Application Of Artificial Neural Network Models For Predicting Total Dissolved Solids In Marsh Water

Dr. Mohammed D. Salman

College of Eng. / Mech. Eng. Dept. / University of Thi Qar

E-mail: Mohammed_selman2000@yahoo.com

ABSTRACT

In this paper an Artificial Neural Networks (ANNS) model is designed to predict the Total Dissolved Solids (TDS) concentration in marsh water. A previous data set are selected from previous studies which done on analysis of marsh water quality, these data are arranged in a format of five input parameters to feed forword back-propagation including the acidity (pH), calcium concentration (C), Magnesium Concentration (M) , Chloride Concentration (Cl) and Sulphate Concentration (S), and one output parameter as Total Dissolved Solids concentration. Artificial Neural Network used to study the effect of each parameter on TDS concentration in marsh water. Several structures of ANNs model is examined with different transfer functions, activation functions, number of neurons in each hidden layer and number of hidden layers. Results show that the two hidden layer network with transfer function (trainscg) with (12 & 10) neurons in the first and second hidden layer respectively and (tansig-tansig-purelin) gives the best performance (Mean Square Error: 3.05e-5) network for this prediction.

Keyword: Total Dissolved Solids, Neural Networks, Prediction, Marsh Water.

الخلاصة

في هذه الورقةِ صمم نموذج شبكات عصبيةً صناعية لتَخمين تركيز المواد الصلبة الذائبة الكلية في مياهِ الاهوار. أختيرت مجموعة معلومات سابقة مختارة مِنْ الدراساتِ السابقةِ التي عَملتْ على تحليلِ نوعيةِ ماءِ الاهوار ، رتبت هذه البيانات في صيغة خمسة عوامل داخلة للشبكة متضمنتا ذلك الحموضةِ (pH)، تركيز الكالسيومِ، تركيز المغنيسيومِ ، تركيز الكلوريدِ، وتركيز الكبريتاتِ، و عامل ناتجِ واحد متمثلا بتركيز المواد الصلبة الذائبة في مياه الاهوار. استعملت الشبكة العصبية الصناعية لدِراسَة تأثيرِ كُلّ من العوامل الداخلة على تركيز المواد الصلبة الذائبة في مياه الاهوار. اختبرت عِدّة تراكيب لنموذجِ الشبكة العصبية في مختلف : دوال النقلِ ، دوال التنشيطِ، عدد العقد في كُلّ طبقة مخفية وعدد الطبقاتِ المخفيةِ. اوضحت النَتائِجَ بأنّ الشبكةَ العصبية ذو الطبقتين المخفيتين بوظيفةِ النقلِ (trainscg) مَع (12 و10) عقد عصبية في الطبقةِ المخفيةِ الأولى والثانيةِ على التوالي و دالة تنشيط متمثلة بـ (tansig tansig purelin) تَعطي أفضل أداءِ (متوسط مربع نسبة الخطأ : $3,05 *10^{-5}$) للشبكة لهذا التنبؤ في هذه الدراسة.

Introduction:-

Iraqi marshes are one of the most ancient marshes in the world and pride themselves on its beautiful scenery and its rich natural ecological system. Closely involved in the lives of people over many years, marshes have been an important water resource and helped create and preserve distinctive culture. Lack of water resources and optimum management have been two recent challenges of water resources engineering [1]. Population growth, decrease of useable water resources, improvements in lifestyle, growing rate of consumption, climate change and several other parameters have caused useable water to be a significant problem for future. Economic and efficient use of water resources and its management have an increasingly significant role. Prediction of Total Dissolved solids(TDS) in water is one of the methods which have been recently considered for management of water resources. The predictions can be used for water resources planning and management in case they are of acceptable accuracy. There are two methodologies for prediction of TDS, like other water quality parameters,; first, precise study of different processes which can affect water salinity and developing statistical or deterministic models according to the obtained information. Second, developing Data Driven Models using information and collected data; In the latter technique, relationship between input and output data can be found using input data, but still physical understanding of phenomena is significant for having suitable input data for model [2,3].

The artificial neural network (ANN) technique is an artificial intelligence technique that attempts to mimic the human brain's problem solving capabilities. Artificial neural networks are capable of self-organization and learning; patterns and concepts can be extracted directly from historical data. In general, artificial neural networks can be applied to the following types of problems: pattern classification, clustering and categorization, function approximation, prediction and forecasting, optimization, associative memory, and process control. When presented with data patterns, sets of historical input and output data that describe the problem to be modeled, ANNs map the cause-and-effect relationships between the model input data and output data. This mapping of input and output relationships in the ANN model architecture allows developed models to be used to predict the value of the model output parameter, given any reasonable combination of model input data, with satisfactory accuracy[4].

This paper, artificial neural network model is used to represent the theoretical work, this model is coded using *MATLAB* (R2008a). Then '*MATLAB Function* program' is developed to call the model results (the correct weights and bases); this function is saved with the M-File basic functions of *MATLAB*. In order to simplifying this model for general use, graphical user interface is developed using *MATLAB* code, the result of this program is based on the '*MATLAB* Function program' which, in turn, uses neural network model result.

Neural network model is used to predict the Total Dissolved Solids in water which depends upon the results of the experimental work as training and testing data. The aim of the theoretical work is to obtain the best neural network model used to predict the overall mass transfer coefficient.

1. Artificial Neural Networks Model:

Theoretical work which is represented by ANN modelling. This program implements several different neural network algorithms, including back-propagation algorithm. The configuration and training of neural networks is a trial–and–error process due to such undetermined parameters as the number of hidden layers, the number of nodes in the hidden layers, the learning parameter, and the number of training patterns. An artificial neural network is developed to predict TDS. This section describes the data selection for training and testing patterns, the topology of the constructed network, the training process and the verification of the neural network results. The successful application of neural network to a problem depends on the problem representation and learning. Problem representation means the selection of a proper topology of the network.

The back propagation networks are most useful for problems involving forecasting and pattern recognition. Two subsets of data are used to build a neural network model: a training set and a testing set. The training phase needs to produce a neural network that is both stable and convergent. Therefore, selecting what data to use for training a network is one of the most important steps in building a neural network model. The training set is used for computing the gradient and updating the network weights and biases to diminish the training error, and find the relationship between the input and output parameters. Hence, the learning process is a crucial phase in NN modeling. The testing set is used to evaluate the generalization ability of the learning process. In this study the testing set contains approximately (20) % of total database. The parameters used in this study are shown in Table (1) [5,6,7,8]. The experimental values used to train the neural network as training data. The total number of

(140) test cases were utilized. The training set contains (112) cases and the testing set comprises of (28) cases.

1.1 Structure of Back propagation Neural Network:-

The nodes in the input layer and output layer are usually determined by the nature of the problem. The main difficulty in the structural identification of a complex nonlinear system arises from the huge amount of possible relationships among variables. The selection of outputs is straightforward and depends on the modeling goal. However, informed input-variable selection is critical to achieving efficient model performance [9].

In this study the (140) sample data are chosen for parameters which may be introduced as the components of the input vector consist of the pH, Calcium concentration (C), Magnesium Concentration (M), Chloride Concentration (Cl) and Sulfate Concentration. The output data is the Total Dissolved Solids Concentration (TDS) in marsh water Therefore, the nodes in the input layer and output layer are (5) and (1), respectively as shown in figure (1).

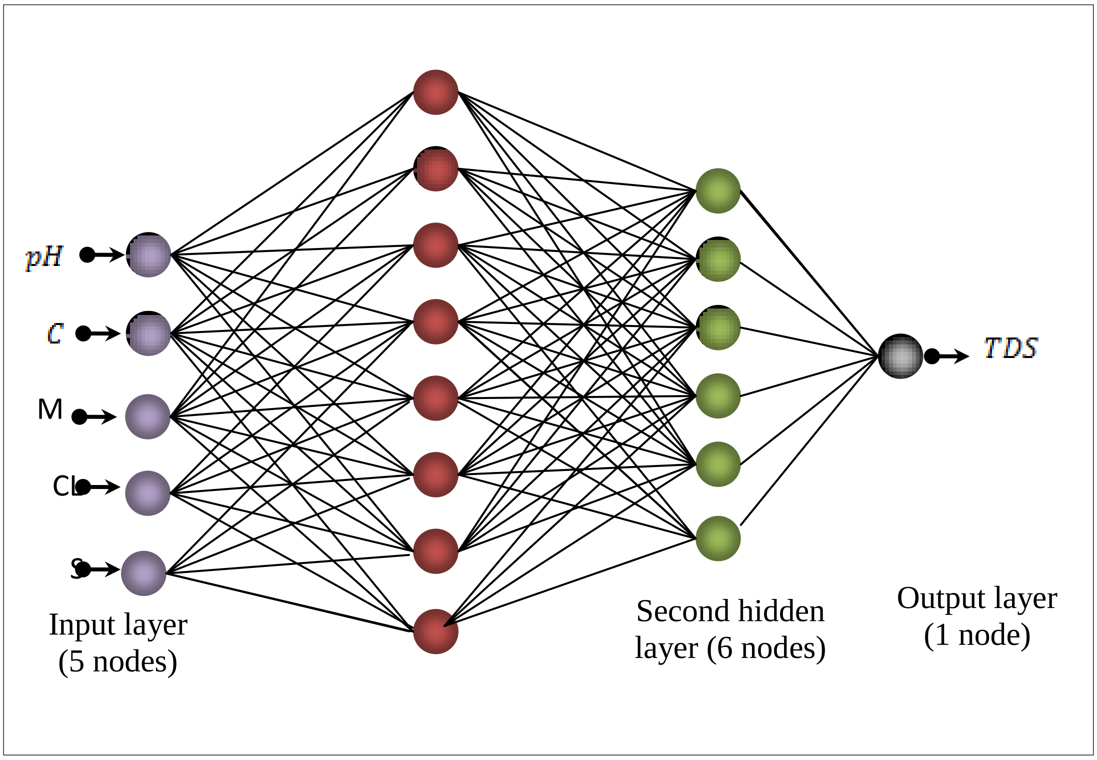


Figure 1 Configuration of Neural Network (3-7-5-2)

2.2 Normalizing Input and Output Data Set

The input and output data sets should be normalized before they are applied to the neural network so as to limit the input and output values within a specified range. This is due to the large difference in the values of the data provided to the neural network. Besides, the activation function used in the back propagation neural network is a hyperbolic tangent function, the lower and upper limits of this function are -1 and +1 respectively.

In this work, the used function for normalization is [10]:

$$\mathrm{pn_i} = 2[\mathrm{p_i} - \mathrm{p_{min}}/\mathrm{p_{max}} - \mathrm{p_{min}}] - 1 \qquad (1)$$

Where: $\mathrm{p_i}$is the value of *i*-th variable, $\mathrm{p_{min}}$is the minimum value of $\mathrm{p_i}$and $\mathrm{p_{max}}$is the maximum value of $\mathrm{p_i}$.

2.2 Optimization Technique and Error Estimates

Neural network functions depend non-linearly on their weights and so the minimization of the corresponding error function requires the use of iterative non-linear optimization algorithms. These algorithms make use of the derivatives of the error function with respect to the weights of the network. After completing the training process, the model is tested using another batch of data which has not been used in the training set.

The following statistical parameters of significance are calculated, for the present work, at the end of the training and testing calculations :

1. *Mean square error* (MSE): is a statistical measure of the differences between the values of the outputs in the training set and the output values the network is predicting. The goal is to minimize the value of MSE.
2. *Correlation coefficient* (*R*): is a measure of how the actual and predicted values correlate to each other. The goal is to maximize the value of *R*.The correlation coefficient function can be described as following [11]:

   $$R = an(tn'/Q - 1)/sta * stt \qquad (1)$$

   Where:an and tn' are the normalized outputs and transpose matrix of normalized target data respectively,sta and stt are the standard deviation of the output and target data respectively.

   Q is the number of the data in target vector.

2.3 Number of Hidden Layers and Number of Nodes in Hidden Layer

The choice of the number of hidden layers, number of nodes in the hidden layer and the activation function depends on the network application.

It is usually to start with a relatively small number of hidden units and increase it until we are satisfied with the approximation quality of the network. Unfortunately, the network needs to be fully retrained after each modification of its structure. The number of nodes in a hidden layer(s) drastically affects the outcome of the network training [12].

Therefore, trial-and-error approach is carried out to choose an adequate number of hidden layers and number of nodes in each hidden layer. The number of nodes in the hidden layer is selected according to the following rules:

1. The maximum error of the output network parameters should be as small as possible for both training patterns and testing patterns.
2. The correlation coefficient should be as high as possible especially. It is a measure of how well the variation in the output is explained by the targets. If this number is equal to (1), then there is perfect correlation between targets and outputs.

In this study the network is tested with one and two hidden layer configurations with an increasing number of nodes in each hidden layer(s). Different training function types and activation functions are investigated.

In this work, all training algorithms available in *MATLAB* (R2008a) are examined in this investigation. These algorithms are:

1. Conjugate gradient (traincgf, traincgp, traincgb, trainscg).
2. Quasi-Newton (trainbfg, trainoss).
3. Levenberg-Marquardt (trainlm, trainbr).
4. Gradient Descent (traingd, traingdm).
5. Variable Learning Rate (traingdx).
6. Resilient Back propagation (trainrp).

These functions were used for one hidden layer investigation and two hidden layers investigation [13].

The program of this work can be computerized in a three steps as following:

1. The first step is the "*Neural Network"* program that is coded in *MATLAB* (R2008a) language realizes the training and generalization processes of the back propagation network. The structure of this program is shown in Fig. 2. The main variables are stored using the cell arrays. The cell arrays in *MATLAB* are multidimensional arrays whose elements are copies of other arrays, and then the neural network description is extracted and saved in a separate file. The results of this step are suitable values of the weight and the biases.

2. The second step is the "*MATLAB Function*" program that is coded in *MATLAB* (R2008a) also. This function uses the network parameters extracted in step 1 to put the selected model in its operating mode.

3. The third step is the "*Graphical User Interface*" program that is coded using *MATLAB* (R2008a). This program uses the function extracted in step 2 to put this function in its operating mode as shown in Fig(2).

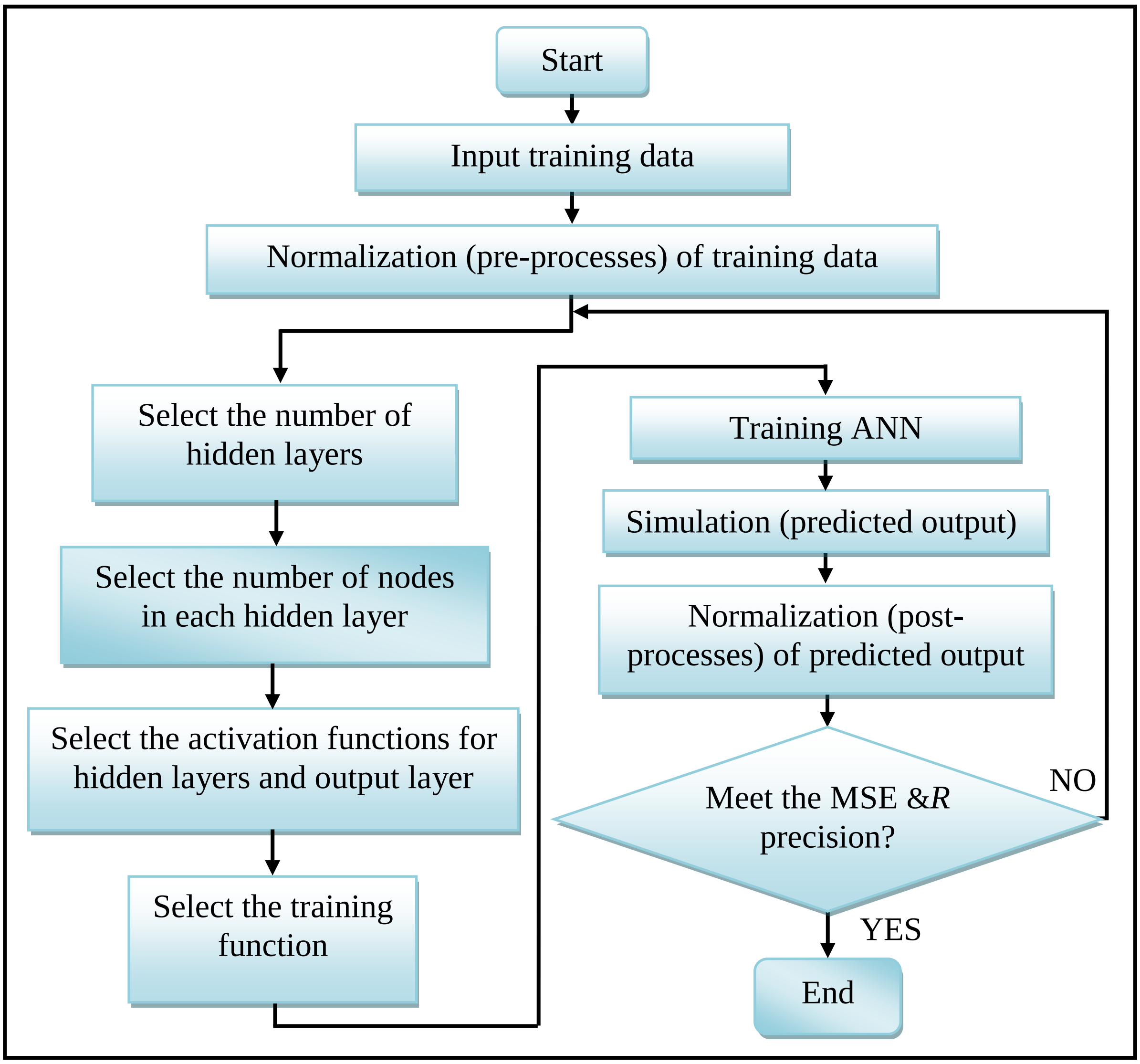

Start
Input training data
Normalization (pre-processes) of training data
Select the number of hidden layers
Select the number of nodes in each hidden layer
Select the activation functions for hidden layers and output layer
Select the training function
Training ANN
Simulation (predicted output)
Normalization (post-processes) of predicted output
Meet the MSE &*R* precision?
NO
YES
End

Fig. 2: The Structure of the Neural Network Program

Table (1): Input and Output Parameters

Item	*Parameters*	*Range of Parameters*		*Units*
		From	*To*	
Input Parameters	pH	6.9	8.48	-
	Calcium Concentration (ppm)	41.88	700	ppm
	Magnesium Concentration (ppm)	21.7	661	ppm
	Chloride Concentration (ppm)	30	1940	ppm
	Sulfate Concentration (ppm)	74	2500	ppm
Output Parameter	Total Dissolved Concentration (TDS) (ppm)	500	6189	ppm

3. Results and Discussions:-

3.1 One Hidden Layer Network:

One hidden layer networks are investigated with different training and activation functions for the hidden and output layer. Different numbers of nodes in each hidden layer from (6 to 12) nodes are used. The performance and regression of these topologies of network for both training and testing are shown in Tables (2,3,4,5,6 and 7).

Table (2): MSE and R for Conjugate Gradient training functions

Activation Function : {tansig, purelin}	*Training Function*	*Node No.*	*Training data (80%)*		*Testing data (20%)*	
			MSE	*R1*	*MSE*	*R1*
	traincgf	6	***0.0155***	***0.925***	***0.015***	***0.9036***
		8	***0.0153***	***0.926***	***0.0159***	***0.8961***
		10	***0.0078***	***0.963***	***0.01***	***0.935***
		12	***0.0078***	***0.963***	***0.01***	***0.935***
	traincgp	6	***0.015***	***0.9274***	***0.0147***	***0.9033***
		8	***0.0103***	***0.9507***	***0.0113***	***0.9267***
		10	***0.006***	***0.9717***	***0.0049***	***0.9672***
		12	***0.0069***	***0.9674***	***0.0091***	***0.943***
	traincgb	6	***0.0131***	***0.9371***	***0.0174***	***0.8874***
		8	***0.0119***	***0.9431***	***0.0106***	***0.9345***
		10	***0.088***	***0.9581***	***0.0086***	***0.9456***
		12	***0.0054***	***0.9745***	***0.0053***	***0.9665***
	trainscg	6	***0.0112***	***0.9461***	***0.0167***	***0.8865***
		8	***0.01***	***0.9523***	***0.012***	***0.9213***
		10	***0.0083***	***0.9605***	***0.0049***	***0.9688***
		12	***0.0059***	***0.9727***	***0.0082***	***0.9546***

Table (3): MSE and R for Quasi-Newton training functions

Activation Function : {tansig, purelin}	*Training Function*	*Node No.*	*Training data (80%)*		*Testing data (20%)*	
			MSE	*R*	*MSE*	*R*
	trainbfg	6	***0.0132***	***0.9366***	***0.0139***	***0.9113***
		8	***0.009***	***0.9569***	***0.0111***	***0.9261***
		10	***0.0058***	***0.9724***	***0.0058***	***0.9534***
		12	***0.0072***	***0.9657***	***0.0075***	***0.9511***
	trainoss	6	***0.0142***	***0.9314***	***0.0152***	***0.9027***
		8	***0.011***	***0.9471***	***0.0152***	***0.8954***
		10	***0.0099***	***0.9534***	***0.0148***	***0.9105***
		12	***0.0069***	***0.968***	***0.0098***	***0.9447***

Table (4): MSE and R for Gradient Descent training function

Activation Function : {tansig, purelin}	*Training Function*	*Node No.*	*Training data (80%)*		*Testing data (20%)*	
			MSE	*R*	*MSE*	*R*
	traingd	6	***0.04833***	***0.74***	***0.035***	***0.7376***
		8	***0.0388***	***0.7983***	***0.0277***	***0.8317***
		10	***0.0377***	***0.8042***	***0.0219***	***0.8502***
		12	***0.0377***	***0.8042***	***0.0219***	***0.8502***
	traingdm	6	***0.0369***	***0.8095***	***0.0228***	***0.8471***
		8	***0.0438***	***0.7682***	***0.0291***	***0.8004***
		10	***0.03398***	***0.8261***	***0.0197***	***0.8631***
		12	***0.0358***	***0.815***	***0.0202***	***0.865***

Table (5): MSE and R for Levenberg-Marquardt training function

Activation Function : {tansig, purelin}	*Training Function*	*Node No.*	*Training data (80%)*		*Testing data (20%)*	
			MSE	*R*	*MSE*	*R*
	Trainlm	6	***0.0171***	***0.9164***	***0.0148***	***0.9017***
		8	***0.0086***	***0.9591***	***0.0128***	***0.9175***
		10	***0.006***	***0.9721***	***0.0064***	***0.9652***
		12	***0.0055***	***0.9741***	***0.0043***	***0.9731***

Table (6): MSE and R for Variable Learning Rate training function

Activation Function : {tansig, purelin}	*Training Function*	*Node No.*	*Training data (80%)*		*Testing data (20%)*	
			MSE	*R*	*MSE*	*R*
	Traingdx	6	***0.0176***	***0.9144***	***0.0179***	***0.8826***
		8	***0.0172***	***0.9161***	***0.017***	***0.8888***
		10	***0.0186***	***0.9091***	***0.014***	***0.9066***
		12	***0.0149***	***0.9282***	***0.0177***	***0.8844***

Table (7): MSE and R for Resilient Backpropagation training function

Activation Function : {tansig, purelin}	***Training Function***	***Node No.***	***Training data (80%)***		***Testing data (20%)***	
			MSE	***R***	***MSE***	***R***
	Trainrp	6	***0.0186***	***0.909***	***0.0207***	***0.8568***
		8	***0.0118***	***0.9433***	***0.0147***	***0.9054***
		10	***0.0102***	***0.9515***	***0.011***	***0.9296***
		12	***0.0081***	***0.9613***	***0.009***	***0.9413***

It can be seen that From Tables (2,3,4,5,6 and 7), the networks with (12) nodes in the hidden layer and activation function as hyperbolic tangent (*tansig*) and (*purelin*) function for hidden and output layers respectively gives best performance and correlation coefficient for networks with training function (traincgf, traincgb, trainscg, trainoss, traingd, trainlm and trainrp)while networks with (10) nodes in the hidden layer and training function (traincgp, trainbfg, traingdm and traingdx) gives best results than other. The best performance network for predicting TDS concentration in water is the one with (12) nodes in hidden layer with training function (trainlm).

3.2 Two Hidden Layers Investigation

Artificial Neural Networks(ANNs) with Two hidden layers and different training and activation functions for each layer are investigated. Different nodes numbers in each hidden layer from (6 to 12) nodes in the first hidden layer and (5-10) in the second one are choosing. The performance and regression of these topologies for both training and testing are shown in Tables (8, 9, 10, 11, 12 and 13).

Table (8): MSE and R for Conjugate Gradient training functions

Activation Function : {tansig, tansig, purelin}	*Training Function*	*Node No.*	*Training data (80%)*		*Testing data (20%)*	
			MSE	*R*	*MSE*	*R*
	traincgf	*6-5*	*0.0052*	*0.9758*	*0.0111*	*0.9291*
		8-6	*0.0023*	*0.9896*	*0.0081*	*0.9516*
		10-8	*8.831e-4*	*0.9959*	*0.0026*	*0.9828*
		12-10	*4.422e-4*	*0.9979*	*6.477e-4*	*0.9963*
	traincgp	*6-5*	*0.0088*	*0.9587*	*0.0119*	*0.9295*
		8-6	*0.0128*	*0.9389*	*0.0148*	*0.9119*
		10-8	*0.0021*	*0.9903*	*0.0041*	*0.9734*
		12-10	*0.0019*	*0.9911*	*0.0024*	*0.9848*
	traincgb	*6-5*	*0.0064*	*0.97*	*0.0111*	*0.931*
		8-6	*0.0015*	*0.9927*	*0.0034*	*0.9927*
		10-8	*5.4205e-4*	*0.9975*	*0.0014*	*0.9912*
		12-10	*2.3210e-4*	*0.9986*	*2.3210e-4*	*0.9986*
	trainscg	*6-5*	*0.0027*	*0.987*	*0.0071*	*0.9528*
		8-6	*0.0019*	*0.9911*	*0.0043*	*0.9717*
		10-8	*3.923e-4*	*0.9982*	*8.54e-4*	*0.9944*
		12-10	*3.052e-5*	*0.9999*	*1.485e-5*	*0.9999*

Table (9): MSE and R for Quasi-Newton training functions

Activation Function : {tansig, tansig, purelin}	*Training Function*	*Node No.*	*Training data (80%)*		*Testing data (20%)*	
			MSE	*R*	*MSE*	*R*
	trainbfg	*6-5*	*0.0022*	*0.9896*	*0.0064*	*0.9599*
		8-6	*0.001*	*0.9952*	*0.002*	*0.9872*
		10-8	*2.013e-4*	*0.9991*	*3.255e-4*	*0.9979*
		12-10	*1.588e-4*	*0.9993*	*1.939e-4*	*0.9987*
	trainoss	*6-5*	*0.0039*	*0.9817*	*0.0071*	*0.9561*
		8-6	*0.0047*	*0.9777*	*0.0052*	*0.9667*
		10-8	*8.796e-4*	*0.9959*	*0.0021*	*0.9863*
		12-10	*8.018e-4*	*0.9962*	*9.428e-4*	*0.9939*

Table (10): MSE and R for Levenberg-Marquardt training function

Activation Function : {tansig, tansig, purelin}	Training Function	Node No.	Training data (80%)		Testing data (20%)	
			MSE	R	MSE	R
	trainlm	6-5	0.003	0.9859	0.0027	0.9839
		8-6	3.132e-4	0.99895	5.042e-4	0.9967
		10-8	2.49e-22	1	3.807e-22	1
		12-10	3.044e-24	1	4.245e-24	1

Table (11): MSE and R for Gradient Descent training function

Activation Function : {tansig, tansig, purelin}	Training Function	Node No.	Training data (80%)		Testing data (20%)	
			MSE	R	MSE	R
	Traingd	6-5	0.0463	0.7529	0.028	0.8062
		8-6	0.0385	0.8002	0.0279	0.8031
		10-8	0.0384	0.8214	0.0175	0.8837
		12-10	0.0287	0.8555	0.025	0.8185
	Traingdm	6-5	0.038	0.798	0.026	0.8126
		8-6	0.0186	0.9096	0.0197	0.8653
		10-8	0.0348	0.8214	0.0175	0.8837
		12-10	0.0379	0.8032	0.025	0.8213

Table (12): MSE and R for Variable Learning Rate training function

Activation Function : {tansig, tansig, purelin}	Training Function	Node No.	Training data (80%)		Testing data (20%)	
			MSE	R	MSE	R
	traingdx	6-5	0.0138	0.9338	0.019	0.877
		8-6	0.0147	0.9289	0.0127	0.9157
		10-8	0.0132	0.9396	0.016	0.8991
		12-10	0.0089	0.9577	0.0103	0.9366

Table (13): MSE and R for Resilient Backpropagation training function

Activation Function : {tansig, tansig, purelin}	Training Function	Node No.	Training data (80%)		Testing data (20%)	
			MSE	R	MSE	R
	trainrp	6-5	0.0065	0.9692	0.0107	0.9327
		8-6	0.0061	0.9714	0.0091	0.9458
		10-8	0.0047	0.9776	0.0057	0.9633
		12-10	0.0012	0.9944	0.0027	0.9853

It can be seen that tables from (8) to (13) show that the response of network with different training functions changed with variation of nodes in the first and the second hidden layers. The training function selected here is conjugate gradient back propagation type (*TRAINSCG*) which indicated high regression and best performance with lowest mean square error. Other functions are investigated and compared in the present work. The transfer function (trainlm) with node (14-10) and (15-12) have high MSE values but its performance suffer from overfitting data, therefore it is not good function compared with other functions.

The selected training function (TRAINSCG) in this study must be examined with another different activation function to complete this investigation as showing in Table (14) and Fig. (14), which are shown that *(tansig, tansig, purelin)* activation function arrangement gives the best performance and regressions for both training and testing phases.

Table (14): MSE & Regression with Different Arrangements of Activation Functions Each of Two hidden layer (12-10) Network with TRAINSCG Training Function.

TRAINSCG Training Function	***Arrangements of Activation Functions***				
	(tansig, purelin, purelin)	***(tansig, tansig, purelin)***	***(tansig, tansig, tansig)***	***(purelin, tansig, tansig)***	***(tansig, purelin, tansig)***
MSE (train)	0.0066	***3.052e-5***	1.7649e-4	0.0045	0.0052
MSE (test)	0.0074	***1.485e-5***	1.5775e-4	0.0132	0.0081
R (train)	0.9688	0.9999	0.9992	0.9788	0.9751
R (test)	0.9540	0.9999	0.9990	0.9172	0.9546

The analysis o these results lead to the fact that the training function ((*trainscg*) with activation functions (*tansig*) and (purelin) for the two hidden layers and output layer respectively between all other different arrangements of neural networks gives the best MSE and correlation coefficients for both training and testing than other. Therefore, this network can be selected as a suggested network for this study.

Figures (3, 4 and 5) show the regression analysis between the output of neural network and the corresponding target for training, testing and overall data respectively for Total Dissolved Solids Concentration in water. Outputs are plotted versus the targets as open circles. The solid line indicates the best linear fit and the broken line indicates the perfect fit (output equals target). The high regression analysis (R=0.999) is obtained in these figures which obtained in network with (12 and 10) nodes in the first and second hidden layer respectively using (trainscg) as transfer function with activation function (tansig, tansig and purelin).

Figures (6) and (7) show the behavior of ANN to predict Total Dissolved Solids Concentration (TDS) for training and testing data set respectively. It can be seen that the actual and predicted values are close to each other, can concluded that ANN model has high accuracy levels of prediction.

Figure (8) shows training and testing performance of the best two hidden layer network. The performance of a trained network can be measured to some extent by the errors on the training and testing sets, but it is often useful to investigate the network response in more detail. One option is toper form a regression analysis between the network response and the corresponding targets. Figure (9) represents a comparison between the best one and two hidden layers performance which gives good error for the network with two idden layer usin training fuction as trainscg.

Figure (10) studied the variation of Mean Square Errors (MSE) with different activation functions each of (5-12-10-1) network with *trainscg* trianing function, it can be seen that the function (tansig - tansig – purelin) indicated high performance than other with MSE of (3.052e-5) to the best network.

Figure (11) represent Graphical User Interface (GUI)of the Neural Network Program which act as an artificial neural network model to predict Total Dissolved Solids Concentration (TDS) in (ppm) at formation behavior of another data to another input water quality values inside and outside training data range respectively.

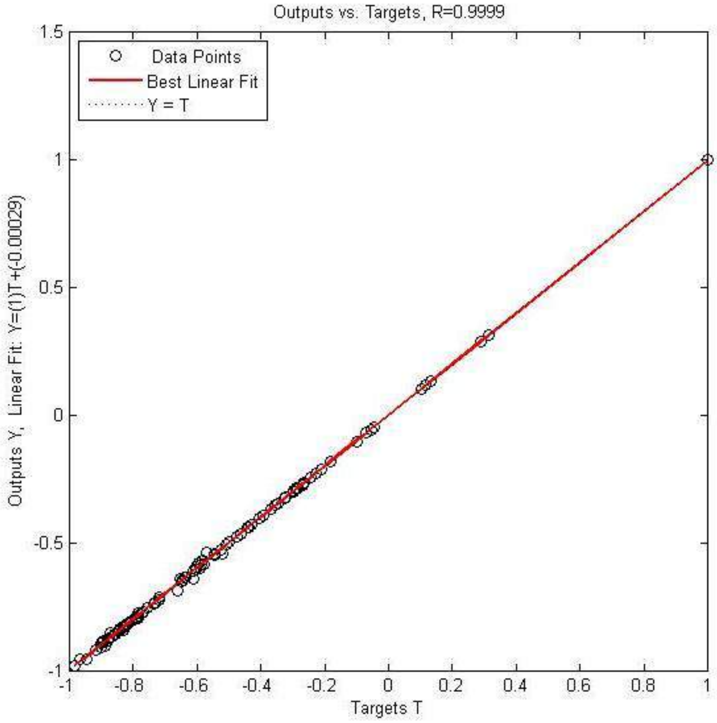


Fig.(3) : Training TDS Regression of the Two Hidden Layer (trainscg) Network

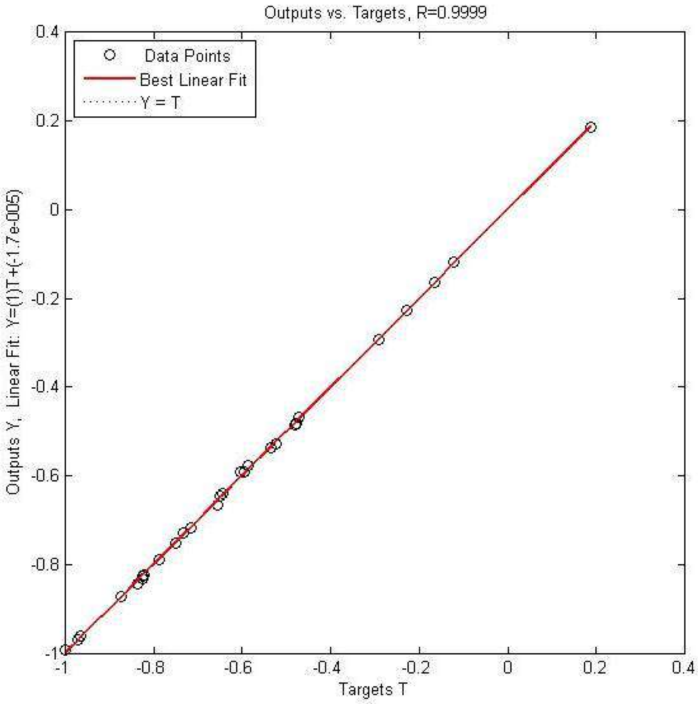


Fig. (4) : Testing TDS Regression of the Two Hidden Layer (trainscg) Network

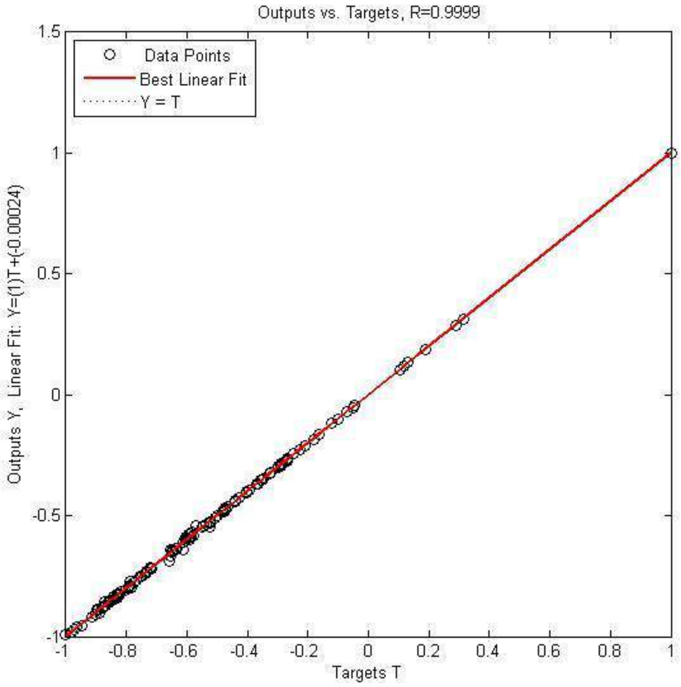


Fig. (5) : Overall TDS Regression of the Two Hidden Layer (trainscg) Network

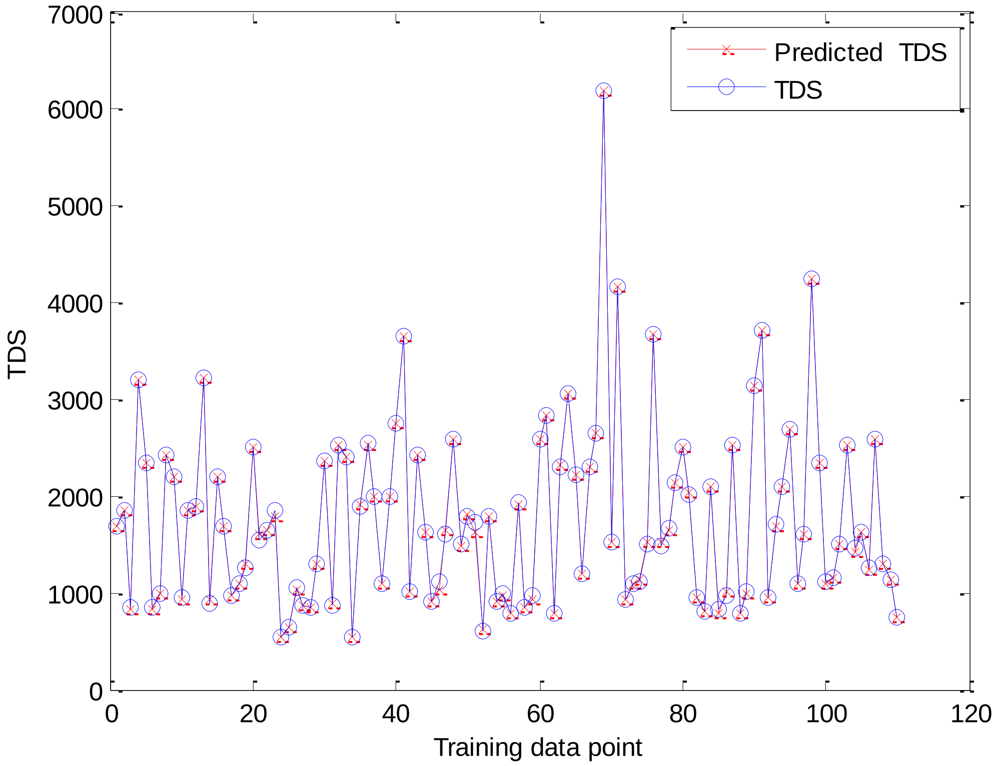


Fig. (6): Training Behavior of Predicted TDS.

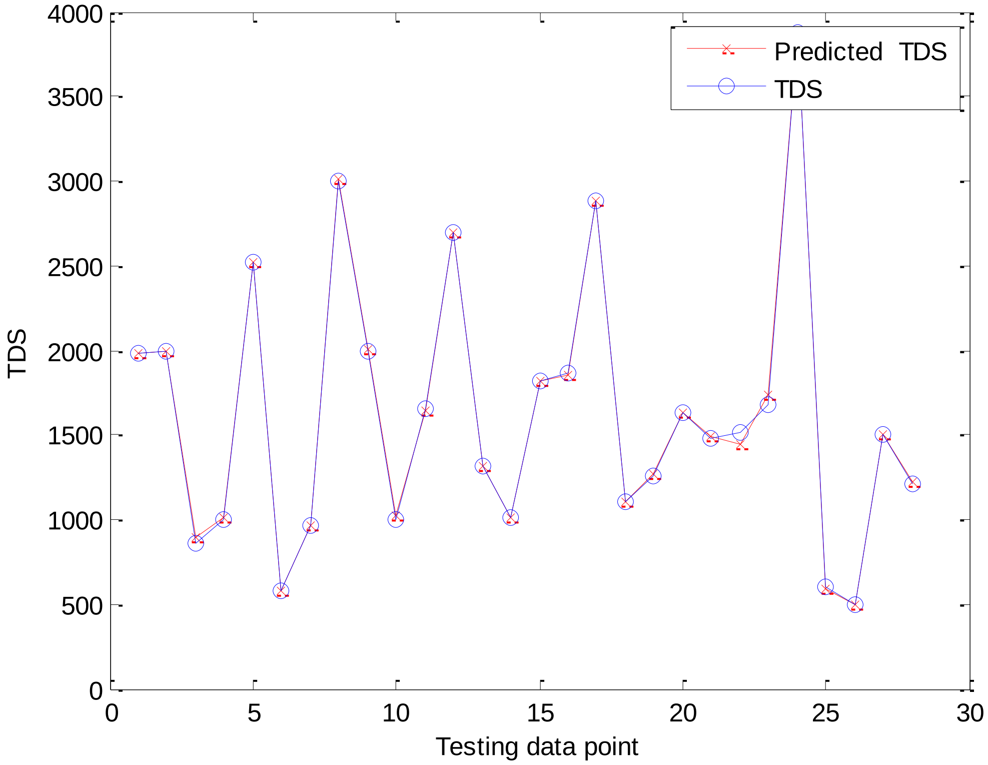


Fig (7): Testing Behavior of Predicted TDS.

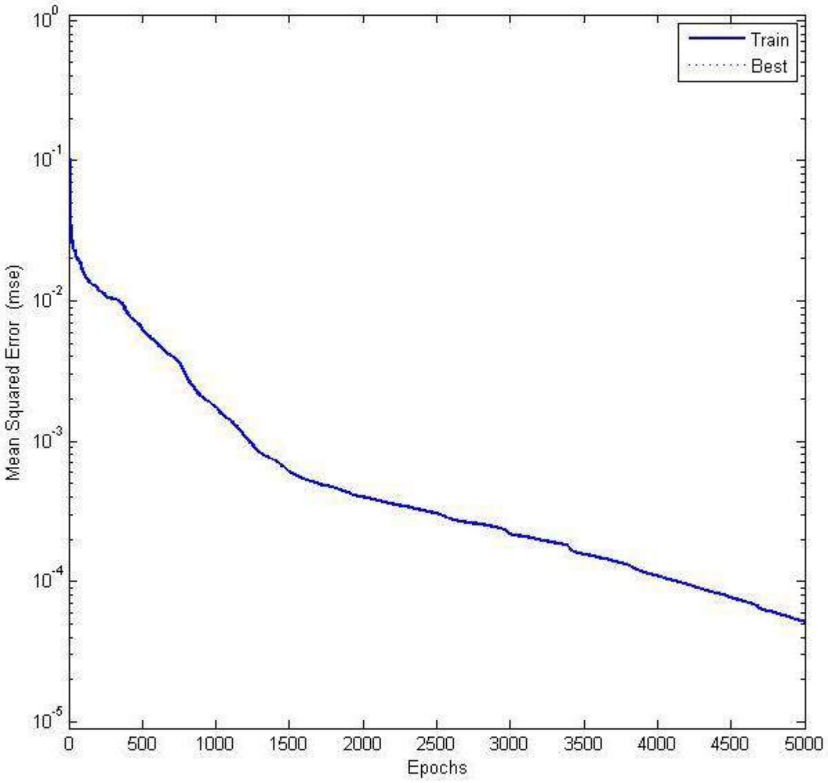


Fig. (8) : Training MSE vs. Epochs of the Best Two Hidden Layer Network (trainscg)

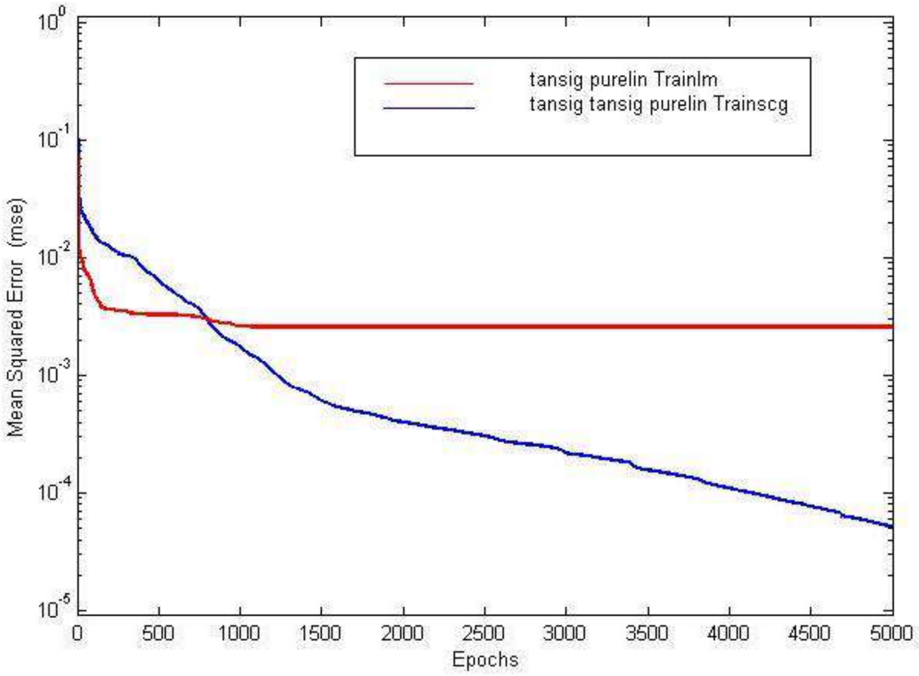


Fig. (9) : Comparison between the Best One and Two Hidden Layers Networks Performance

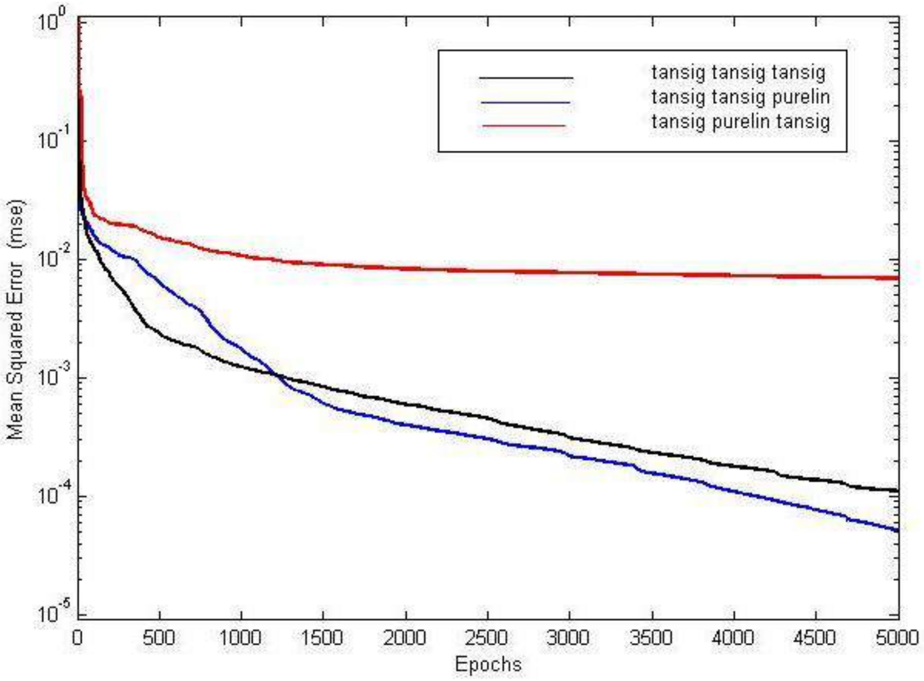
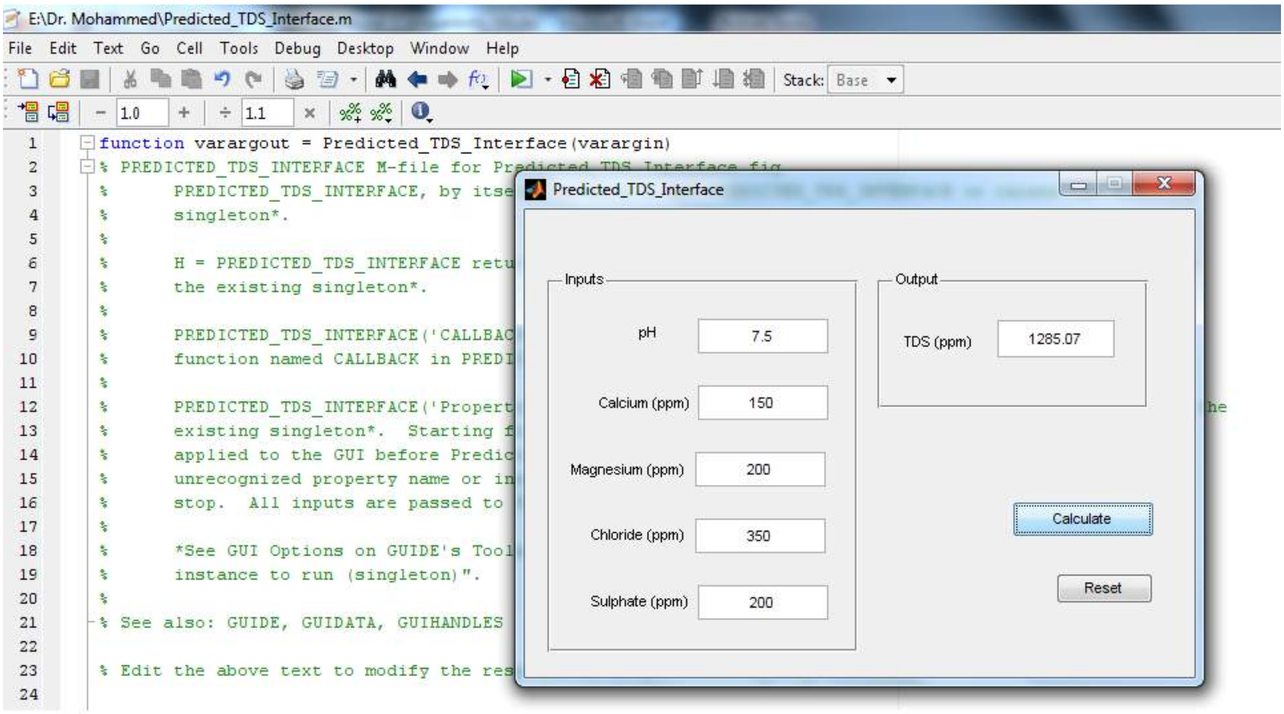


Fig. (10) : Comparison between Activation Function for the (trainscg) in Two Hidden Layer Network


E:\Dr. Mohammed\Predicted_TDS_Interface.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
1.0
1.1
Predicted_TDS_Interface
Calculate
Reset

```
function varargout = Predicted_TDS_Interface(varargin)
% PREDICTED_TDS_INTERFACE M-file for Predicted_TDS_Interface.fig
%      PREDICTED_TDS_INTERFACE, by itse
%      singleton*.
%
%      H = PREDICTED_TDS_INTERFACE retu
%      the existing singleton*.
%
%      PREDICTED_TDS_INTERFACE('CALLBAC
%      function named CALLBACK in PREDI
%
%      PREDICTED_TDS_INTERFACE('Propert
%      existing singleton*.  Starting f
%      applied to the GUI before Predic
%      unrecognized property name or in
%      stop.  All inputs are passed to
%
%      *See GUI Options on GUIDE's Tool
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the res
```

Inputs

pH 7.5

Calcium (ppm) 150

Magnesium (ppm) 200

Chloride (ppm) 350

Sulphate (ppm) 200

Output

TDS (ppm) 1285.07

Figure (11) : Graphical User Interface (GUI)of the Neural Network Program.

4- Conclusion:

This paper, one and two hidden layer feedforword back-propagation artificial Neural network models were applied for prediction of total dissolved solids in marsh water . a (140) sample data are choose in the ANNs with five input parameters (pH, Calcium, magnesium, chloride and sulphate concentrations) in one and two hidden layers with different network conditions. Results show best performance and regression analysis reached (0.9999) for both training and testing date of this research with two hidden layer (trainscg as training function) with (5-12-10-1) neurons in their layers. Also, the results indicated that ANNs model provided a reliable and simple tool for the prediction of TDS in marsh water. It was concluded that this research can be considered as contribution to an on ongoing effort to develop artificial neural network model to solve water and waste water treatment.

Reference:

1- Mohammed D. (2011) "Assessment Of Toxic Materials In Sunaf Marsh (Alkhawisa Village)", J.Thi-Qar Sci vol. 3 No.1.

2- Kanani S., Asadollahfardi G. and Ghanbari A. (2008) " Application of Artificial Neural Network to Predict Total Dissolved Solid in Achechay River Basin", World Applied Sciences Journal 4 (5): 646-654.

3- Zhang, Q., and Stanley, S.J. (1997). "Forecasting raw-water quality parameters for the North Saskatchewan River by neural network modeling. Water Research", **31**(9): 2340–2350.

4- Eman Abd Mohammed (2010) " Water Quality Monitoring Of Al-Hawizeh Marsh", Al-Qadisiya Journal For Engineering Sciences Vol. 3 No. 3.

5- Riyadh Z. and Shaimaa U. (2009) "Effect Of Al Huwayza Marsh Boundary Configuration On The Velocity Patterns And Water Quality Distribution", Vol. 15 No.1

6- Mohammed D. (2011) " Assessment Of Toxic Materials In Sunaf Marsh (Alkhawisa Village)", J.Thi-Qar Sci vol. 3 No.1.

7- Mohammed D., Kaissar I. and Haider A. (2012) "Effect of Chemical and Physical Parameters of Almasab Alaam Drainage on Properties of Marsh water in South of Iraq" accepted in Al-Qadisiya Journal For Engineering Sciences

8- Martin T. (1996) " Neural Network Design", PWS Publishing company.

9- ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a. Artificial Neural Networks in Hydrology: Preliminary Concepts. Part I, J. Hydrologic Engrg., ASCE, 5 (2): 115-123.

10- Howard Demuth (2008) " Neural Network Toolbox 5 User's Guide", The Math Works, Inc.

11- Hornik, K.M., M. Stinchcombe and H. White, 1989. Multilayer Feedforward Networks are Universal Approximators. Neural Networks, Vol: 2.

12- Jain, A.K., Mao, J.C., and Mohiuddin, K.M. (1996). "Artificial neural networks: a tutorial. Computer", **29**(3): 31–44.

13- Rafiq M.Y., Bugmann G. and Easterbrook D.J (2001) " Neural Network Design for Engineering Application" Journal of computers and Structures", Vol. 79 No. (17), 1451-1552.