Estimating Reference Evapo- transpiration in Mosul (Iraq) Using Cascade Neural Networks

Fatin Mahmoud Shehab
Engineering Technical College, Foundation of Technical Education /Mosul
Email: fatinalobaid@yahoo.com
Raid Rafi Omar
Engineering Technical College, Foundation of Technical Education /Mosul
Radhwan Yousif Sedik
Engineering Technical College, Foundation of Technical Education /Mosul

Received on: 22/9/2013 & Accepted on: 6/4/2014

ABSTRACT

Recently artificial neural network (ANN) has been applied for estimating reference evapo-transpiration (ET_o).In this study a mathematical model was built by application the cascade forward network technique (CCANN) to estimate the daily reference evapo-transpiration in the city of Mosul, north of Iraq .The input parameters for the CCANN were the: temperature, solar radiation, wind speed at 2m height, and relative humidity. A check for the accuracy of the performance of the network was made using values of reference evapo-transpiration obtained from pan evaporation method. The results revealed linear correlation between the network output and the data of the measured pan evapo-transpiration with correlation coefficient of (0.9679). This indicates the possibility of use of CCANN to determine the daily reference evapo- transpiration. The results also show that the CCANN model performs better more accurate compared to other models.

Keywords: evapo-transpiration, cascade –forward, neural network.

تخمين التبخر- نتح المرجعي اليومي لمنطقة الموصل (العراق) باستخدام الشبكات العصبية نوع Cascade

الخلاصة

لقد شاع في الآونة الأخيرة استخدام أسلوب الشبكات العصبية الصناعية في تخمين مقدار التبخر-نتح. في هذا البحث تم بناء انمودج رياضياتي باعتماد تطبيق تقنيات الشبكات العصبية الصناعية نوع Cascade Forward لغرض تخمين مقدار التبخر-نتح المرجعي اليومي لمنطقة الموصل (شمال العراق). البيانات التي استخدمت كمدخلات للشبكة العصبية هي درجة الحرارة , الإشعاع الشمسي , سرعة الرياح على ارتفاع 2 متر والرطوبة النسبية. ولغرض التأكد من صحة أداء الشبكة استخدمت بيانات التبخرالانائي اليومي Class A Pan للمنطقة ولنفس الفترة كمخرجات مأمولة أو كهدف .أظهرت النتائج توافق خطي بين مخرجات الشبكة وبيانات التبخر الانائي المقاسة وبمعامل ارتباط (0.9679) وهذا يبين إمكانية استخدام CCANN لحساب التبخر-نتح المرجعي اليومي . وبمقارنة هذه النتائج بالنتائج المستحصلة من البحوث السابقة والتي استخدمت فيها نماذج أخرى من الشبكات العصبية الصناعية نجد أن هذه الطريقة أعطت نتائج أكثر دقة وكفاءة .

INTRODUCTION:

Reference evapo-transpiration is a complex and nonlinear process which depends on several interacting weather climatological factors, such as temperature, humidity, wind speed, bright sunshine hours, etc. Artificial neural networks (ANNs) are a mathematical construct whose architecture is essentially analogous to the human brain. Basically, the highly interconnected processing units are arranged in layers similar to the arrangement of neurons in the brain [1].

ANN is a flexible mathematical structure, which is capable of identifying complex nonlinear relationships between input and output data sets. The ANN models have been found useful and efficient particularly in problems for which the characteristics of the processes are difficult to represent by conventional mathematical equations. ANN has found successful applications in the areas of a science, engineering, industry, business, economics, and agricultures [2].

Review of Literature:

The ANNs provide better modeling flexibility than the other statistical approaches with its successive adaptive features of error propagation, where each meteorological variable takes its share proportionally.

Al-Aani (2007) [3] tested an artificial neural network (ANN) for estimating the reference evapo-transpiration (ET_o) using the (NN-tool box) which is one of the MATLAB tools using various daily climatic data. A multi-layers (ANN) architecture of error back- propagation algorithm was built. It was found that the (ANN) architecture of (4-5-1) (4 neurons in inputs layer -5 neurons in hidden layer -one neuron for output layer).

Chauhan and Shrivastava (2008) [4] tried to find best alternative method to estimate ET_o when input climatic parameters are insufficient to apply standard FAO-56 Penman–Monteith equation. The ANNs, using varied input combinations of climatic variables have been trained using the back-propagation with variable learning rate training algorithm. ANNs were performed better than the climatic based equations in all performance indices. The analyses of results of ANN suggest that the ET_o can be estimated from maximum and minimum temperature using ANN approach [5].

The Radial Basis Function (RBF) network was applied in Trajkovic,s.(2009a) [5] for pan evaporation to evapo-transpiration conversions. The obtained RBF network, Christiansen, FAO-24 pan, and FAO-56 Penman-Monteith equations were verified in comparison to lysimeter measurements of grass evapo-transpiration. Based on summary statistics, the RBF network ranked first with the lowest RMSE value. The RBF network was additional tested using mean monthly data collected in Novi Sad, Serbia, and Kimberly, Idaho, U.S.A. The overall results recommended RBF network for pan evaporation to evapo-transpiration conversions[6].

Diamantopoulou and others (2011) **[7]** investigated the modeling of evapo-transpiration using the back propagation algorithm (BPANN) and the cascade correlation architecture (CCANN) with minimal meteorological data. The comparison was based on error statistical technique using P-M daily ET_o values as reference. It was found that taking into account only the mean, maximum and minimum air temperatures, the selected ANN models markedly improved the daily ET_o estimates and provided unbiased prediction and systematically better accuracy compared with the HGadj equation. The results also show that the CCANN model performed better than the BPANN model at all station.

The objective of this study was to test CCANN-forward model for estimating daily ET_o with varying daily climatic data and then compare the results obtained with values of reference evapo-transpiration obtain by pan evapo- transpiration method. Fu A comparisons with other models of ANN were also done depending on error statistical techniques.

Theoretical Background:

Architecture of standard Cascade-Forward

Cascade-Forward network is a layered network in which each layer only receives inputs from previous layers. It is a multilayer neural network with one layer of hidden units (the H units) as shown in Fig (1). The output units (the O units) and the hidden units also may have biases. The bias on a typical output unit O_k is denoted by b_{0k}; the bias on a typical hidden unit H_j is denoted e_{0j}. These bias terms act like weights on connections from units whose output is always 1. Only the direction of information flow for the feed forward phase of operation is shown. During the Cascade-forward phase of learning, signals are sent in the reverse direction. The algorithm in the next section is presented for one hidden layer, which is adequate for a large number of applications [8].

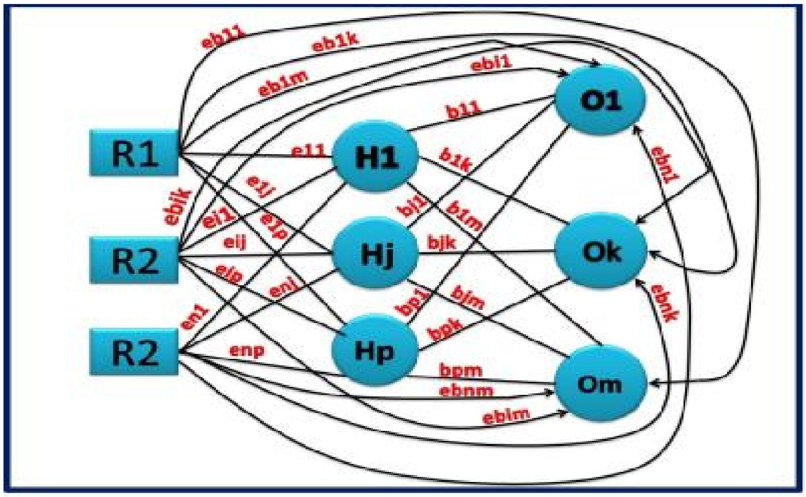


Figure (1). Cascade-Forward with one hidden layer

Results and Discussion :

The training Algorithm of Cascade-forward

As mentioned earlier, training a network by Cascade-forward involves three stages: the feed forward of the input training pattern of the associated error, and the adjustment of the weights.

During feed forward, each input unit (R_i) receives an input signal and broadcasts this signal to each of the hidden units H_1 …….. H_p through the (e) weights. Each hidden unit then computes its activation and results in its signal (h_j) to each output unit through the (b) weights. Each output unit (O_k) computes its activation (o_k)

to form the response of the net for the given input pattern. During training, each output unit compares its computed activation o_k with its target value d_k to determine the associated error for that pattern with that unit. Based on this error, the factor d_k *(k=1 ……. m)* is computed, d_k is used to distribute the error at output unit O_k back to all units in the previous layer (the hidden units that are connected to O_k). It is also used (later) to update the weights between the output and the hidden layer. In a similar manner, the factor d_j *(j=1 …….. p)* is computed for each hidden unit H_j. It is not necessary to propagate the error back to the input layer, but d_j is used to update the weights between the hidden layer and the input layer. When all the *d* factors have been determined, the weights for all layers are adjusted simultaneously. The adjustment to the weight b_{jk} (from hidden unit H_j to output unit O_k) and eb_{ik} (from input unit R_i to output unit O_k) is based on the factor d_k and the activation h_j of the hidden unit H_j. The adjustment to the weight e_{ij} (from input unit R_i to hidden unit H_j) is based on the factor d_j and the activation r_i of the input unit.

An epoch is one cycle through the entire set of training vectors. Typically, many epochs are required for training a Cascade-forward . The foregoing algorithm updates the weights after each training pattern is presented. A common variation is a updating 1batch, in which weight updates are accumulated over an entire epoch before being applied [8].

The general training algorithm for Cascade-forward is as follows:

Step 0. Initialize weights. (Set to small random values).

Step 1. While stopping, condition is false (output meet the goal), do Steps 2-9.

Step 2. For each training pair, do Steps 3-8.

Feed forward:

Step 3. Each input unit (*R_i, i = 1……n*) receives input signal r_i and broadcasts this signal to all units in the layer above (the hidden units).

Step 4. Each hidden unit (*H_j, j = 1………p*) sums its weighted input signals.

$$h\_ins_j = v_{0i} + \sum_{i=1}^{n} r_i v_{ij} \qquad \text{….. (1)}$$

applies its activation function to compute its output signal.

$$H_j = f(z\_ins_j) \qquad \text{….. (2)}$$

and sends this signal to all units in the layer above (output units).

Step 5. Each output unit (*O_k, k = 1 ……… m*) sums its weighted input signals.

$$o\_ins_k = b_{0k} + \sum_{j=1}^{p} hb_{j\ jk} + \sum_{i=1}^{n} x_i eb_{ik} \qquad \text{….. (3)}$$

and applies its activation function to compute its output signal.

$$o_k = f(o\_ins_k) \qquad \text{….. (4)}$$

of error:

Step 6. Each output unit (*O_k, k = 1……..m*) receives a target pattern corresponding to the input training pattern, computes its error information term.

$$d_k = (d_k - o_k)\, f'(o\_ins_k) \qquad \text{…(5)}$$

calculates its weight correction term (used to update b_{jk} later).

$$\Delta b_{ik} = ad_k h_j \qquad \text{…….. (6)}$$

Also, calculates another weight correction term (used to update eb_{ik} later).

$$\Delta eb_{ik} = ad_k r_i \qquad \text{…….. (7)}$$

calculates its bias correction term (used to update b_{0k} later).

$$\Delta b_{0k} = ad_k \qquad \text{…….. (8)}$$

and sends d_k to units in the layer below.

Step 7. Each hidden unit (H_j, $j = 1\ldots\ldots\ldots p$) sums its delta inputs (from units in the layer above).

$$d\_ins_j = b_{0k} + \sum_{k=1}^{m} d_k b_{jk} \qquad \text{…… (9)}$$

multiplies by the derivative of its activation function to calculate its error information term.

$$d_j = d\_ins_j \; f'(h\_ins_j) \qquad \text{……(10)}$$

calculates its weight correction term (used to update e_{ij} later).

$$\Delta e_{ij} = ad_j r_i \qquad \text{…….. (11)}$$

and calculates its bias correction term (used to update e_{0j} later).

$$\Delta e_{0j} = ad_j \qquad \text{…… (12)}$$

Update weights and biases:

Step 8. Each output unit (O_k, $k = 1\ldots\ldots m$) updates its bias and weights ($i = 0\ldots\ldots\ldots n$) , ($j = 0\ldots\ldots\ldots p$):

$$b_{jk}(new) = b_{jk}(old) + \Delta b_{jk} \qquad \text{……(13)}$$

$$eb_{ik}(new) = eb_{ik}(old) + \Delta eb_{ik} \qquad \text{…..(14)}$$

Each hidden unit (H_j, $j = 1\ldots\ldots\ldots p$) updates its bias and weights ($i = 0\ldots\ldots n$):

$$e_{ij}(new) = e_{ij}(old) + \Delta e_{ij} \qquad \text{……(15)}$$

Step 9. Test stopping condition. [8][9].

Neural Network and Models Evaluation

Different CCANN models where tested to determine the optimum number of hidden layers and number of nods in each layer .The architecture of the best CCANN model for forecasting the daily ET_o at Mosul station is composed of one input layer with four inputs variables, temperature (T_{max} 36.75_T_{min} 4.5)C°, solar radiation (Rs_{max} 13.3_Rs_{min} 0) hr, wind speed (U_{max} 3.38_ U_{min} 0.37)m/sec, and relative humidity (RH_{max} 97.5% _ RH_{min} 11.5%), one hidden layer with five nods and one output layer with one output variable, figure (2).

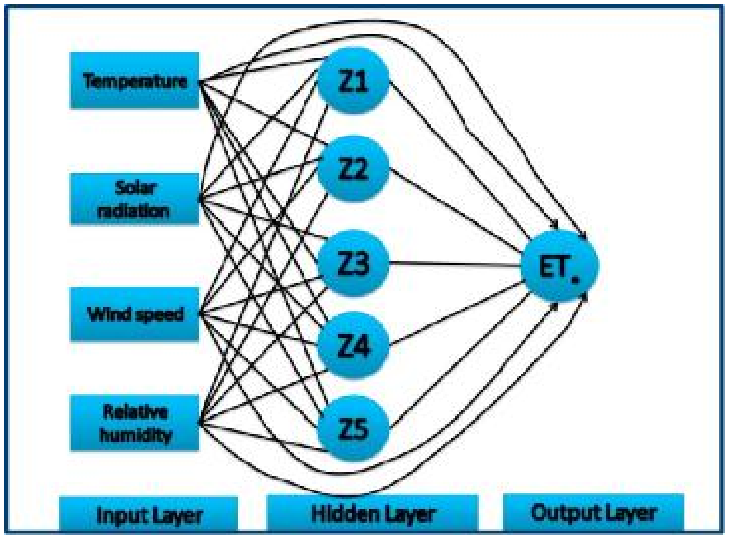


Figure (2) : The proposed CCANN to estimate ET_o

The data used in this study was taken from Al-Aani, Iftekher” (2007) [3] who used Back propagation algorithm for estimating the reference evapo-transpiration (ET_o)with Square error average (MSE) equal to 0.00105. Here the data was accessed to appropriate network architecture .The total aggregate data used were 366 case .This has been divided into two groups, the first data group is that of the train with total cases of 183, and the second is that of the test with total cases of 183 also .The proposed neural network CF consisted of 4 nodes at input layer, 5 nodes at the hidden layer, and one output at the output layer. The activation functions used were tansig type at the hidden layer and logsig at the output layer, Figure3.

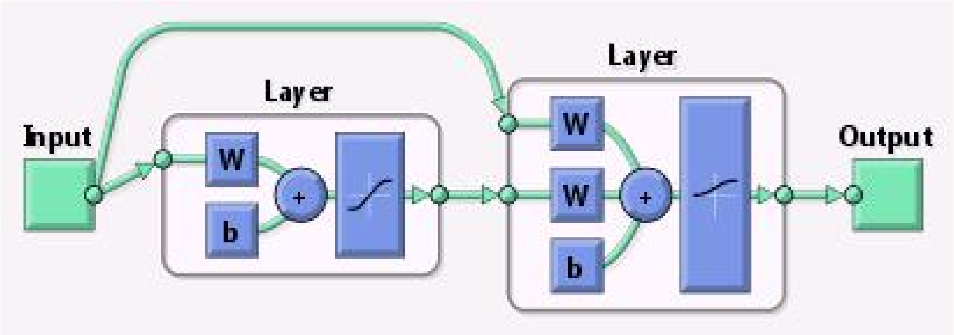


Figure (3): The proposed artificial neural network CF

The training of the data by the new propose neural network was achieved all of the training objective with a descending training curve, which reflected the training error that decreases gradually until reaching the lowest values of 0.0002 with 15 iterations, Figure (4).

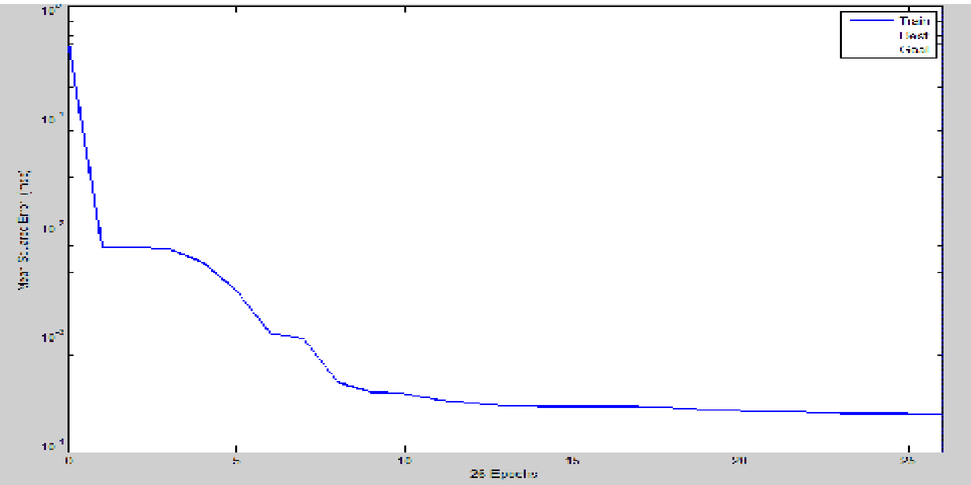


Figure (4): The neural network CF training curve

A statistical analysis was made by using the regression analysis to find the liner correlation between the propose networks output and the measured daily pan evaporation for the year 1980 ,Figure (5).The results revealed a liner correlation between them with a correlation coefficient of (0.9679),and this indicate the possibility of using this technique in this field.

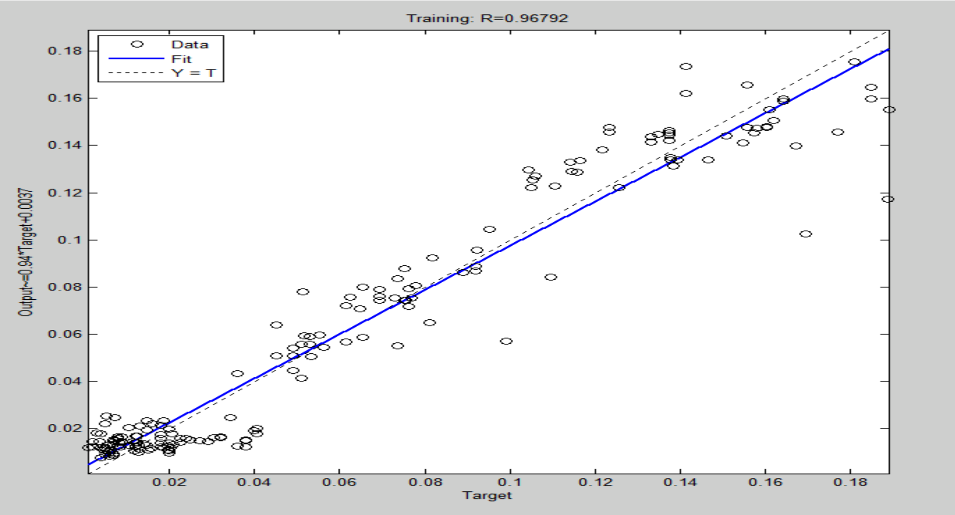
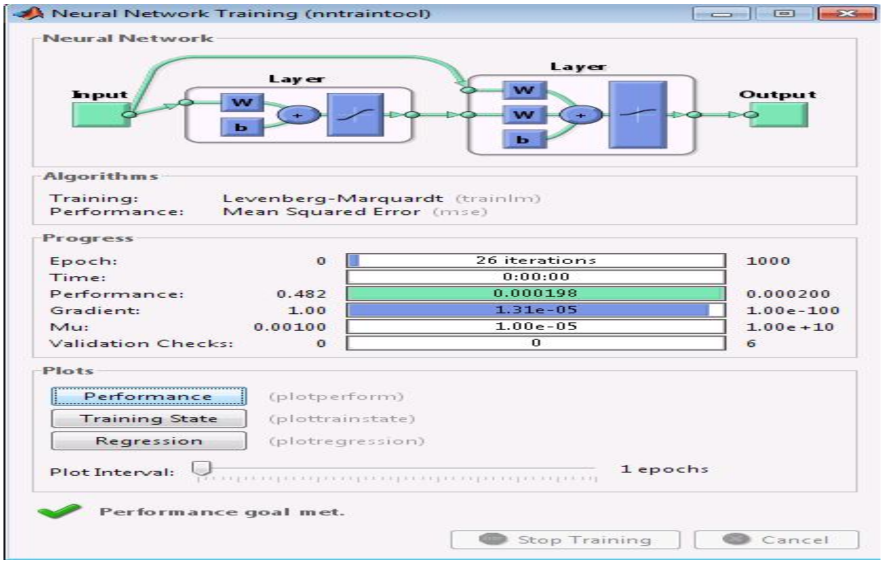


Figure (5): Linear correlation between the network output and measured pan evaporation for (1980).

Figure (6) depicts the elements of the ANNCF training. The training ended by reaching the lowest error value without excess number of iterations. The training time was too fast.



Algorithms

Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)

Progress

Epoch:	0	26 iterations	1000
Time:		0:00:00	
Performance:	0.482	0.000198	0.000200
Gradient:	1.00	1.31e-05	1.00e-100
Mu:	0.00100	1.00e-05	1.00e+10
Validation Checks:	0	0	6

Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Performance goal met.

Figure (6): The elements of the ANNCF training.

To check the responsiveness of the artificial network to the inputs circulated to every nodes of the hidden layer, and this wall already compute the activating value of the input and send it to every nodes in the output layer in order to compute it is activation a plot of the reference evapo- transpirations value in Mosul for 1980 estimated by the CCANN and the pan evaporation method, see Figure (7).

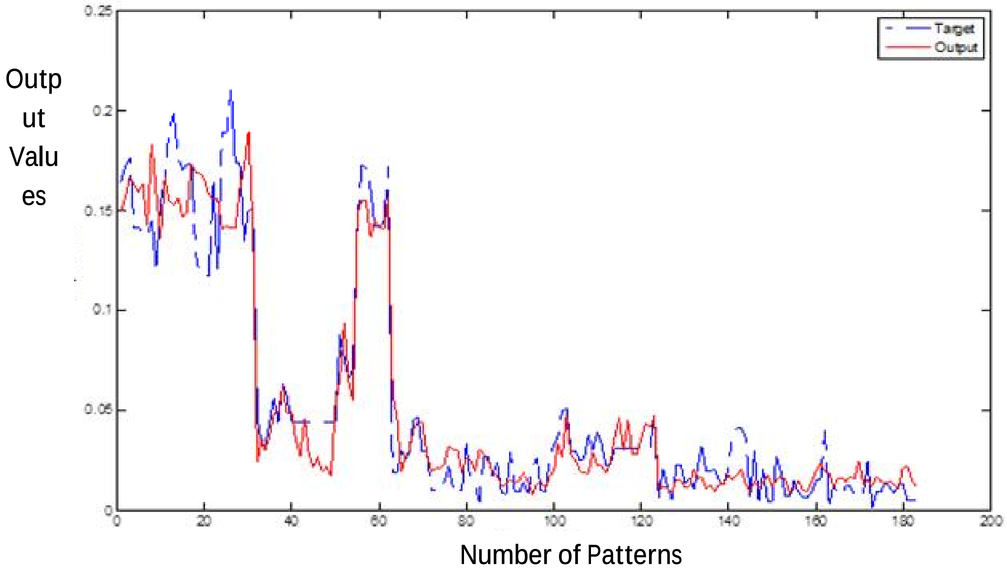


Figure (7): The outputs of the ANNCF with the targets.

It was illustrated the close relationship between ET_o from CCANN and ET_o compute by the pan evaporation .The test data was suitable as the highest error value between the output of the ANN and the target was 0.0590 and the lowest error value was 0.0000.The test data was also promoted when the ANN performance reached the square error average of 0.00029445 and the error was so closed to the training error.

CONCLUSION

The results obtained In this study suggests the following conclusions :

1. Estimating the daily reference evapo-transpiration is feasible through a mathematical built by application the cascade forward neural network technique (CCANN).
2. Use of Tansig type at the hidden layer and Logsig at the output layer gives the best performance for Square error average (MSE) by adoption a fixing epochs and fixing initialized weights also.
3. The results revealed a liner correlation between them with a correlation coefficient of (0.9679),and this indicate the possibility of using this technique in this field.

REFERENCES

[1]Trajkovic, S., Stankovic, M., and Todorovic B. 2000. Estimation of FAO Blaney-Criddle b Factor by RBF Network. Journal of Irrigation and Drainage Engineering 126(4), 268-271.

[2]P.S.Shrigure, G.S.Rajput. 2011. Evaporation modeling with neural networks –A Research review. (IJRRSIC).Vol. 1, No. 2, June 2011.

[3]Al-aani ,Iftekher 2007 Artificial Neural Network Model for Estimating daily Reference Evapo-transpiration in Mosul Region, PH.D. Thesis .College of engineering, University of Mosul/Iraq.

[4]Chauhan, S. and Shrivastava, R.K. 2008 Evapotranspiration Estimation Using Climate Based Methods and Artificial Neural Networks. Water Resources Management, 1-13, in press.

[5]Trajkovic, S. 2009a. Comparison of Radial Basis Function Networks and Empirical Equations for Converting from Pan Evaporation to Reference Evapotranspiration. Hydrological Processes, 23 (6), 874-880.

[6]Sudheer, K.P., Gosain, A. K., and Ramasastri K.S. 2003. Estimating Actual Evapotranspiration from Limited Climatic Data Using Neural Computing Technique. Journal of Irrigation and Drainage Engineering, 129(3), 214-218.

[7]M.J.Diamantopoulou,P.E .Georgiou , D.M.Papamichail.2011 Performance Evaluation of Artificial Neural Networks in Estimating Reference Evapotranpiration with Minimal Meteorological Data Global NEST Journal, Volume 13, No. 1, PP. (18_27).

[8]L. Fausett, 1994 Fundamental of Neural Networks, Architectures, Algorithms and Applications, Printice Hall Int. Snc..

[9]S. Suresh, S.N. Omkar, and V. Mani, 2005. Parallel Implementation of Back-Propagation Algorithm in Networks of Workstations, IEEE Transactions On Parallel And Distributed Systems, Vol. 16, No. 1, January 2005.