Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 5, Issue. 11, November 2016, pg.82 – 88

Improving the Artificial Neural Network Model of Temperature Prediction for the City of Baghdad

Eman Salim Ibrahim Harba

Computer unit and internet/College of Arts/ University of Baghdad, Iraq
emanharba_121212@coart.uobaghdad.edu.iq

Abstract— In this work, an Artificial Neural Networks (ANN) model for prediction the daily maximum and minimum temperature of July was developed. Using weather data recorded by underground weather service during 2010-2014. The training used 155 patterns. The impacts of the number of neurons (nodes) and the iterations of training (number of tours) on model performance were investigated. Two ANN models for the prediction of daily high temperature (Tmax) and low temperature (Tmin) were trained in the same manner. In performance testing, the model-predicted temperatures of the 31 days of July for the year of 2015. The results for both the Tmin and Tmax were compared with actual data recorded by Iraqi Meteorological Organization & Seismology constitution in 2015. The resulting ANN model from this study serves as a prototype for the daily temperature forecasting model which is suitable for Baghdad. It provides information that could be beneficial for weather forecasters as well as other in the related field. The model could be further developed and applied to other regions.

Keywords— Weather Forecasting, Temperature Prediction, Temperature Forecasting Artificial Neural Network, Back-Propagation.

I. INTRODUCTION

Weather predicting is a technique to forecast the state of the weather for any specific location. The weather predictions are designed by gathering quantitative data concerning the current condition of the weather and utilizing scientific knowledge of atmospheric strategies to undertaking how the natural environment will change. Most of the live systems depend on atmospheric condition to make essential improvements in their systems. Predicting helps to take required measures to avoid damage to property and life to a significant extent. Quantitative predict like rainfall, humidity and temperature are very important in agriculture area, in addition to merchants within commodities markets. Temperature predictions are utilized by utility companies to determine desire over future. Because outdoor tasks are seriously limited by chill, snow and heavy rain, the predictions could be used to program tasks close to these types of events, and also to plan ahead and survive individuals [1].

Currently, several computing approaches are presented which can be employed for predicting improving its accuracy. Various types of predicting techniques are found such as Artificial intelligence methods, Causal or

econometric forecasting methods, Quantitative and Qualitative method, Judgmental methods, Time series methods, Naive approach, etc. [2].

The weather predicting is live predicting in which creation of the model are usually necessary for daily weekly or monthly weather guide and plans. Therefore, the accuracy of predicting results is a necessary consideration in this predicting. Various issues are reviewed which is often considered to obtain the accurate results. Intelligent computing is most important technique one among of these methods which is able to check the nonlinear data and create some rules and patterns to analyse and train from the observed data to forecast weather in the future. Using ANN will provide results which are more reliable. At this point, the error might or might not minimize entirely. However, the accuracy will enhance when compared with prior forecasts. In this paper we review multiple literature on weather forecasting that used ANN methods and introduces different terms about the neural network. We have been prepossessed the ANN model considering the technical specs for predicting weather using a high level of accuracy [3].

II. PREVIOUS WORKS

In some recent studies, the temperature prediction models will go ahead, as an example temperature a couple of hours ahead or maximum and minimum temperature of any given day. Many works have been performed and various ANN models have been tested. The studies with significant results are as follows.

Hayati and Mohebi 2007 [4], have created temperature prediction model using ANN for Kermanshah city, Iran. In their study the Multi-Layer Perceptron (MLP) technique was used with 10 years (1996-2006) meteorological data for training and testing. The input data for their models included wet and dry temperatures, wind speed, humidity, pressure, sunshine duration, and radiation. Input Data were divided into 4 parts by seasons. The ANN was run using 2,000 number of tours. The resulting model showed minimum prediction error and considered a good model for one-day temperature predictions.

Sanjay et al. 2007 [5], is focused on forecasting the relative humidity and forecasting the minimum and maximum temperature by using time series researching. The ANN model that used is a Multilayer feed forward artificial neural network (FFMLP ANN) with back propagation learning. Statistical and direct input parameters as well as the period are compared. For maximum/minimum temperature predicting the maximum appears to be a 15-weeks duration of input data. Input features are maximum and minimum features, respectively. Specifically, these features are rate of change, oscillator, exponential moving average, moving average, and the third moment. For the Fifteen-weeks period the error was under 3%. The most important result is that generally statistical parameters could be used to extract trends. Great parameters are moments, exponential moving average, moving average, rate of change and oscillator. kurtosis and Skewness didn't work well.

In 2009, Dombayc and Golcu [6], designed a model for prediction the daily mean temperature for Denizli City, Turkey, by used ANN. Three years (2003-2005) of meteorological data were used in training the model while the data from 2006 were used for testing. The study found that the model best performed when the number of hidden neurons was twice the number of inputs.

A fully linked, FFMLP ANN of three layer for temperature forecast has been also presented by Baboo and Shereef 2010 [7]. Their study investigated the application of Back Propagation Algorithm (BPA) technique in creating temperature prediction models. Five daily meteorological elements used in their study were pressure, temperature, humidity, wind velocity, and wind direction. Two hundred set of data from 2009 were used in training with the testing done on other unseen data. Various numbers of tours in the range of 1,000 to 5,000 were investigated. The study found that the ANN/BPA temperature forecast models had a good performance. In regard to the number of tour, the study showed that more tour produced less error.

In 2012, Devi et al. [8], 2012 proposed an ANN based algorithm for forecasting the temperature. The BPNN has been used since it can quite approximate a big class of functions. Authors suggest a model that can take real-time dataset with 15 parameters as input, that is then normalized by using max-min normalization to scale data between 0 to 1. Then it's trained and tested employing the BPNN. The results are comparison with the department of meteorological to figure out least error and reliability of the model. They found that the model offers the possibility of temperature predicting.

III. MYTHOLOGY

The daily temperature of Baghdad for month of July are obtained from weather data recorded by underground weather service during 2010-2014 in addition to data recorded by Iraqi Meteorological Organization & Seismology constitution in 2015 that used to comparison. Parameters used in the forecast included maximum temperature (Tmax) and minimum temperature (Tmin). Due to the different types of data gained, it was necessary for these data to be standardized before using them. The standardization not only made the training process easier but also helped in forming the motivation function. This paper develops BPNN model step-by-step to forecast the minimum and maximum temperature in Baghdad by studying the data that offered by underground weather website.

The 3-layer Back Propagation Algorithm was used to create the ANN models for this study. The 3 layers included one input layer, one hidden layer, and one output layer. All of the above factors were stored in the model for the process of generation of a 3-day forecast as outlined in Figure 1. In the hidden layer, the transfer of tangent sigmoid released a one-day highest or lowest temperature as the model output.

The aim is to gather dataset consisting weather parameters like maximum minimum temperature, and perform all required data preprocessing tasks. Dataset is gathered from a well-known website of underground weather service [9] this normalized data is passed to the back propagation neural network (BPNN) to learn the NN. Therefore, the NN is trained by updating all of the biases and weights as obtained errors in each iteration until it will stop when reach to minimal error or selected iteration. Testing results of network that trained by used BPNN model with the developed structure can achieve good forecast with least error. Similar to this all of the ANN forecasting algorithms is effective in weather forecasting. To enhance the performance speed and accuracy of BPNN we intend to use large data for many years to increase performance of model. Finally, we calculated the mean values and treat them to be the last forecast values [10].

IV. A PROPOSED BPNN MODEL FOR PREDICTION MAXIMUM AND MINIMUM TEMPERATURE OF BAGHDAD

The back propagation consists of one input layer, one output layer of Back Propagation in our model Is present in figure (1). Only the neuron of the input layer receives external input components and distributes them, without modification, to all other neurons of the next layer, the hidden layers is called hidden because it is not directly connected to the outside World. Each hidden layer acts as layer of "feature detector", neurons that respond to specific feature in input pattern. These feature detectors are organized as learning takes presented as shown in figure (1): [11].

X0: Bias node for input layer
X1: input node
H0: Bias node for hidden layer
Hj: output for hidden layer
Vij: weights between the input and hidden layers
Wjk: weights between the hidden and output layers
N1: number of the input neuron
N2: number of the input hidden neuron
N3: number of the output neuron
I: index of the input layer,0<i≤n1
J: index of the input layer, 0<j≤n2
K: index of the input layer, 0<k≤n3

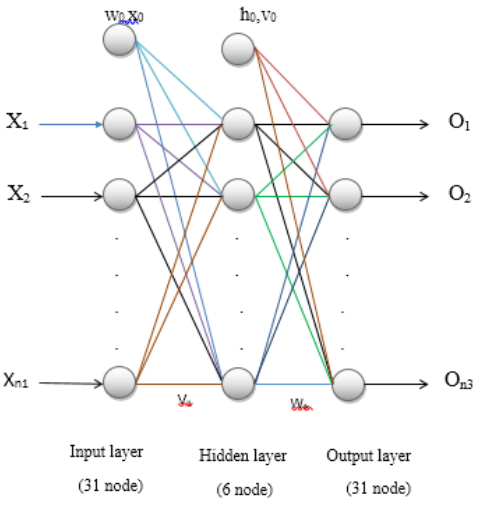

Figure (1): structure of 3-layers backpropagation neural network model [11]

V. ALGORITHM OF BACKPROPAGATION

The algorithm of BPNN that used in our model is based on of the study done by Fausett [94] and Zuiada [97], which are described in follows: [11]

1. **Initialization:** set all weights and biases of the network to small random numbers. Note that the activation of node biases is fixed at 1.

2. **Presentation of training example:** the network will train using selected samples training select (or instances), it is expected that the network will learn in one cycle of the algorithm. Step3 and 4 below are performed for every training sample (instance).

3. **Forward Pass:** calculation of activation:

   A. The activation level of an input unit (Si) is determined by the instance present to:

   $Si = Xi, \ (0 \leq i \leq nin)$ ….. (1)

   Where: Xi: input nodes (i) input the layer including bias node(X_0=1), nin: number of input nodes.

   B. The activation level (hj) of a hidden unit and the output(ok) of an output unit is determined by the equations given (Unipolar Sigmoid function)

   $$f(net) = \frac{1}{1+e^{-\lambda net}} \quad \text{where } \lambda \geq 1 \text{ ……..…….. (2)}$$

   $$h_j = \mathrm{f}\left(net_j\right) = f(\sum_{i=0}^{n_{in}} V_{ij} X_i \,) \quad \text{……. (3)}$$

   Where: h_j: output of the node(j)in the hidden layer including bias node(h_0=1), net_j=the activation output of the hidden node(j)in the hidden layer, n_{hid}=number of hidden nodes, v_{ij}=the weight of connection from node(i)in the input layer to node(j) in the hidden layer including bias node weight(v0j)

   $$o_k = f(net_k) = f(\sum_{j=0}^{n_{hid}} W_{jk} h_j) \text{ …….. (4)}$$

   Where $1 \leq k \leq nout$, net_k: is the activation output of the output node(j)in the output layer, n_{out}= number of output nodes, O_k: output of the node(k)in the output layer, w_{ij}: is the weight of connection from node(j)in the hidden layer to node(k) in the output layer including bias node weight(w_{0j})

4. **Back word Pass (Weight Training)**

   A. compute the difference between the actual output and the desired output in order to calculate the errors of the output neurons s_k and the of hidden neurons s_j errors as describe by the following equations:

   $$s_k = o_k(1 - o_k)(d_k - o_k) \quad \text{……. (5)}$$

   $$s_j = h_i\,(1 - h_i) \sum_{k=1}^{n_{out}} \left(s_k . v_{jk}\right) \quad \text{…… (6)}$$

   B. use the resulting difference s_k and s_j to calculate the changed in weight. Δw_{jk} and Δv_{ij} as shown by the following equation:

   $$\Delta w_{jk}^{(t)} = \eta . s_{k.h_j} \quad \text{…… (7)}$$

   $$\Delta v_{ij}^{(t)} = \eta . s_j . s_i \quad \text{……..…….. (8)}$$

   Sometimes, the convergence need to be faster, this could be done by adding momentum term to both of Δw_{jk} and Δv_{ij} as given in the following equation:

   $$\Delta w_{jk}^{(t+1)} = \eta . s_k . h_j + \alpha . \Delta w_{jk}^{(t)} \quad \text{……...…….. (9)}$$

$$\Delta v_{ij}^{(t+1)} = \eta . s_j . s_i + \alpha . \Delta v_{ij}^{(t)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

C. Adjust the weights by using the following equations

$$w_{jk}^{(t+1)} = w_{jk}^{(t)} + \Delta w_{jk}^{(t)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (11)$$

$$\boldsymbol{v_{ij}^{(t+1)} = v_{ij}^{(t)} + \Delta v_{ij}^{(t)}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (12)$$

5. **Iteration:** Steps 2-4 are iterated until reach the convergence (in terms of the selected error criterion)

6. **Equation for predication and compute error (Performance Measurement):** the error (predication) of neural network is compute by the equation: -

P = (dk - ok) ……...…..…… (13)

$$error = \sum (d_k - o_k)/d_k \quad \ldots\ldots\ldots\ldots\ldots\ldots (14)$$

Where: dk the desire output for k cell, O_k: the actual output for k cell

7. **Learning Rate:** here used an adaptive learning rate:

$$\eta^t = \eta_0 \left(1 + \frac{t}{T_{max}}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (15)$$

Where: η^t: is the learning rate at time/iteration t (0<η<1), η_0: is the initial learning rate, t: iteration, T_{max}: total number of iteration

VI. Experiment and Result

To test the proposed model, a pattern of dataset has been taken (as described before) from Weather Underground. This dataset has the real-time records of the weather for a specific period of time. For this purpose, a record data of the full Juley days of earlier 5 years from 2010 to 2014 has been taken. Figure (2), illustrated the distribution of temperatures for 31 days of Juley for five years of (2010, 2011, 2012, 2013 and 2014).

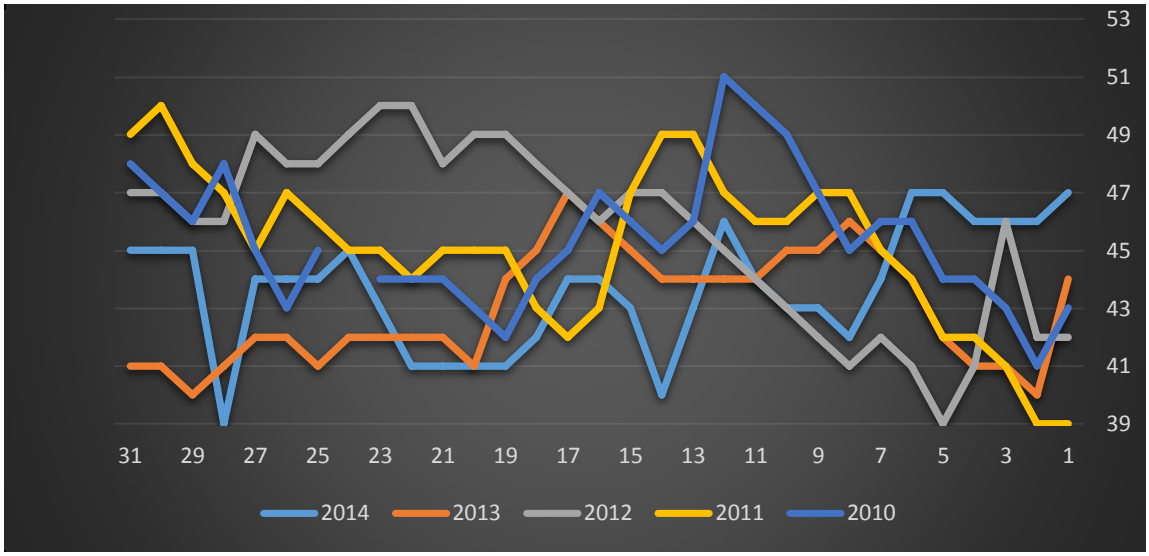

Figure (2): The temperatures of the 2010.2011,2012,2013 and 2014 years

The dataset contains Temp attributes both maximum and minimum temperature. The learning rate has considered to be 0.8. A 3 layered FFNN with BP has been designed. The initial weights are selected randomly in range between 0 to 1. After each training (each iteration), the network has been tested for their performance on validation data set. The training operation is finished once the overall performance achieved the maximum on validation data set. After testing and training, the forecasting error values are calculated for every model. Then the model will predict the temperatures of July days in 2015. These production temps have been compared with real data recorded by Iraqi Meteorological Organization &

Seismology constitution in 2015. Figure (3) and figure (4) shows the comparison result between predication data (result) and real data.

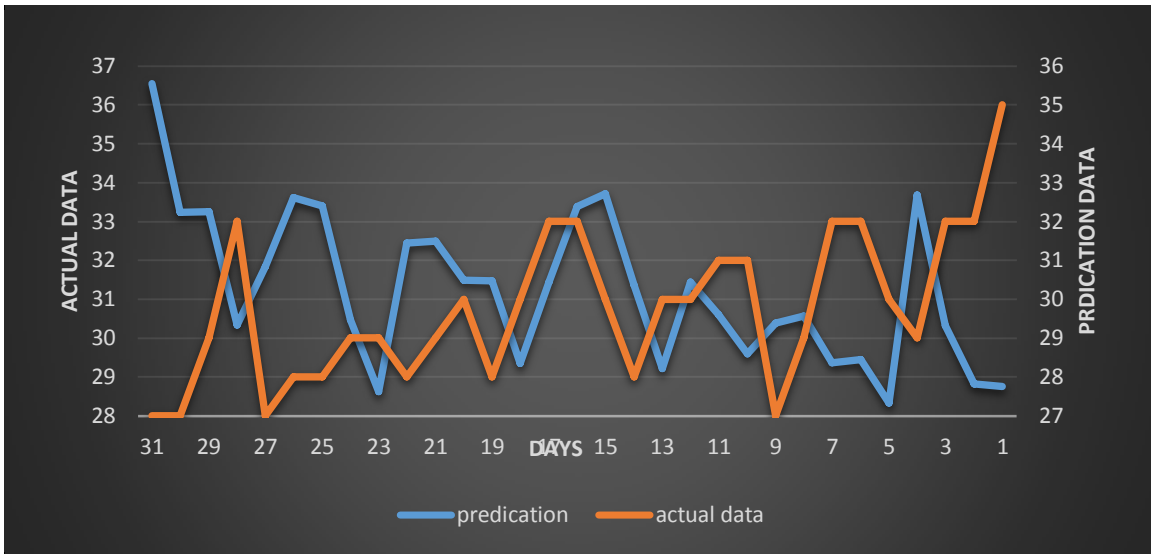

Figure (3): show minim temperature between predication data(result) and actual data (2015)

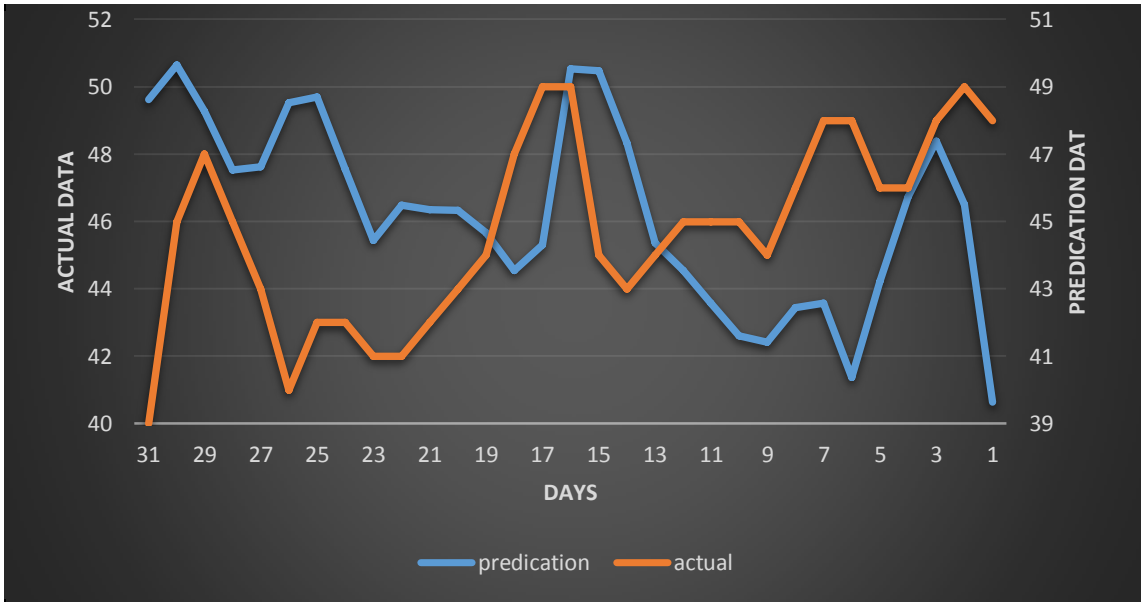

Figure (4): show maximum temperature between predication data(result) and actual data (2015)

VII. CONCLUSIONS

In this paper, an ANN with backpropagation model has been used to predicate the daily high temperature and low temperature for city of Baghdad. After each iterative process, it has been compared observed output with targeted output repeatedly, then it calculates the error which has been used to readjust the weights values and bias to obtain a better output. As a result, this approach attempts to reduce the error. From experimental result it appeared that there is a convergent result between predicate results and real records which near to be close in some days Thus, ANN-BP algorithm is definitely most suitable technique for predicting weather accurately. Also, this model can be extended to predict other weather parameters like rainfall, cyclonic storms, hurricanes, snowfall and tornadoes.

REFERENCES

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.

[4] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.

[5] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[6] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[7] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[8] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.

[9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[10] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

