



# PREDICTION OF ZINC CONSUMPTION AS SACRIFICIAL ANODE IN CATHODIC PROTECTION OF STEEL IN SEA WATER USING ARTIFICIAL NEURAL NETWORK

APRAEL S. YARO

Chemical Engineering Department – College of Engineering – University of Baghdad

## ABSTRACT

Corrosion has gained special attention due to its significance, when predicting corrosion rates. However, the complexity and variability makes it hard to model its effects. This study evaluates the usefulness of Artificial Neural Networks (ANN) to predict the corrosion rate as a function of several factors which have been related in previous studies to the protectiveness of low carbon steel in sea water, i.e. Temperature, Flow rate, pH, and time. Results showed that neural networks are a powerful tool and that the validity of the results is closely linked to the amount of data available and the experience and knowledge that accompany the analysis. Statistical analysis showed that the proposed correlation has an Average Absolute Relative Error (AARE) of 0.09% and Standard Deviation (S.D) 0.46%

## KEYWORDS

artificial neural networks, corrosion prediction, cathodic protection

## الخلاصة

ان ظاهرة التآكل تكتسب اهمية خاصة عندما يتعلق الأمر بالتنبؤ بمعدلات التآكل. من ناحية ثانية ان التعقيدات و المتغيرات اثناء حدوثها تجعل من الصعب نمذجة تأثيراتها. هذه الدراسة تقيم فوائد شبكة الأعصاب الاصطناعية للتنبؤ بمعدلات تآكل الحديد الكربوني في عملية الحماية الكاثودية في ماء البحر و بدلالة عدد من المتغيرات كدرجة الحرارة، سرعة جريان ماء البحر و حامضيته اضافة الى زمن تعرض النموذج للوسط. أظهرت النتائج ان تقنية شبكات الأعصاب الاصطناعية اداة فعالة و ان صحة النتائج تربط و تعبر بقوة بين كمية النتائج المتوفرة و الخبرة المعرفة المرافقة لتحليل النتائج. التحليل الاحصائي اظهر ان العلاقة المقترحة في هذه الدراسة افرزت معدل الخطأ النسبي المطلق يساوي 0.09% و الأتحراف القياسي يساوي 0.46% .

## INTRODUCTION

Corrosion is an electrochemical process in which a current leaves a structure at the anode site, passes through an electrolyte and reenters the structure at the cathode site. Current flows because of a potential difference between the anode and cathode that is the anode potential is more negative than the cathode potential, and the difference is the driving force for the corrosion current. The total system—anode, cathode, electrolyte and metallic connection between anode and cathode is termed a corrosion cell <sup>1</sup>.

There are many methods for corrosion control as illustrated some of them in the <sup>2</sup>:

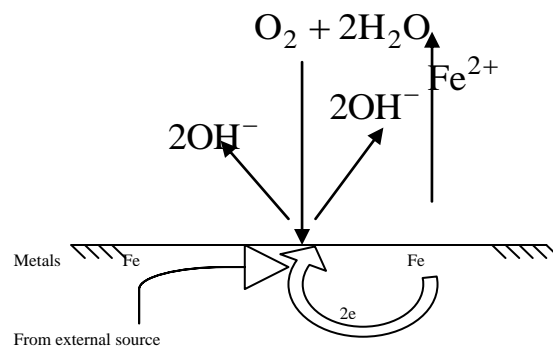
1. Cathodic protection.
2. Anodic protection.
3. Protective coating such as paint.
4. Corrosion-resistant metals and alloys.
5. Addition of inhibitors.
6. Very pure metals.

The selecting method depends on many factors such as cost, availability, contamination of environment with corroding metal ... etc.

Cathodic protection is unique amongst all the methods of corrosion control in that if required it is able to stop corrosion completely, but it remains within the choice of the operator to accept a lesser, but quantifiable, level of protection. Manifestly, it is an important and versatile technique. In principle, cathodic protection can be applied to all the so-called engineering metals.

## CATHODIC PROTECTION PRINCIPLES

It is possible to envisage what might happen if an electrical intervention was made in the corrosion reaction by considering the impact on the anodic and cathodic reaction. For example, if electrons were withdrawn from the metal surface it might be anticipated that the anodic reaction would speed up (to replace the lost electrons) and cathodic reaction would slow down because of the existing shortfall of electrons. It follows that the rate of metals consumption would increase. By contrast, if additional electrons were introduced at the metal surface, the cathodic reaction would speed up (to consume the electrons) and the anodic reaction would be inhibited; metal dissolution would be slowed down. This is basis of cathodic protection. <sup>3</sup>



**Fig.1** Schematic illustration of partial cathodic protection of steel in an aerated environment



Artificial neural networks (ANNs) are simplified models of central nervous system. The network of highly inter connected neural computing elements that have the ability to respond to input stimuli and learn to adapt to environment.<sup>4</sup> As the term of artificial neural networks implies early work in the field of neural networks centered on modeling the behavior of neurons found in the human brain, engineering systems are considerably less complex than the brain, hence from an engineering view point ANN can be viewed as non linear empirical models that are especially useful in representing input-output data. Making predication, classifying data, reorganization patterns, and control process. ANN which will be referred to as a node in this work and is analogous to a single neuron in the human brain. The advantages of using artificial neural network in construct with first principles models or other empirical models.<sup>5</sup>

1. ANN can be highly non linear.
2. The structure can be more complex and hence more representative than most other empirical models.
3. The structure does not have to be prespecified.
4. Quite flexible models.

### THEORY AND MODELING OF ANN

Artificial Neural Networks (ANNs) have been increasingly applied to many problems in transport planning and engineering, and the feed forward network with the error back propagation learning rule, usually called simply Back propagation (Bp), has been the most popular neural network.<sup>6</sup>

Back propagation networks are among the most popular and widely used neural networks because they are relatively simple and powerful. Back propagation was one of the first general techniques developed to train multi-layer networks, which does not have many of the inherent limitations of the earlier, single -layer neural nets criticized by Minsky and Papert. These networks use a gradient descent method to minimize the total squared error of the output. A back propagation net is a multilayer, feed forward network that is trained by back propagating the errors using the generalized Delta rule.<sup>7</sup>

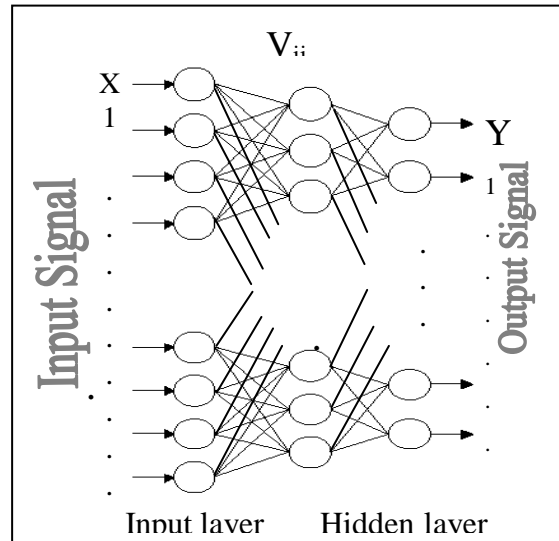
The input is the input to the hidden layer and the output layer is the output from the immediate previous layer, so it is called feed forward neural network. The number of the input units and the output units are fixed to a problem, but the choice of the number of the hidden units is somehow flexible as shown in figure 2. Too many hidden units may cause over fitting, but if the number of hidden units is too small, the problem may not converge at all. Usually a large number of training cases may allow more hidden units if the problem requires so.<sup>8</sup>

The conventional algorithm used for training a MLFF is the Bp algorithm, which is an iterative gradient algorithm designed to minimize the mean-squared error between the desired output and the actual output for a particular input to the network.<sup>9</sup>

Basically, Bp learning consists of two passes through the different layers of the network: a forward pass and backward pass as shown in figure 2.

During the forward pass the synaptic weights of the network are all fixed. During the backward pass, on the other hand, the synaptic weights are all adjusted in accordance with an error-correction rule.<sup>10</sup>

A MLFF consists of layers of interconnected denoted as the input layer, the hidden layer and the output layer. So the number of neurons in the hidden layer can be varied based on the complexity of the problem and the size of the input information.<sup>10</sup>



**Fig.2** Multi- layer feed forward network (one hidden layer).

Two learning factors that significantly affect convergence speed as well as accomplish avoiding local minima, are the learning rate and momentum.

The learning rate ( $\eta$ ) determines the portion of weight needed to be adjusted. However, the optimum value of  $\eta$  depends on the problem. Even though as small learning rate guarantees a true gradient descent, it slows down the network convergence process. If the chosen value of  $\eta$  is too large for the error value, the search path will oscillate about the ideal path and converges more slowly than a direct descent.

The momentum ( $\alpha$ ) determines the fraction of the previous weight adjustment that is added to current weight adjustment. It accelerates the network convergence process.

During the training process, the learning rate and the momentum are adjusted to bring the network out of its local minima, and to accelerate the convergence.

The algorithm of the error back-propagation training is given below<sup>9</sup>:

Step1: initialize network weight values.

Step2: sum weighted input and apply activation function to compute output of hidden layer.

$$h_j = f[\sum_i x_i w_{ij}] \quad (1)$$

Where

$h_j$ : The actual output of hidden neuron  $j$  for input signals  $X$ .

$x_i$ : Input signal of input neuron ( $i$ ).

$w_{ij}$ : Synaptic weights between input neuron hidden neuron  $j$  and  $i$ .



$f$ : The activation function.

Step3: sum weighted output of hidden layer and apply activation function to compute output of output layer.

$$O_k = f \left[ \sum_j h_j W_{jk} \right] \quad (2)$$

where

$O_k$ : The actual output of output neuron  $k$ .

$W_{jk}$ : Synaptic weight between hidden neuron  $j$  and output neuron  $k$ .

Step4: compute back propagation error.

$$\delta_k = (d_k - O_k) f' \left( \sum_j h_j W_{jk} \right) \quad \text{where} \quad (3)$$

$f'$ : The derivative of the activation function.

$d_k$ : The desired of output neuron  $k$ .

Step5: calculate weight correction term.

$(n-1)$

$$\Delta W_{jk}(n) = \eta \delta_k h_j + \alpha \Delta W_{jk}(n-1) \quad (4)$$

Step6: sums delta input for each hidden unit and calculate error term.

$$\delta_j = \sum_k \delta_k W_{jk} f' \left( \sum_i X_i W_{ij} \right) \quad (5)$$

Step7: calculate weight correction term.

$$\Delta W_{ij}(n) = \eta \delta_j X_i + \alpha \Delta W_{ij}(n-1) \quad (6)$$

Step8: update weights.

$$W_{jk}(n+1) = W_{jk}(n) + \Delta W_{jk}(n) \quad (7)$$

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n) \quad (8)$$

Step9: repeat step2 for given number of error.

$$MSE = \frac{1}{2p} \left[ \sum_{p,k} (d_k^p - O_k^p)^2 \right] \quad (9)$$

where

$p$ : The number of patterns in the training set.

Step10: END.

Bp is easy to implement, and has been shown to produce relatively good results in many applications. It is capable of approximating arbitrary non-linear mappings. However, it is noted that two serious disadvantages in the Bp algorithm are the slow rate of convergence, requiring very long training times, and getting stuck in local minima.

The success of Bp methods very much depends on problem specific parameter settings and on the topology of the network [Leonard 1990]. So in the next section the quick propagation will be presented.

## MODELING CORRELATION OF ANN

In the current study, neural networks are used to fit a set of experimental points in order to provide a purely empirical model. The experimental points are called the training cases (or learning cases) and another are called testing cases. They consist of input vectors (values of input variables) associated with the experimental output value. To solve a problem with a back-propagation network, it is shown sample inputs with the desired outputs, while the network learns by adjusting its weights. If it solves the problem, it would have found a set of weights that produce the correct output for every input. This work includes computer simulation, implemented on a Pentium 4 computer using MATLAB, version 7.

The modeling of ANN correlation began with the collection of large data bank following the learning file was made by randomly selecting about 70% of the data base to train the network. The remaining 30% of data is then used to check the generalization capability of the model. The last step is to perform a neural correlation and to validate it statistically. So that the steps of modeling are:-

**Collection of Data:** The first step in neural network modeling is collection of data. The data is necessary to train the network and to estimate its ability to generalize. In this model about 256 experimental points have been collected for corrosion rate used cathodic protection of low carbon steel in sea water [Khalid, 2006]. The data were divided into training and test sets: the neural network was trained on 70% (180) of the data and tested on 30% (76). The data includes a large range of temperature, flow rate, pH, and time. All of these parameters are input to neural network and there is one output; it is the corrosion rate.

## THE STRUCTURE OF ARTIFICIAL NEURAL NETWORK

In this work, a multilayer neural network has been used, as it is effective in finding complex non-linear relationships. It has been reported that multilayer ANN models with only one hidden layer are universal approximates. Hence, three layer feed forward neural network are chosen as a correlation model. The weighting coefficients of the neural network are calculated using MATLAB programming. Structure of artificial neural network built as:-

**1. Input layer:** A layer of neurons that receive information from external sources and pass this information to the network for processing. These may be either sensory inputs or signals from other systems outside the one being modeled. In this work four input neurons in the layer and there is a set of (180) data points available of the training set.

**2. Hidden layer:** A layer of neurons that receives information from the input layer and processes them in hidden way. It has no direct connections to the outside world (inputs or output). All connections from the hidden layer are to other layers within the system. The

number of neuron in the first hidden layer consists of nine neurons and the second hidden layer consists of sixteen neurons. This gave best results and was found by trial and error. If the number of neurons in the hidden layer is more, the network becomes complicated. Results probably indicate that, the present problem is not too complex to have a complicated network routing. Hence, the results can be satisfactorily achieved by keeping the number of neurons in hidden layer at a best value with two hidden layer.

**3. Output layer:** A layer of one neuron that receives processed information and sends output signals out of the system. Here the output is the corrosion rate or Zn consumption as sacrificial anode in Cathodic protection of a steel in sea water.

**4. Bias:** The function of the bias is to provide a threshold for activation of neurons. The bias input is connected to each of hidden neurons in network.

The structure of multi layer ANN modeling is illustrated in figure.3

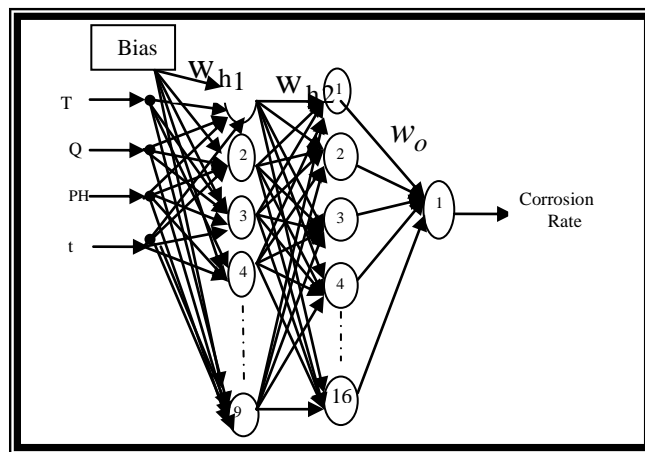


Fig.3 Structure of a layer neural network

## TRAINING OF ARTIFICIAL NEURAL NETWORK

Training is just the procedure of estimating the values of the weights and establishing the network structures and the algorithm used to do this is called a “learning” algorithm. Learning typically occurs through training or exposure to set of input, output data where the training algorithm iteratively adjusts the connection weights. These connection weights represent the knowledge necessary to solve specific problems (i.e. calculates the coefficients of correlation).

The training phase starts with randomly chosen initial weight values. Then a back-propagation algorithm is applied after each iteration, the weights are modified so that the cumulative error decreases. In back-propagation, the weight changes are proportional to the negative gradient of error. Back-propagation may have an excellent performance, this algorithm is used to calculate the values of the weights the following procedure is then used (called “supervised learning”) to determine the values of weights of the network:-

1. For a given ANN architecture, the value of the weights in the network is initialized as small random numbers.
2. The input of the training set is sent to the network and resulting outputs are calculated.

3. The measure of the error between the outputs of the network and the known correct (target) values is calculated.
4. The gradients of the objective function with respect to each of the individual weights are calculated.
5. The weights are changed according to the optimization search direction.
6. The procedure returns to step 2.
7. The iteration terminates when the value of the objective function calculated using the data in the test approaches experimental value. The trial and error to find the best ANN correlation model shown in table 1:-

**Table (1)**  
Network parameters in ANN model

Structure	network Parameters				
	MSE	No.of iteration	Learning rate	Momentum coefficient	Transfer function
[4-16-1]	0.1	1340	0.7	0.9	Tan sigmoid
[4-18-1]	0.01	7654	0.65	0.9	Tan sigmoid
[4-9-16-1]	0.00001	929	0.9	0.9	Tan sigmoid

With reduced MSE the network is more accurate, because MSE is defined as:-

$$\text{MSE} = \frac{1}{2p} \left[ \sum_p \sum_k (d_k^p - o_k^p)^2 \right] \quad (10)$$

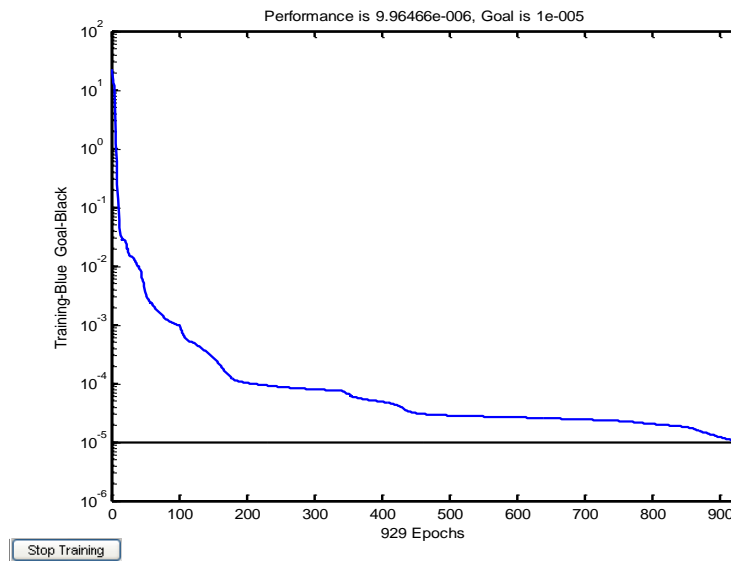
Where p: the number of patterns in training set  
k=No. of iterations.

$d_k^p$  =The desired output.

$o_k^p$  =The actual output.

Learning rate: the main purpose of the learning rate is to speed up the rate at which the network learns. This is also accomplished by multiplying the learning rate by the change in weight factor from the previous iteration in order to determine the new weight factors.





**Fig.4** illustrates the number of epochs with MSE for corrosion rat

weight factors by multiplying by the change in the weight factor that was produced in last iteration.

The choice of transfer function is based on the function of the network being used. The tan-sigmoid function is appropriate for most types of network, especially prediction problems.

The learning process includes the procedure when the data from the input neurons is propagated through the network via the interconnections. Each neuron in a layer is connected to every neuron in adjacent layers. A scalar weight is associated with each interconnection.

Neurons in the hidden layers receive weighted inputs from each of the neurons in the previous layer and they sum the weighted inputs to the neuron and then pass the resulting summation through a non-linear activation function (tan sigmoid function).

Artificial neural networks learn patterns for this study can be equated to determining the proper values of the connection strengths (i.e. the weight matrices  $W_{h1}$ ,  $W_{h2}$  and  $w_o$  illustrated in figure.3 that allow all the nodes to achieve the correct state of activation for a given pattern of inputs. The matrix, bias, and vector, given eq. (11), (12), (13) and (14) illustrate the result of coefficient weights for ANN correlation for this case where  $w_h$  is the matrix containing the weight vectors for the nodes in the hidden layer and  $w_o$  is the vector containing the weight for the nodes in the output layer.

$$W_{h1} = \begin{pmatrix} -0.2101 & -0.0522 & 0.5464 & 6.8392 \\ 0.0313 & 0.0006 & -0.2814 & 0.4433 \\ 0.2087 & -0.0050 & 1.0017 & -2.1666 \\ -0.0202 & -0.0010 & 0.3037 & 0.1873 \\ -0.0025 & -0.0077 & 0.7551 & 1.5166 \\ -0.0307 & -0.0023 & 0.2219 & -0.5138 \\ -0.1467 & -0.0169 & -3.8696 & 4.9871 \\ 0.0190 & 0.0001 & -1.7795 & 0.5599 \\ 0.0079 & 0.0009 & -0.0318 & -0.9718 \end{pmatrix} \dots (11)$$

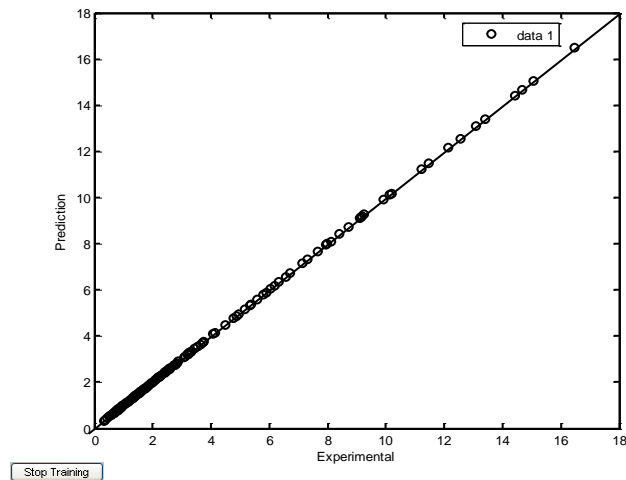
$$b = \begin{pmatrix} -4.9377 \\ -1.6042 \\ -1.7661 \\ -1.0162 \\ -5.5346 \\ 1.0628 \\ 8.9828 \\ 1.1397 \\ 0.2220 \end{pmatrix} \dots(12)$$

$$W_{h2} = \begin{pmatrix} 0.0406 & -0.3100 & -0.8888 & -0.3131 & -2.4295 & 0.5815 & 1.5191 & 2.2825 & -0.7296 \\ 8.7740 & -1.6365 & 1.5967 & -4.3531 & 1.2756 & -2.2288 & -1.5553 & -0.9756 & 7.6799 \\ 0.4378 & -0.4767 & 0.4341 & -1.2425 & -0.1427 & -0.8395 & -0.9642 & 1.6665 & -1.9187 \\ -3.5163 & -0.3959 & 1.7808 & -0.0079 & 3.7486 & -11.0608 & -1.7049 & -3.1292 & -0.6846 \\ -3.1893 & -1.5956 & 0.5030 & -1.9342 & 1.2678 & -3.5760 & -0.4841 & 4.5798 & 0.8413 \\ -0.2755 & -0.0541 & -4.7363 & -0.8619 & -6.6302 & 0.3583 & -5.6679 & -1.0520 & -3.6661 \\ 0.3525 & -2.6604 & -4.5104 & 1.6472 & -1.3579 & 0.3411 & -4.5446 & 0.3707 & 0.2862 \\ -1.6356 & -0.1007 & 2.6733 & -0.8482 & -3.0927 & 3.1922 & 2.1168 & -1.1799 & -1.2983 \\ -1.3712 & 0.0500 & 1.0391 & 4.8353 & 1.8285 & -1.3331 & 0.4950 & 0.0743 & 1.8176 \\ -6.8859 & -3.5244 & 6.0626 & -3.1854 & -3.5642 & -8.2899 & 1.0237 & 4.4005 & -1.8944 \\ -0.6302 & -1.4180 & -4.4862 & -1.1765 & -0.4018 & 0.1284 & -0.5081 & -0.3942 & -0.5212 \\ -0.2282 & 1.6494 & 0.0727 & -4.0256 & -1.6346 & 0.4222 & -0.9693 & 5.3954 & -6.9461 \\ 0.5453 & -0.2046 & -1.9745 & 0.5938 & -0.9798 & -0.0110 & 0.9162 & -0.9258 & -0.3238 \\ -0.0664 & -0.9405 & -1.2722 & 4.1727 & -8.6158 & 0.1517 & -3.5267 & 1.0009 & -10.2560 \\ 5.4313 & -0.1749 & 1.5971 & -4.8051 & -8.7707 & 7.6181 & -2.6448 & -0.4801 & -2.4320 \\ 0.0486 & 0.2388 & 0.3050 & 0.0095 & 0.9498 & -0.0550 & -0.2476 & 0.6169 & 0.3752 \end{pmatrix} \dots(13)$$

$$w_o = [4.3844 \quad -0.7402 \quad 2.7131 \quad 1.6743 \quad 1.8703 \quad 2.8731 \quad -1.7626 \quad 0.7479 \quad -2.5953 \quad 0.5427 \quad -1.5195 \quad 0.6907 \quad 4.0277 \quad 3.3468 \quad -0.6299 \quad 10.6241] \dots(14)$$

**RESULTS**

The network architecture used for prediction corrosion rate for cathodic protection of low carbon steel in sea water in fig.3 consist of four inputs neurons corresponding to the state variables of the system, with hidden neurons and one output neuron. All neurons in each layer were fully connected to the neurons in an adjacent layer. The prediction of ANN correlation result is plotted in figure.5 compares the predicted of corrosion rate with experimental for training set.

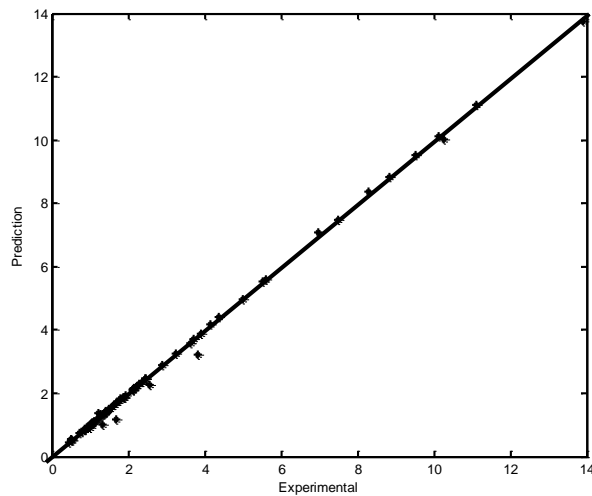


**Fig.5** Comparison between experimental and prediction for Zinc consumption in training set

### TEST OF THE PROPOSED ANN

The purely empirical model was tested on data that were not used to train the neural network and yielded very accurate predictions. Having completed the successful training, another data set was employed to test the network prediction corrosion rate. We made use of the same model to generate (76) new data values.

The result of prediction is plotted with experimental value as shown in figure.6.



**Fig(6)** Comparison between experimental and prediction for Zinc consumption in testing set.

## CONCLUSIONS

The ANN correlation shows noticeable improvement in the prediction of corrosion rate. The neural network correlation yields an AARE of 0.09% and standard deviation of 0.46%. The number of input units and output units are fixed to a problem, but the choice of the number of the hidden units is flexible. In this work it was started with a number of neurons in the hidden layer, but was found that two hidden layer the first layer consists of nine neurons and second consists of sixteen neurons gives better results of prediction corrosion rate shown in table(2). Below two layer of hidden the network cannot to be trained.

Table (2) Gives the statistical information of neural networks models for prediction of corrosion rate.

ANN models	Structure	AARE%	S.D%	R
Case study	[4-9-16-1]	0.09	0.46	0.998

## Acknowledgment

Partial support for this work was provided by Suhaiela A.Akkar.The author appreciate her valuable discussion.

## NOMENCLATURE

$d_k$  The desired of output neuron k

$f$  The activation function

$f'$  The derivation of the activation function

$h_i$  The actual output of hidden neuron j

$o_k$  The actual output of output neuron k

P The number of patterns in the training set

$V_{ij}$  Weight on link from  $X_i$  to  $Z_j$

$W_{ij}$  Synaptic weights between input and hidden neurons

$W_{jk}$  Synaptic weight between hidden and output neuron

$x_i$  Input signal of input neuron i

Y Target vector

$Y_k$  Output unit

$Z_i$  Hidden unit

## GREEK SYMBOLS

### SYMBOL

### DEFINITION

$\delta_k$  The error term

$\eta$  The learning rate



## REFERENCES

- Shrier[1],LL,(1976)“Corrosion 1,matal/environment reactions", newness-Butter Worth,
- Bosich, F, G.,(1970) “Corrosion prediction for practicing engineering”, Barnes and Noble INC.
- Uhlig,H.H.(2000) “Corrosion handbook”, Edited by Winston, R., R., second Edition, John-Wiley and son.
- Patterson,H.(1996) “Artificial neural networks theory and applications”, Prentice Hall.
- David M.H.(2002), “Applications of artificial neural networks in chemical engineering”, Korean. J. Chem. Eng., 17, 373-392.
- Freeman, J.A and Skapura, D.M.,(1992) “Neural networks”, M.sc.thesis,Jordan University of Science and Technology.
- MATLAB, Version 7, June 2003, “Neural network toolbox”
- Sivanadam, S.N.,(2003) “Introduction to artificial neural networks”, Vikas Publishing House Pvt. Ltd.
- Lendaris, G.,(2004) “Supervised learning in ANN from introduction to artificial intelligence”, NewYork, April 7.
- Lippmann,R.P.,(1987) “An introduction to computing with neural nets”, IEEE Magazine, April, pp.4-22.
- Khalid,W.,Hameed.(2006)”Cathodic protection of low carbon steel in sea water”, Ph.D. thesis, Iraq, University of Baghdad.

## APPENDIX

### STATISTICAL ANALYSIS

Statistical analysis based on the test data is calculated to validate the accuracy of the output for pervious correlation model based on ANN. The structure for each model should give the best output prediction, which is checked by using statistical analysis. The statistical analysis of prediction is based on the following criteria:-

1. The AARE (Average Absolute Relative Error) should be minimum

$$AARE = \frac{1}{N} \sum_{i=1}^N \left| \frac{x_{prediction} - x_{experimental}}{x_{experimental}} \right|$$

Where N=Number of data points.

x=Corrosion rate.

2. The standard deviation should be minimum.

$$S.D = \sqrt{\sum_{i=1}^N \frac{[(x_{\text{prediction}} - x_{\text{experimental}} / x_{\text{experimental}}) - AARE]^2}{N-1}}$$

3. The correlation coefficient R between input and output should be around unity.

$$R = \frac{\sum_{i=1}^N (x_{\text{experimental}(i)} - \bar{x}_{\text{experimental}})(x_{\text{prediction}(i)} - \bar{x}_{\text{prediction}})}{\sqrt{\sum_{i=1}^N (x_{\text{experimental}} - \bar{x}_{\text{experimental}})^2} \sqrt{\sum_{i=1}^N (x_{\text{prediction}} - \bar{x}_{\text{prediction}})^2}} \quad \bar{x}_{\text{experimental}} = \text{mean corrosion rate}$$

mean of experimental points

$\bar{x}_{\text{prediction}}$  = corrosion rate mean for prediction points