World Applied Sciences Journal 2 (6): 613-620, 2007
ISSN 1818-4952

Temperature Forecasting Based on Neural Network Approach

[1,2]Mohsen Hayati and [1]Zahra Mohebi

[1]Department of Electrical Engineering, Islamic Azad University-Kermanshah Branch, Kermanshah, Iran
[2]Department of Electrical Engineering, Faculty of Engineering, Razi university, Kermanshah, Iran

Abstract: This paper utilizes artificial neural networks (ANN) for one day ahead prediction of an important weather parameter i.e., temperature of Kermanshah city located in west of Iran. Our study based on Multi Layer Perceptron (MLP) which trained and tested using ten years past (1996-2006) meteorological data. In order to improve the accuracy of prediction, we split data into four seasons and then for each season one network is presented. The results show that MLP network has the minimum forecasting error for each season and can be considered as a good method for temperature forecasting.

Key words: Artificial neural networks · Temperature Forecasting · Multi-layer perceptron

INTRODUCTION

The art of weather forecasting began with early civilizations using reoccurring astronomical and meteorological events to help them monitor seasonal changes in the weather. Over the past few centuries, physical laws governing aspects of the atmosphere have been expressed and refined through mathematical equations. The idea of numerical weather forecasting-predicting the weather by solving mathematical equations was formulated in 1904 by Vilhelm Bjerknes (1862-1951, Norwegian) and developed by British mathematician Lewis Fry Richardson (1881-1953, British). Richardson's work highlighted the obvious fact that a large number of calculations had to be made very rapidly in order to produce a timely forecast. In the late 1940s, using one of the earliest modern computers, significant progress toward more practical numerical weather forecasts was made by a team of meteorologists and mathematicians at the Institute for Advanced Study (IAS) in Princeton, New Jersey. Mathematician John von Neumann (1903-1957, Hungarian-American) directed the construction of the computer and put together a team of scientists led by Jule Charney (1917-1981, American) to apply the computer to weather forecasting. Charney determined that the impracticality of Richardson's methods could be overcome by using the new computers and a revised set of equations, filtering out sound and gravity waves in order to simplify the calculations and focus on the phenomena of most importance to predicting the evolution of continent-scale weather systems. In April 1950, Charney's group made a series of successful 24-hour forecasts over North America and by the mid-1950s, numerical forecasts were being made on a regular basis. Modern technology, particularly computers and weather satellites and the availability of data provided by coordinated meteorological observing networks, has resulted in enormous improvements in the accuracy of weather forecasting.

The need for accurate weather prediction is apparent when considering the benefits that it has. These predictions may not stop a tornado, hurricane or flood, but they can help us prepare for one. A neural network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. During last four decades various complex problems like weather prediction [1, 2], heat transfer prediction [3], short term load forecasting [4-7], numerical simulation of nonlinear equations [8-10], etc has been proved to be areas with ample scope of application of this sophisticated mathematical tool. A multilayer neural network can approximate any smooth, measurable function between input and output vectors by selecting a suitable set of connecting weight and transfer functions. Despite so much of emphasis given to the application of ANN in prediction of different weather events all over the world, Iranian meteorological forecasters did not put much precedence on application this potent mathematical tool in atmospheric prediction. The objective of this study is to develop ANN-based models by using seasonal

Corresponding Author: Dr. Mohsen Hayati, Department of Electrical Engineering, Islamic Azad University-Kermanshah Branch, Kermanshah, Iran

Table 1: Meteorological variables

No.	Meteorological variables in the every 3 hours time frame	Unit
1	Wind speed	Knot
2	Wind direction	Deg
3	Dry Bulb temperature	Deg.C
4	Wet Bulb temperature	Deg.C
5	Relative humidity	%Rh
6	Dew point	Deg.C
7	Pressure	mb
8	Visibility	Km
9	Amount of cloud	octa

Table 2: Meteorological variables

No.	Daily meteorological variables	Unit
1	Gust wind	Knot
2	Mean temperature	Deg.C
3	Maximum temperature	Deg.C
4	Minimum temperature	Deg.C
5	Precipitation	mm
6	Mean humidity	%H
7	Mean pressure	mb
8	Sunshine	H
9	Radiation	Cal/m^2
10	Evaporation	mm

meteorological data of Kermanshah for temperature forecasting of this area. In comparison to our work about temperature forecasting [11], in this paper, we tried another method of choosing data (seasonal data) to improve the prediction accuracy and results show that this method has the minimum forecasting error.

Data collection: Weather data of ten years were collected from the meteorological department of Kermanshah, IRAN, which has shown in Table 1 and Table 2. Table 1 shows the part of data which has measured every 3 hours and Table 2 is daily value of variables.

Another variable which we got every 6 hours was soil temperature at 5, 10, 20, 30, 50 and 100 cm of soil depth. From these variables, hourly wind speed, dry and wet bulb temperature, relative humidity, pressure, daily sunshine and radiation variables are taken as inputs for ANN model.

The chosen weather data were split into four seasons namely spring, summer, fall and winter where for each season one network is considered. The general structure of inputs/output for each season is shown in Fig. 1.

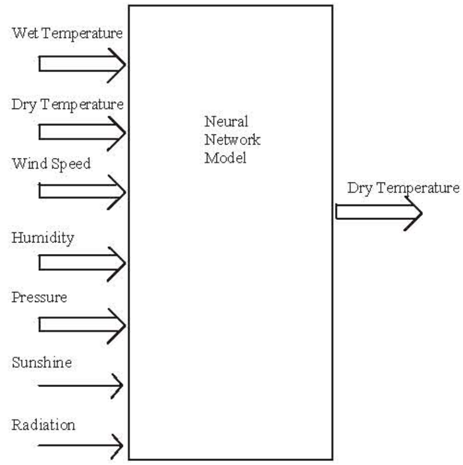

Fig. 1: General structure of inputs/output for each season

The global set of patterns is divided into two randomly selected groups, the training group, corresponding to 65% of the patterns and the test group, corresponding to 35% of patterns; so that the generalization capacity of network could be checked after training phase. Two random days in each season are selected as unseen data which have not been used in train group. We used the Mean Absolute Error (MAE) as a measure of error made by the neural network,

$$MAE = \frac{1}{M_{total}} \sum_{i=1}^{M_{total}} \left| P_i - P_i^* \right|$$

Where P_i, P^*_i and M_{total} are exact values, predicted values and total number of the test data respectively.

Meteorological parameter which we want to forecast, for each season and for each year have vast variation, for example figure 2 shows this variation over ten falls (1996-2006).

Neural Network Model: A neural network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways:

I- A neural network acquires knowledge through learning.
II- A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

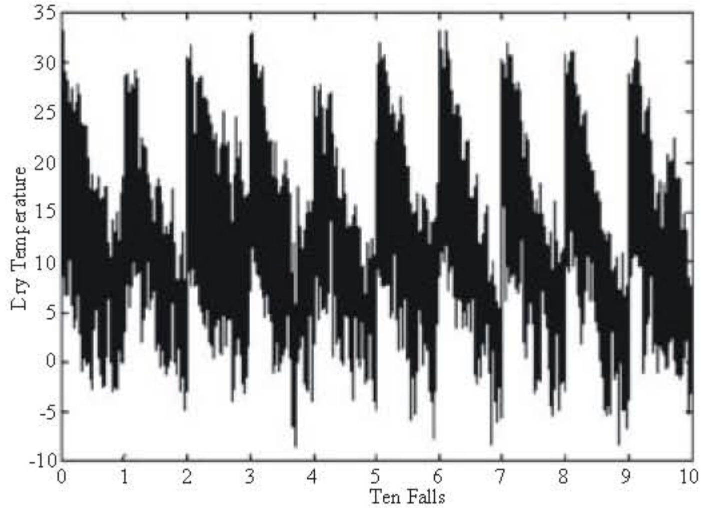

Fig. 2: Temperature variations over ten falls (1996-2006)

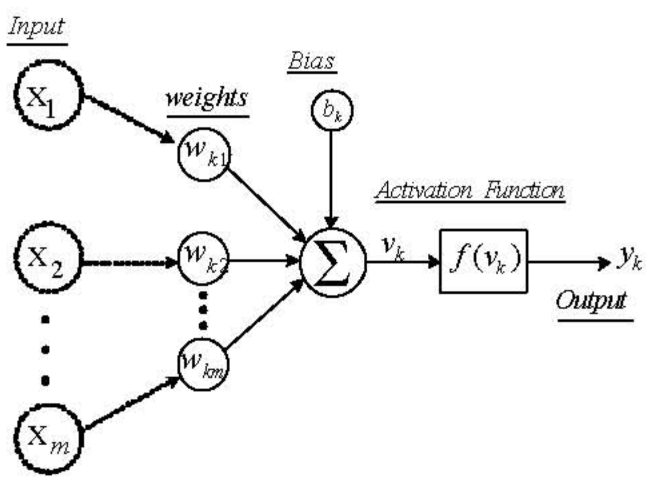

Fig. 3: Neuron model

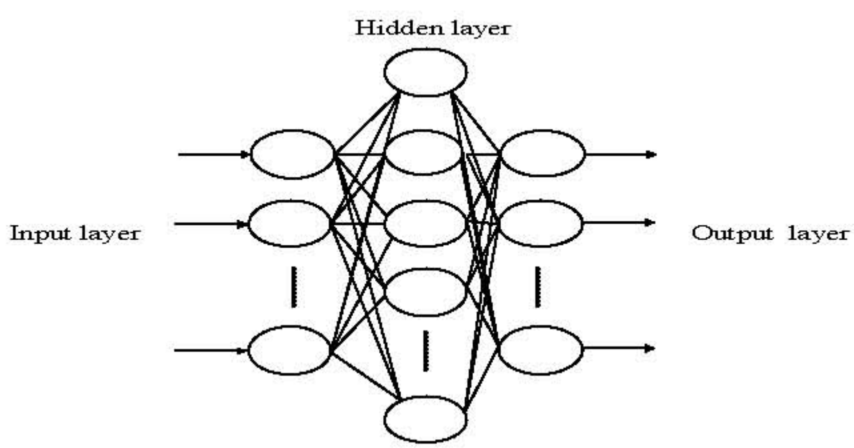

Fig. 4: Three layer MLP network

The true power and advantage of neural networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics. In this paper, one model of neural network is selected among the main network architectures used in engineering. The basis of the model is neuron structure as shown in Fig. 3. These neurons act like parallel processing units. An artificial neuron, is a unit that performs a simple mathematical operation on its inputs and imitates the functions of biological neurons and their unique process of learning. From Fig. 3 will have,

$$v_k = \sum_{j=1}^{m} x_j w_{kj} + b_k$$

The neroun output will be

$$y_k = f(v_k)$$

MLP Network: The most common neural network model is the multilayer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown. A graphical representation of an MLP is shown in Fig. 4.

Its units each perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their input and the units are arranged in a layered feed forward topology. The network thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases) the free parameters of the model. Such networks can model functions of almost arbitrary complexity with the number of layers and the number of units in each layer, determining the function complexity. Important issues in Multilayer Perceptron design include specification of the number of hidden layers and the number of units in these layers [12-14]. Once the number of layers and number of units in each layer, has been selected, the network's weights and thresholds must be set so as to minimize the prediction error made by the network. This is the role of the training algorithms. The best known example of a neural network training algorithm is back propagation [12, 15]. Modern second-order algorithm such as conjugate gradient descent and Levenberg-Marquardt [13] are substantially faster for many problems, but Back propagation still has advantages in some circumstances and is the easiest algorithm to understand. With this background we designed and trained these networks as follow:

The three-layer network with sigmoid transfer function for hidden layer and linear transfer function for output layer can represent any functional relationship between inputs and outputs, if the sigmoid layer has enough neurons [14] so we chosen this structure for all four networks. Back propagation training algorithms are often too slow for practical problems, so we can use several high performance algorithms that can converge from ten to one hundred times faster than back propagation algorithms. These faster algorithms fall into two main categories: heuristic technique (variable learning rate back propagation, resilient back propagation) and numerical optimization techniques (conjugate gradient, quasi-Newton, Levenberg-Marquardt). We tried several of these algorithms to get the best result. Levenberg-Marquardt is the fastest algorithm [16] but as the number of weights and biases in the network increase, the advantage of this algorithm decrease, so we tried another algorithm which perform well on function approximation and converge rather fast. From these algorithms, scaled conjugate gradient was suitable for our purpose. From an optimization point of view learning in a neural network is equivalent to minimizing a global error function, which is a multivariate function that depends on the weights in the network. Many of the training algorithms are based on the gradient descent algorithm. Minimization is a local iterative process in which an approximation to the function, in a neighborhood of the current point in the weight space, is minimized. Most of the optimization methods used to minimize functions are based on the same strategy. The Scaled Conjugate Gradient (SCG) algorithm [17] denotes the quadratic approximation to the error E in a neighborhood of a point w by:

$$E_{qw}(y) = E(w) + E'(w)^T y + \frac{1}{2} y^T E''(w) y$$

In order to determine the minimum to E_{qw} *(y)* the critical points for E_{qw} *(y)* must be found. The critical points are the solution to the linear system defined by Moller [17].

$$E'_{qw}(y) = E''(w)y + E'(w) = 0$$

SCG belongs to the class of Conjugate Gradient Methods, which show superlinear convergence on most problems. By using a step size scaling mechanism, SCG avoids a time consuming line-search per learning iteration, which makes the algorithm faster than other second order algorithms. And also we got better results than with other training methods and neural networks tested, as standard back-propagation.

Neural networks generally provide improved performance with the normalized data. The use of original data as input to neural network may cause a convergence problem. All the weather data sets were therefore, transformed into values between -1 and 1 through dividing the difference of actual and minimum values bye the difference of maximum and minimum values subtract by 1. At the end of each algorithm, outputs were denormalized into the original data format for achieving the desired result. We know that from one initial condition the algorithm converged to global minimum point, while from another initial condition the algorithm converged to a local minimum so it is best to try several different initial condition in order to ensure that optimum solution has been obtained [14, 18, 19]. For a network to be able to generalize, it should have fewer parameters than there are data points in the training set [14]. Training goal for the networks was set to 10^{-4}. Finding appropriate architecture needs trial and error method. Networks were trained for a fixed number of epochs. Performance of the network was evaluated by increasing the number of hidden neurons. After finding hidden neurons, epochs increase till we find the suitable epochs.

RESULTS AND DISCUSSION

The obtained optimal MLP structure for each season is shown in Table 3.

Two random days in each season are selected as unseen data. Therefore the exact and predicted values of Temperature for each unseen days at each season for MLP network is shown in Fig. 5.

The minimum and maximum error between exact values and predicted values of unseen days for each season is shown in Table 4.

Table 3: The optimal MLP structure for each season

MLP	spring	summer	autumn	winter
Number of hidden neuron	4	6	4	4
Number of epochs	2000	2000	2000	2000
Activation function used in hidden layer	tan-sig	tan-sig	tan-sig	tan-sig
Activation function used in output layer	pure linear	pure linear	pure linear	pure linear

Table 4:

Temperature	Spring	Summer	Fall	Winter
MLP Minimum Error	0.001	0.0148	0.0336	0.0019
MLP Maximum Error	0.3569	1.6417	0.7896	0.5679

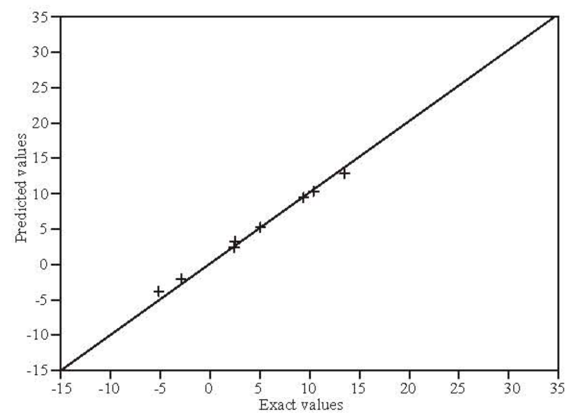

Fig. 5a: Comparison between exact and predicted temperature values for unseen days i.e., 22-Feb-1997

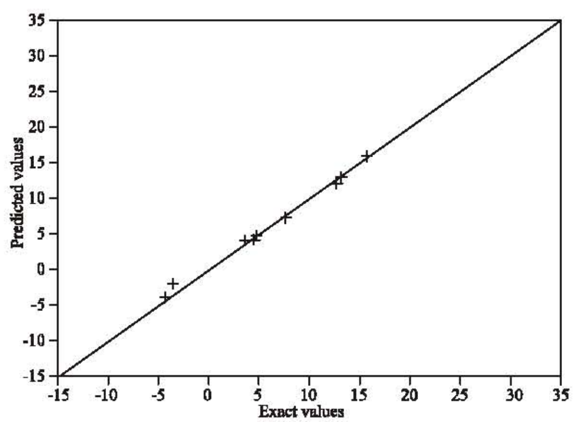

Fig. 5b: Comparison between exact and predicted temperature values for unseen days i.e., 5-Mar-1998

After training phase we tested each network by using about 35% of patterns so that the generalization capacity of each network could be checked.

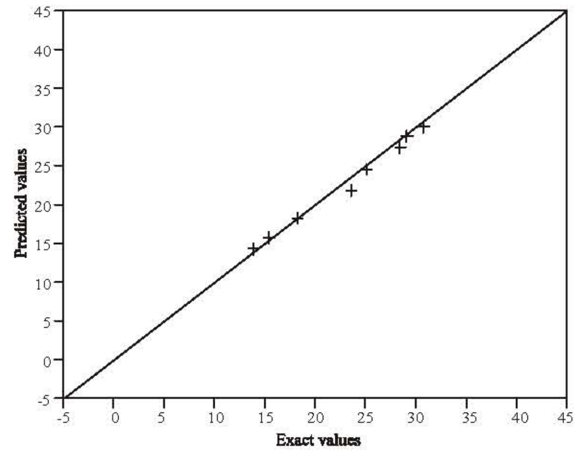

Fig. 5c: Comparison between exact and predicted temperature values for unseen days i.e., 26-May-1999

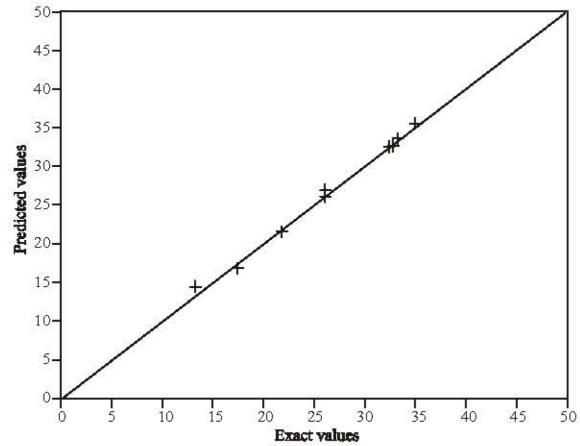

Fig. 5f: Comparison between exact and predicted temperature values for unseen days i.e., 4-Sep-2003

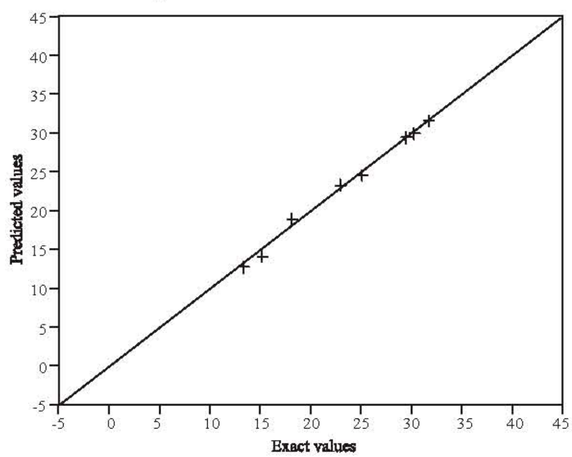

Fig. 5d: Comparison between exact and predicted temperature values for unseen days i.e., 9-June-2000

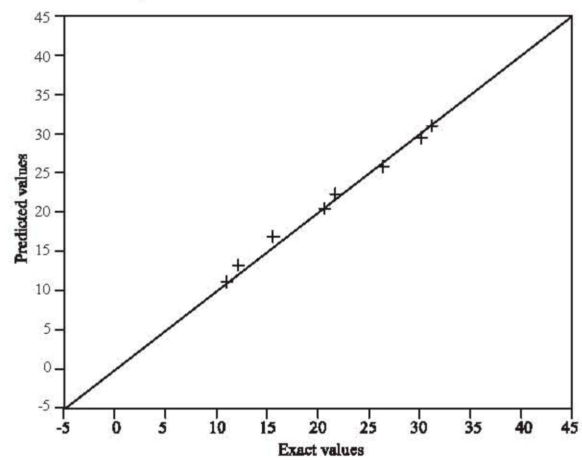

Fig. 5g: Comparison between exact and predicted temperature values for unseen days i.e., 12-Oct-2005

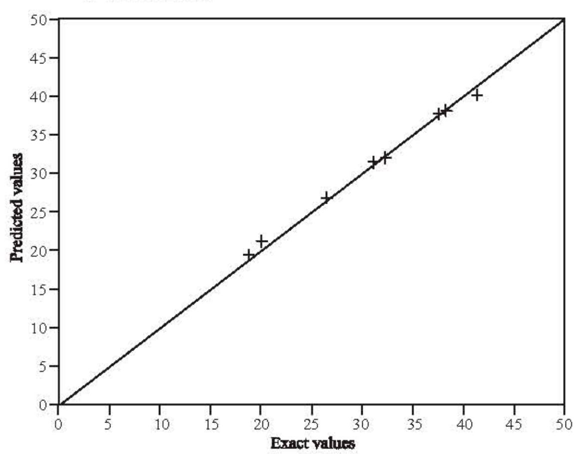

Fig. 5e: Comparison between exact and predicted temperature values for unseen days i.e., 28-Jul-2001

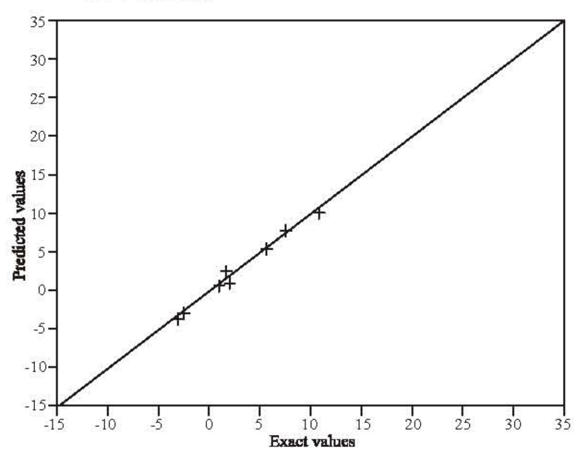

Fig. 5h: Comparison between exact and predicted temperature values for unseen days i.e., 21-Dec-2006

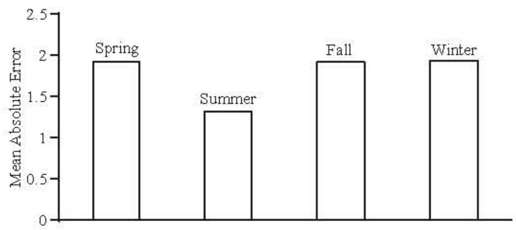

Fig. 6: Mean Absolute Error for seasonal test data of ten years

By considering these results we can evaluate this method. We used the Mean Absolute Error (MAE) as a measure of error. Figure 6 shows the seasonal results using MAE (Mean Absolute Error).

CONCLUSIONS

The results of neural network models based on seasonal prediction i.e., spring, summer, fall and winter for one day ahead temperature forecast for the Kermanshah city, Iran, shows that MLP network with this structure has minimum error between exact and predicted values at each day and has a good performance, reasonable prediction accuracy and minimum prediction error in general. The forecasting reliability was evaluated by computing the mean absolute error between the exact and predicted values. The results show that this network can be an important tool for temperature forecasting.

ACKNOWLEDGEMENTS

We would like to express our gratitude to Mr. A. Karimi, the manager of west electric company, Mr. T. Hyderi manager and Mr. K. Ghasemi, deputy manager of west of Iran Meteorological Research Center for providing us the required data.

REFERENCES

1. Maqsood, I., M.R. Khan and A. Abraham, 2000. Intelligent weather monitoring systems using connectionist models. International Journal of Neural, Parallel and Scientific Computations, No. 10: 157–178.
2. Maqsood, I., M.R. Khan and A. Abraham, 2002. Neuro-computing based Canadian weather analysis, The 2nd International Workshop on Intelligent Systems Design and Applications, Computational Intelligence and Applications. Dynamic Publishers, Atlanta, Georgia, 39-44.
3. Hayayi, M., T. Yousefi, M. Ashjaee, A.S. Hamidi and Y. Shirvany, 2007. Application of Artificial Neural Network for Prediction of Natural Convection Heat Transfer from a Confined Horizontal Elliptic Tube" International Journal of Applied Sciences, Engineering and Technology, 4(3): 157-162.
4. M. Hayati and B. Karami, 2005. Application of Neural Networks In Short-Term Load Forecasting, 7th WSEAS international conference on Mathematical Methods and Computational Techniques in Electrical Engineering, Sofia, Bulgaria, 37-41.
5. Hayati, M., 2007. Short Term Load Forecasting Using Artificial Neural Networks for the West of Iran, Journal of Applied Sciences, 7(12): 1582-1588.
6. Hayati, M. and Y. Shirvany, 2007. Artificial Neural Network Approach for Short Term Load Forecasting for Illam Region" International Journal of Electrical, Computer and System Engineering, 1(2): 121-125.
7. Hayati, M. and B. Karami, 2005. Application Of Computational Intelligence In Short-Term Load Forecasting, WSEAS Transactions On Circuits and Systems, 1(4): 1594-1599.
8. Hayati, M., B. Karami and M. Abbasi, 2007. Numerical Simulation of Fuzzy Nonlinear Equations by Feedforward Neural Networks, World Applied Sciences Journal, 2(3): 229-234.
9. Hayati, M. and B. Karami, 2007. Feedforward Neural Network For Solving Partial Differential Equation" Journal of Applied Sciences, 7(19): 2812-2817.
10. Shirvani, Y., M. Hayati and R. Moradian, 2007. Numerical Solution of the Nonlinear Schrodinger Equation by Feedforward Neural Network" Communication in Nonlinear Science and Numerical Simulation, ScienceDirect (Article in Press).
11. Hayati, M. and Z. Mohebi, 2007. Application of Artificial Neural Network for Temperature Forecasting" International Journal of Applied Sciences, Engineering and Technology, 4(3): 163-167.
12. Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. New York: Macmillan Publishing.
13. Bishop, C., 1995. Neural Networks for Pattern Recognition: University press.
14. Hagan, M.T., H.B. Demuth and M.H. Beale, 1996. Neural Network Design. PWS Publishing Company, Boston, Massachusetts.
15. Patterson, D., 1996, Artificial neural networks. Singapore, Prentice Hall.

16. Zurada, J.M., 1992, Introduction to Artificial Neural Systems, West Publishing Company, Saint Paul, Minnesota.
17. Moller, A., 1993, Scaled Conjugate Gradient Algorithm for Fast Supervised Learning, Neural Networks, 6 (4): 525-533.
18. Moody, J. and C. Darken, 1989. Fast learning in networks of locally-tuned processing units". Neural Computation, 1(2): 281-294.
19. Vander, S.P.P., 1994, Minimization methods for training feedforward neural networks. Neural Networks, 1: 1–11.