallic structures corrode, even steel, which is a man made substance produced from iron oxide.

Cathodic protection (CP) is perhaps the most important of all approaches to corrosion control. By means of an externally applied electric current, corrosion is reduced virtually to zero, and a metal surface can be maintained in a corrosion environment without deterioration for an indefinite time.[2].The cathodic protection can be defined as an effective electric method of corrosion control on metallic structures exposed to an electrolyte, namely soils and liquids[3,4]. Cathodic protection converts all active anode sites (area that corrode) on the structure into cathode sites that do not corrode. It is important to understand that corrosion is only mitigated on the surface of the metallic structure in contact with the electrolyte.[4].

Ajeel (1) studied cathodic protection system for low carbon steel pipe. This system was used to investigate the influence of various conditions on the minimum cathodic protection current that would provide a full cathodic protection for steel tube immersed in sea water. The variable conditions studied are concentration of (0.01 – 3.5) % NaCl, temperature (30 – 50 C^0), distance between pipe (cathode) and graphite electrode (anode) of (10 - 20) cm and PH solution of (5.0 – 9.0) using a selected range of these conditions, the experimental results indicated that the cathodic protection current density increases with increasing temperature, concentration and PH respectively. The current density also slightly increases with an increase in distance between cathode and anode.

Intelligent control is now becoming a common tool in many engineering and industrial applications .It has the ability to comprehend and learn about plants, disturbances, environment, and operating conditions[7,8] . Some examples of the factors to be learned are plant characteristics such as its static and dynamic behaviors. [5,6] .

Artificial neural network, with their self-organizing and learning ability, are now used as promising

tools for such purposes. The architecture and functions of the artificial neural network are based on the biological brain. Neural network provides a different computing architecture compared with the Von Neumann computers. The main characteristics of neural network are parallel and distributed in nature as well as self-organization, however, conventional computers have series, local , and algorithmic properties [7,8] .

The neural network can learn static or dynamic properties autonomously based on the past history of measurement data and then act in away such that a better solution can be obtained under unknown environment conditions. However, conventional computers have to be programmed before data can be processed and they cannot work beyond the decision given by the program.Therefore, knowledge- based engineering has not been well accepted in real applications since it has no solution when it has to make adecision under new environments,[6].

The cathodic protection method for corrosion prevention requires identification of the minimum current density that gives the full corrosion protection with the presence of certain environment variables. The neural network can be used to identify this minimum current density taking the environment variables as input and using the practical results data for the learning process. This can be done by creating a mathematical model for the process and choosing the best neural network architecture, decision function and learning algorithm for this application.

2. Neural Network

2.1. Artificial Neural Network Classifications

Artificial Neural Networks (ANNs) can be viewed as weighted directed graphs in which artificial neurons are nodes and direct edges (with weights) are connections between neuron inputs based on the connection pattern (architecture)[9].

i - Feed-back

Feed-back networks are the type of neural that contains feedback connection, and they are called "recurrent" [10]. Recurrent networks recalculate previous outputs back to inputs hence, output is determined both by their current input and their previous outputs. For this reason, recurrent network can be regarded very similar to short- term memory in humans in that the state of the network outputs depends: upon their previous input.

The Hopfield model is the simplest and most widely used feedback neural architecture. Another example of feedback network is Boltzman machine which is close to Hopfield model architecture.

ii-Feed – forward

The feed —forward network is called "Non-Recurrent" and it has no feedback connection that connects through weights expended from the output of layer to the inputs of the same or previous layer.

iii- Self – organization

The self organization network monitors itself and corrects errors without any external intervention.

2.2. The Mathematical Model of the Neuron

The mathematical model of the neuron by McCulloch-Pitts [11] is described as:

Tansig function

$$f(x) = 1/(1+ \exp(-x)) \quad \ldots(1)$$

Where f(x) is an output function or activation function of a neuron "also called transfer function". This equation is showns in Figure (1).

2.3. The Back Propagation Training Algorithm

Training the BP requires the following steps [12]:

1. Set the weights and thresholds to small random values for all the neurons.
2. Put the I/P vector into the I/P layer, and specify the desired O/P layer.
3. Use the Sigmoid non-linearity to calculate the O/P of the hidden and O/P layers.
4. Calculate the error ($Y_d - Y_0$).
5. Calculate δ_k the O/P layer and the changes in the weighs Δw_{jk}.
6. Work back to the hidden layer, calculate δ_j for the hidden layer and Δ_{ij} .
7. Repeat by going to step 2.

Multi Layer Feed Forward network [MLFF] .The MLFF consists basically of three layers , input layer , output layer and one or more hidden layers as shown [MLFF] in Figure (2) . The MLFF have four inputs these are (concentration C%, temperature T, distance between cathode and anode D and pH solution), and it have one output which is the predicted cathodic protection current density.

3. Experimental Procedures

The data related previously for impressed current cathodic protection of low carbon steel pipe are shown in Table (1).[1]The first step was choosing the best neural network architecture for this application; this step used four variables (concentration C%, temperature T, distance D and PH) to identify the current density in a feed forward neural network structure with one hidden layer.The best number of neurons in the hidden layer is chosen by trial and error and it is found to be 25 neurons.The decision function used is the tan-training algorithm with one variable learning rate. The second step was the neural network training. This is done after obtaining the experimental results for the current density in different conditions (concentration, temperature, distance between cathode and anode & PH).

The experiment results were used to train the neural network which have been constructed and trained using 25 data samples form the experimental data and 5 samples were used for generalization test of the trained neural network.The third step was the generalization test for the neural network. The generalization test means to test the neural network on data samples other than those data samples used in the learning process and to check the performance of the neural network on these data. In our case, we used 5 data samples taken form the experimental results for the generalization test. Figure (3) shows the steps for neural network learning. This neural network was simulated using the scientific and engineering package MATALB® 6.5.

4. Results and Discussion

4.1 Neural Network Results

4.1.1 Neural Network Training Results

The training of the neural network was repeated many times using different number of neurons in the hidden layer. The stopping criterion was to obtain an sum square error (SSE) of 0.001 or reach 10000 training cycles (Epochs). Figure (4-A) shows a plot of the reducing SSE against the epochs number. An SSE less than 0.001 were obtained after 5226 Epochs. Also figure (4-B) shows the variation of the learning rate against epoch.

These results were obtained using 25 neurons in the hidden layer which gave the best results. Figure (5) shows a comparison between the actual

experimental output (red) and the NN output (blue) after the learning process and it is clear that the two are almost identical which indicates that a good learning process was achieved.

4.1.2 Neural Network Generalization Results

Generalization of neural network is the process of describing the whole from the some of the parts, moving from a specific to the general case. In the present work, 5 samples form the experimental data were tested using the trained neural network. Figure (6) shows a comparison between the actual data and the output of the neural network for the 5 samples excluded form the training. The SSE for theses samples was 0.0053 and it shows a good generalization result for our application.

5. Conclusion

1. The neural network can be used in cathodic protection system to obtain the current density for different application conditions, and this was obvious from the generalization test.
2. In the neural network of cathodic protection system , the back propagation with variable learning rate training algorithm gives faster convergence and reaches the required SSE faster than other training algorithms.
3. The neural network gives accurate results and is more flexible in terms that it can adapt it self with the change of the environment variables.
4. The sequence effects of temperature , concentration , distance between anode and cathode and pH solution on cathodic protection current density is as follows :

   Temperature > concentration > pH solution > distance between anode and cathode.

6. References

[1] Sami,A. Ajeel and Ghalib A. Ali,"variable conditions on polarization parameters of impressed current cathodic protection of low carbon steel pipes" under the process of publishing.

[2]Uhlig.H.H., Winston Revie. R, Corrosion and Corrosion Control , John Wiley and Sons, 1985.

[3]Wilson Waton,"Cathodic Protection",Marpo Company,05 June 2001 www.marpo.org.

[4]Stephen.K.lower,"Electrochemistry",SimonFraserUniversity,1999 www.chemi.com.

[5]White, D.a. and D.A Sofge," Handbook of Intelligent control " Van Nostrand Reinhold , New York, 1992.

[6]Miller, W.T. R.S , Sutton , and P.J Werbos, " Neural Networks for Control" MIT Press, Cambridge,MA,1990 .

[7]Rumethart, D.E. et al. ," parallel Distributed processing Exploration in the Micro Structure of Cognition Vol.1: Foundation ", MIT press , Cambridge , MA 1968 .

[8]Dayhoff, J.E," Neural Network Architectures: An Interdiction ", Van Nostrand Reinhold, New York , 1990.

[9]Ishii S.; Sato M.., " Constrained neural approaches to quadratic assignment problems " , Neural Network , Vol. 11 , PP. 1073- 1082 , 1998 .

[10]Jack A.K. , " Artificial neural network . A tutorial " , IEEE Vol. 18 , PP. 231-243 , 1996 .

[11]McCulloch-pitts, http://www.matworks.com" System Identification Using Neural networks " , 2001.

[12]H.A Zorada, "Artificial neural network" , Printice – Hall , INE , 1996 .

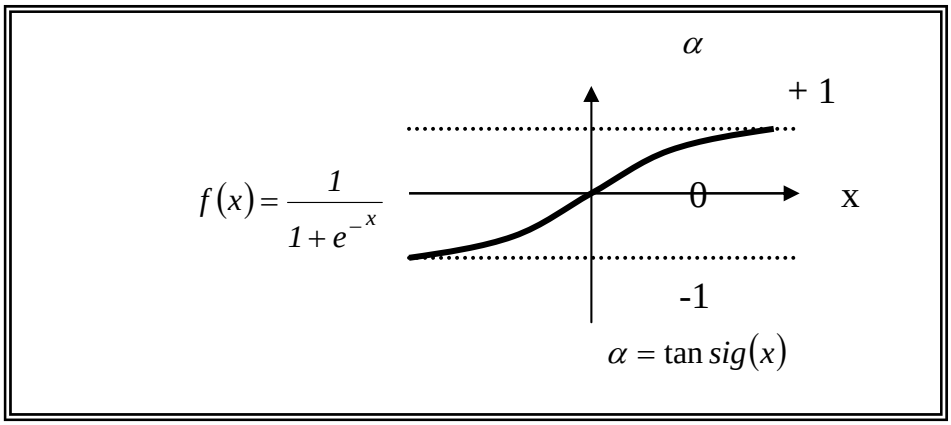

Figure (1) The tan-sigmoid transfer function.

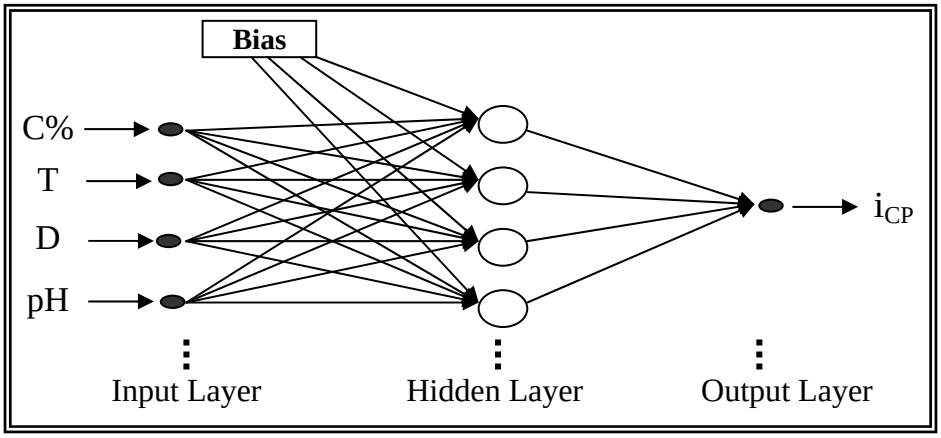

Figure (2) Multi layer feed forward

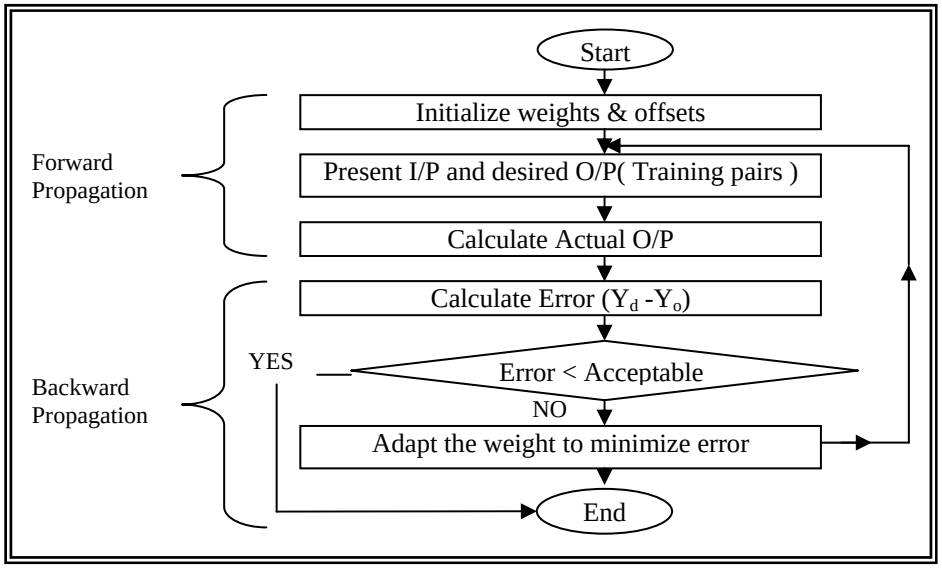

Figure (3) The back-propagation training steps [12].

Table (1) The conditions for the calculation of i_{cp} at E_{cp}(-850 mV).

No.	NaCl C(%)	T(°c)	D(cm)	pH	i_{cp} ($\mu A/cm^2$)
1	3.5	30	10	9.0	26.648
2	2	30	20	8.5	24.840
3	1	30	15	7.0	24.091
4	0.1	30	10	6.5	24.222
5	0.01	30	10	5.0	23.553
6	0.01	30	20	5.0	23.919
7	3.5	35	15	9.0	33.129
8	2	35	20	8.5	32.726
9	1	35	10	7.0	30.761
10	0.1	35	20	6.5	29.918
11	0.01	35	15	5.0	29.228
12	0.1	35	10	7.0	29.001
13	3.5	40	20	9.0	40.787
14	2	40	15	8.5	39.267
15	1	40	10	7.0	38.825
16	0.1	40	15	6.5	37.511
17	0.01	40	10	5.0	36.922
18	1	40	20	7.0	39.115
19	3.5	45	10	9.0	45.949
20	2	45	20	8.5	44.713
21	1	45	15	7.0	43.883
22	0.1	45	20	6.5	43.125
23	0.01	45	15	5.0	42.000
24	2	45	10	8.5	44.018
25	3.5	50	10	9.0	52.641
26	2	50	15	8.5	51.405
27	1	50	10	7.0	50.912
28	0.1	50	15	6.5	49.416
29	0.01	50	20	5.0	48.026
30	3.5	50	20	9.0	53.157

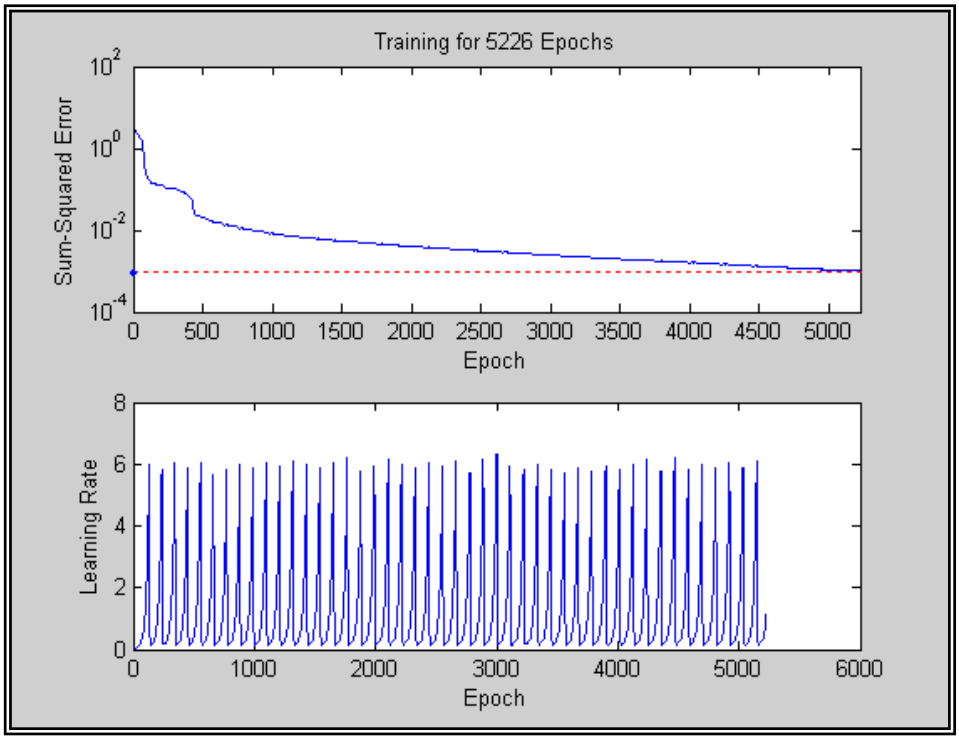

Figure (4) A-Sum-Squared Error vs. Epoch, B-learning rate ζ

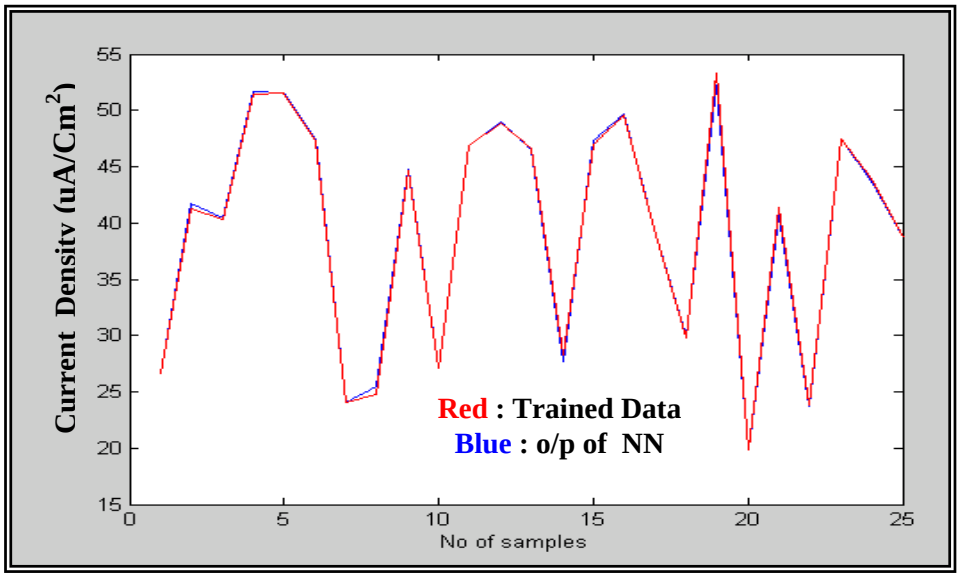

Figure (5) The comparison of the output of the neural network with the experimental output.

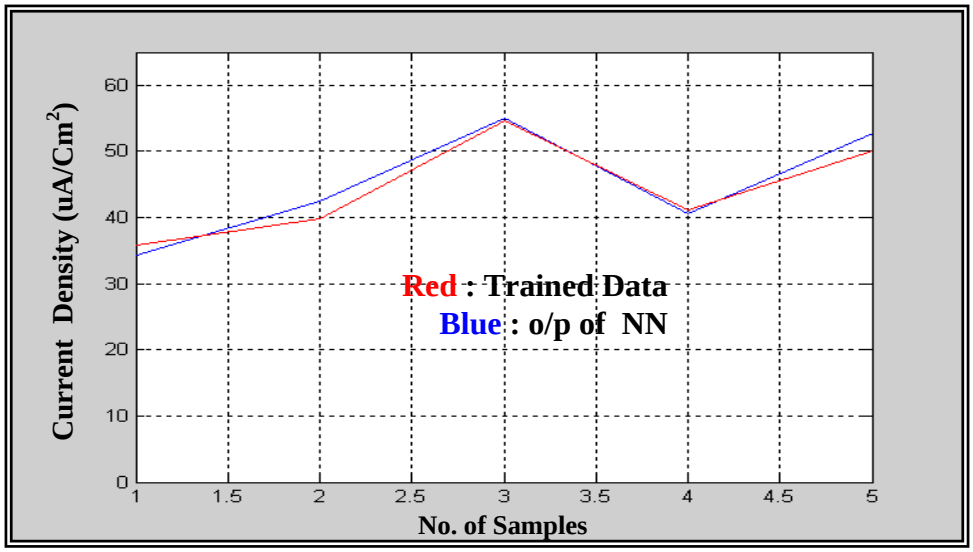

Figure (6) Generalization results

% NEURAL NETWORK TRAINING PROGRAM

```
p2 = [ 1 0.6      0.5      1; 0.002857143 0.8      0.5      0.74057971; 0.028571429
       0.7      1        0.768115942;0.285714286      0.9      0.75
       0.869565217;  0.571428571      1        0.5      0.942028986;
0.571428571    0.9      1        0.942028986; 0.285714286      0.6      0.75
       0.869565217; 0.002857143      0.6      0.5      0.74057971; 0.028571429
       0.8      0.75     0.768115942; 1 0.7      0.75     1; 1     1        1
       1; 0.571428571 0.9      0.5      0.942028986; 0.285714286      0.8      1
       0.869565217; 0.028571429      0.6      0.5      0.768115942;
0.002857143    0.9      0.75     0.74057971; 0.002857143      1        1
       0.74057971; 1  0.9      0.5      1; 0.285714286 0.7      0.5
       0.869565217; 0.028571429      1        0.75     0.768115942;
0.571428571    0.6      1        0.942028986; 0.028571429      0.7      0.5
       0.74057971; 0.002857143      0.6      1        0.768115942; 1 1
       0.5      1; 0.285714286 0.8      0.5      0.869565217; 0.571428571
       0.7      1        0.942028986; 0.002857143      0.7      0.75
       0.74057971; 1  0.8      1        1; 0.285714286 1        0.5
       0.869565217; 0.571428571      0.8      0.75     0.942028986;
0.028571429    0.9      1        0.768115942 ]
inp=inp2(4:25,:);
% Input 4 variables
d2= [
0.487692392
0.755147234
0.73787083
0.940777072
0.943686975
0.865046394
0.44089603
```

```
0.453011475
0.817481378
0.496495306
0.858640947
0.895151992
0.85437675
0.5164986
0.860196556
0.907779872
0.712816383
0.544664263
0.974762541
0.363097308
0.757288483
0.435918083
0.867773284
0.803883531
0.708735199
0.656320346
0.728152852
1
0.753408613
0.915539613
]
d = d2(4:25);
% Output ( desired )
p=inp' % Patterns = Input
t=d' ; % Target = desired = output
s1=25; % Number of Hidden neurons
[w1,b1,w2,b2]=initff(p,s1,'logsig',t,'logsig'); % Initialization of the Weights and
bias
Lr=.1; % Learning rate
Ep=10000;  % Training cycles = Epoch
Eg=.001; % Error goal = stopping criteria
tp=[Lr Ep Eg nan nan nan nan nan];
[w1,b1,w2,b2,ep,tr]=trainbpx(w1,b1,'logsig',w2,b2,'logsig',p,t,tp); % Training
function
figure;
pause % struck any key to see plot of error
w11=[w1 b1];
w22=[w2 b2];
w111=w11';
w222=w22';
plotter(tr,Eg);  % Plot the SSE against the Error goal
pause
  r = simuff(p,w1,b1,'logsig',w2,b2,'logsig'); % Recall Function to calculate the
Output of NN after training and them we can compare it with the real data
  r=r*54.641;
```

```
plot(r);
hold on
t=t*54.641;
plot(t,'r');
xlabel('time');
ylabel('output');
title('neural response');
```

%GENERALIZATION PROGRAM

```
clear all
NNTWARN OFF
inp = [ 1 0.6     0.5     1; 0.002857143 0.8     0.5     0.74057971; 0.028571429
        0.7     1       0.768115942;0.285714286     0.9     0.75
        0.869565217;0.571428571     0.9     1       0.942028986;
0.285714286     0.6     0.75    0.869565217; 0.002857143     0.6     0.5
        0.74057971; 0.028571429     0.8     0.75    0.768115942; 1 0.7
        0.75    1; 1    1       1       1; 0.571428571 0.9     0.5
        0.942028986; 0.285714286     0.8     1       0.869565217;
0.028571429     0.6     0.5     0.768115942; 0.002857143     0.9     0.75
        0.74057971; 0.002857143     1       1       0.74057971; 0.285714286
        0.7     0.5     0.869565217; 0.028571429     1       0.75
        0.768115942; 0.571428571     0.6     1       0.942028986;
0.028571429     0.7     0.5     0.74057971; 0.002857143     0.6     1
        0.768115942; 1 1        0.5     1;  0.5714285710.7      1
        0.942028986; 0.002857143     0.7     0.75    0.74057971; 1   0.8     1
        1; 0.285714286 1        0.5     0.869565217; 0.571428571     0.8
        0.75    0.942028986; 0.028571429     0.9     1       0.768115942 ]
% Input 4 variables
d= [
0.487692392
0.755147234
0.73787083
0.940777072
0.865046394
0.44089603
0.453011475
0.817481378
0.496495306
0.858640947
0.895151992
0.85437675
0.5164986
0.860196556
0.907779872

0.544664263
0.974762541
0.363097308
```

```
0.757288483
0.435918083
0.867773284
0.708735199
0.656320346
0.728152852
1
0.753408613
0.915539613
]% Output ( desired )
p=inp' % Patterns = Input
t=d' ; % Target = desired = output
s1=40; % Number of Hidden neurons
[w1,b1,w2,b2]=initff(p,s1,'logsig',t,'logsig'); % Initialization of the Weights and
bias
Lr=.1; % Learning rate
Ep=2000;  % Training cycles = Epoch
Eg=.001; % Error goal = stopping criteria
tp=[Lr Ep Eg nan nan nan nan nan];
[w1,b1,w2,b2,ep,tr]=trainbpx(w1,b1,'logsig',w2,b2,'logsig',p,t,tp); % Training
function
pause %struck any key to see plot of error
w11=[w1 b1];
w22=[w2 b2];
w111=w11';
w222=w22';
plotter(tr,Eg);  % Plot the SSE against the Error goal
pause
  r = simuff(p,w1,b1,'logsig',w2,b2,'logsig'); % Recall Function to calculate the
Output of NN after training and them we can compare it with the real data
  plot(r);
  hold on
  plot(t,'r');
  xlabel('time');
  ylabel('output');
  title('neural response');
pause
  p1= [  0.571428571    1         0.5       0.942028986; 1 0.9       0.5       1;
0.285714286    0.8       0.5       0.869565217];
  p11=p1'
  gg=simuff(p11,w1,b1,'logsig',w2,b2,'logsig');
  gg=gg*54.641
  ggg=[0.943686975 0.712816383 0.803883531] * 54.641
  figure
  plot (gg)
  hold on
  plot ( ggg, 'r')
  grid
```