Efficient Weather Forecasting using Artificial Neural Network as Function Approximator

I. El-Feghi , Z. Zubia and S. Abozgaya

Abstract—Forecasting is the referred to as the process of estimation in unknown situations. Weather forecasting, especially air temperature, is one of the most important factors in many applications. This paper presents an approach to develop Artificial Neural Networks (ANNs) to forecast air temperatures. One important architecture of neural networks namely Radial Basis Function (RBF) will be used as a function approximator. The RBF trained using meteorological data of one year and tested on another year. The data consist of observations of various meteorological variables such as relative humidity, dew point, wind speed, wind direction and air pressure.

To come up with appropriate centers for the RBF neurons, the weather data was clustered into several groups using k-means clustering algorithm. The goal of developing this network is to forecast the air temperature and to have a regression model with minimum error of prediction

The data of one year is used for supervised training using labelled data while the other year data was used of testing the trained ANN. Several testes were run to come up with the most suitable ANN structure based on lowest MSE in predicting air temperature. Results have shown that these structures gave very good prediction in term of accuracy.

Keywords— Artificial neural network, weather forecasting, , Radial Basis Function, classification, k-means

.

I. Introduction

Forecasting is the process of estimation in unknown situations. Prediction is a similar, but more general term. Weather forecasting is one of the most important factors in many fields, especially air temperature since it touches the life of to human, cattle, and agriculture. Traditionally the role of providing weather forecasts has been the responsibility of the atmospheric meteorological national centers.

While data required to make temperature predictions has been available for quite some time, the complex relationships between the data and its effect on the approximation of temperature has often proved to be difficult using conventional computer analysis [1]. An Artificial Neural Network (ANN) that mimics the behavior of neurons in the brain. The basic components of an ANN are its nodes or neurons and the connections between the nodes. A node is primarily a computational unit. It receives inputs, calculates a weighted sum and presents the sum to an activation function. The nodes are generally arranged in layers [2].

The use of a neural network, however, which learns rather than analyses these complex relationships, has shown a great deal of promise in accomplishing the goal of predicting air temperature with high accuracy [3-5]. ANNs have the advantage of their ability to learn and adapt. They have been extensively used to predict the air temperature [3][4] and Forex market prediction [6][7] and stock market[8].

There are several types of ANN structure such as Multi-Layer Perceptron(MLP), Radial Basis Function (RBF) and Support Vector Machine(SVM)[2]. For an ANN to be useful it should be able to learn or capture the complex relationships between inputs and outputs. This is done by searching for an optimal set of the weights of the connections between the nodes. It is achieved by first sending one set of inputs in the feed forward mode through the ANN.

In this research RBF was selected as a mean of forecasting. The selection of forecasting methods depends on several factors, such as the availability of data, the accuracy required and the ease of operation [9]. However forecasting is the process of estimation in unknown situations from the historical data. For example forecasting weather, use historical data as the basis for estimating future outcomes [4].

Before RBF can be used, the location of the centers and widths for the hidden layer and the weights of the output layer must be determined [2]. To come up with the best centers K-means algorithm is used to cluster the data into several groups [10].

The goal of this research is to highlight how the application of RBF Artificial Neural Networks (RBFANNs) can be used to forecast air temperatures around the clock for a selected region in the Mediterranean sea for which, past weather data is available.

II. Data Preparation

The weather data for this research were collected from the

I.El-Feghi is a professor at University of Tripoli, Tripoli-Libya, (Phone:+218-91-178-7206; e-mail idrise@ee.edu.ly).

Z. Zubia is with University of Sirit, Sirit, Libya (e-mail zszubi@su.edu.ly).

S. Abnozgaya is with the Arabian Cement Company, AlKhomas-Libya (e-mail: S.Abozgaya@yahoo.com).

meteorological department of Misurata, Libya for the year 2006 through 2007. The data represents the selected the weather pattern of Misurata city. The data of January through December for year 2006, for its using to training network, while data year 2007 for testing network. The model development data were based on historical weather data for the year 2006.

We used five input variables and one output for time with variables corresponding to twenty four hours, 00:00 hrs, 03:00 hrs, 06:00 hrs, 09:00 hrs, 12:00 hrs, 15:00 hrs, 18:00 hrs and 21:00 hrs. The inputs required by the air temperature forecasting include the parameters listed in table 1.

TABLE 1. METEOROLOGICAL VRAIABLES

No.	Variable taken every 3 hour	Unit
1	Wind speed	Knots
2	Wind direction	Degree
3	Relative humidity	Rh %
4	Dew point	Deg.^{c}
5	Air Pressure	hPa

These weather parameters were selected due to their effect on air temperatures. For example the relationship between air temperature, dew point temperature and the relative humidity, when the air temperature is close to the dew point, the relative humidity is high (often 80% or greater). Relative humidity is dependent upon the temperature. So, if the temperature changes, the relative humidity will change. [11]

III. FUNCTION APPROXIMATION

The need for function approximation arises in many branches of applied mathematics and computer science in particular. In general a function approximation problem asks us to select a function among a well-defined class that closely matches a target function in specific way. One can distinguish two major classes of function approximation problems: First known target functions approximation theory is the branch of numerical analysis that investigates how certain known function can be approximated by a specific class of function or curve fitting methods. Second, the target function is unknown instead of an explicit formula where only a set of points are provided.

Depending on the structure of the domain, several techniques for approximating may be applicable. For example if an operation on the real numbers, techniques of interpolation, regression analysis, and curve fitting can be used. To some problems as regression, classification has received a unified treatment in statistical learning theory where they are viewed as supervised learning problems[2][3].

IV. RADIAL BASES FUNCTION

Radial Basis Functional Networks (RBFN) is a non-linear layered feed forward networks used as a universal approximator. The RBFN is capable of implementing arbitrary non-linear transformations of the input space. This learning is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data. This is generally known as the curve fitting approximation problem.

These RBFNs are used in a wide range of applications and are most effective in forecasting as weather forecasting, modelling, pattern recognition, and image compression and more[12][13]. RBF consists of three different layers namely input layer, hidden layer and output layer where the hidden layer is multidimensional and known as radial counters [2].

The transformation from the input space to the hidden unit space is nonlinear whereas the transformation from the hidden unit space to the output space is linear. Thus RBFN produce a linear combination of non-linear basis functions where the dimension of input matches with the dimension of each radial center.

Each hidden unit known as radial center and each center is representative of one or some of the input patterns. The network is known as 'localized receptive field network'. The problem is solved the input space is matched with these receptive fields where the inputs are clustered around the centers and the output is linear.

$$y = \sum_{i=1}^{k} \varphi_i w_i \tag{1}$$

Thus we obtain a 'smooth fit' to the desired function. The hidden units in RBFN have Gaussian activation functions as :

$$\phi_i(x) = \varphi(\|x - t_i\|) \tag{2}$$

where $\|x - t_i\|$ is the Euclidean norm function

A)Approximation by RBF

Radial basis function neural networks (RBFNN) which employ for nonlinear function approximation have two stages. The supervised and unsupervised training procedure adapted in numerous RBFNN applications usually provides satisfactory network performance.

This method allow fast training and improves convergence, the initial stage is selecting the network centres. Orthogonal least squares and input clustering are two of such methods that show considerable results of which can provide an amicable solution to problem of initial estimates of the Gaussian kernels[7]. We dispose of a set of inputs *xt* and a set of outputs *yt*. The approximation of *y*, by a RBF will be noted ŷ*t* .This approximation will be the weighted sum of *m* Gaussian kernels ϕ As shown in Fig.1

$$\hat{y}_t = \sum_{i=1}^{m} \lambda_i \phi(x_i, c_i, \sigma_i) \tag{3}$$

t=1 to N

$$\phi(x_i, C_i, \sigma_i) = \exp\left(\frac{-\|x - x_i\|}{\sqrt{2}\sigma_i}\right)^2 \quad (4)$$

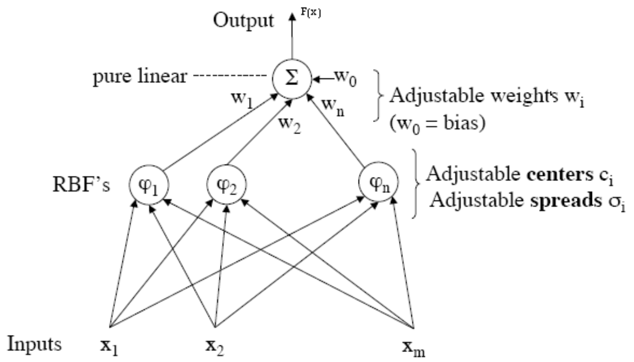


Figure 1. Standard structure of RBFANN

The complexity of a RBFN is determined by the number of Gaussian kernels. The different parameters to specify are the position of the Gaussian kernels (*Ci*), their variances σ_i. The second parameter to be chosen is the standard deviation (or width) of the different Gaussian kernels σ_i. The last parameters to determine are the multiplicative factors λ_i. When all other parameters are defined, these are determined by the solution of a system of linear equations.

If we build a linear model between the inputs and the output, the latter will be approximated by a weighted sum of the different inputs. The weights associated to each input determines the importance that this latter has on the approximation of the output.

A) Hidden layer (Radbas Layer)

The hidden layer in RBFNN is of high dimension, which serves a different purpose than in a multilayer feed forward network. Each neuron of the hidden layer represents a RBF with equal dimensions to the input data, and the number of RBFs depends on the problem to be solved. First, the radial distance d_i, between the input vector x and the center of basis function c_i is computed for each unit i in the hidden layer as

$$d_i = \|x - c_i\| \quad (5)$$

$$y = f(x) = \sum_{i=1}^{k} w_i \varphi_i(x, c_i) = \sum_{i=1}^{k} w_i \varphi(\|x - c_i\|) \quad (6)$$

Where f is a nonlinear activation function, x is the input. c_i are the RBF centers in the input vector space, Each neuron in the hidden layer has its associated center, $\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_m$ are the number of the hidden layer, W , the linear layer neurons weight vector, X, the input vector , k is the total number of hidden layer neurons and i represents the j^{th} node in the hidden layer.

B K-Means

The K-Means most common algorithm used to determine the appropriate centers of the prototype hidden neurons in RBFNs. The basic idea of this iterative clustering algorithm is to start with an initial random partition, and then assign patterns to clusters so as to reduce the squared error. The number of clusters K must be specified beforehand[14][15]. The K-means algorithm works within four steps:

- Step 1: Randomly set cluster centers centroids.
- Step 2: Generate a new partition by assigning each data vector (pattern) to its closest cluster center.
- Step 3: Compute new cluster centers as the centroid of the data vectors assigned to the considered cluster.
- Step 4: Repeat step 2 and 3 until there is no more change in cluster center.

C)Learning Strategies

The RBF networks is trained by a variety of supervised and unsupervised learning algorithms. In the initial approaches, all data samples were assigned to the hidden layer to act like a centroid. The number of hidden units was reduced by the use of k-median clustering algorithms.

The learning phase in the (RBFNN) can be divided into three steps and two phases. The three steps are

- Find the centers (unsupervised).
- Find the widths(unsupervised)
- Weight adjustment on the outer layer (supervised).

The performance of an RBFNN critically depends on the chosen centers and widths. The main objective of the first stage is to optimize the location of centers and width. After the RBF centers have been found, the width is calculated. The width represents a measure of the spread of data associated with each node. In the input space where there are few patterns, it is desirable to have hidden units with a wide area of reception while narrow area is needed for crowded portions. In the second phase, the output layer weights are adjusted in a supervised fashion using the Least Mean Square (LMS) algorithm so that the difference between the desired output and the obtained output is minimized.

IV. FORCASTING METHOD

The selection of forecasting methods depends on several factors, such as the availability of data, the accuracy required and the ease of operation. For example forecasting weather, we use historical data as the basis for estimating future outcomes. As shown in fig. 4 the algorithm can be outlined in the following steps:.

- *Read and load weather data to the system.*
- *Divide the databases in two groups, one group is training set and other group is testing set and hide the test set*

- *Find cluster centers by K-means clustering.*
- *Calculate the Centre Widths of the function.*
- *Supervised training phase.*
- *Use training set as input to the RBFNN.*
- *Test the REBNN system by using test set of databases.*

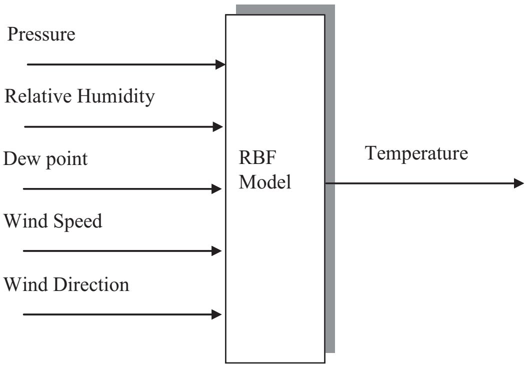
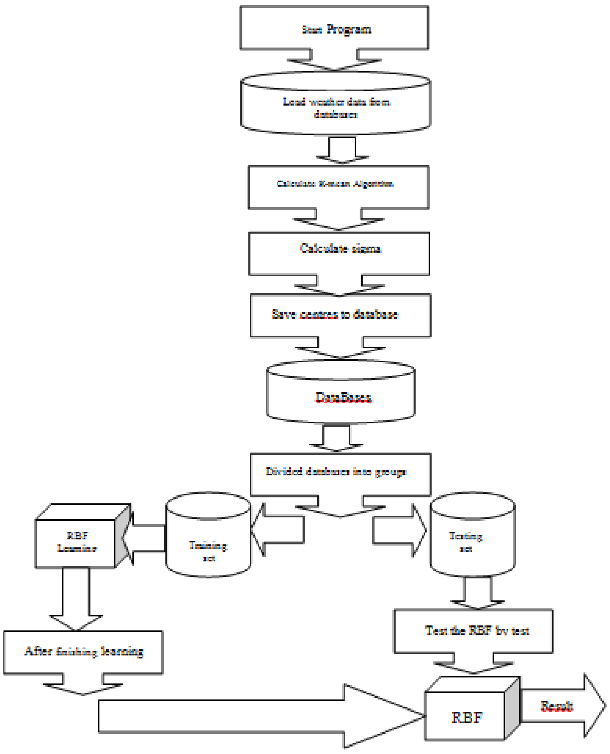


Figure 3. General structure of RBF



Figuer 4. Block diagram of the forcasting algorithm

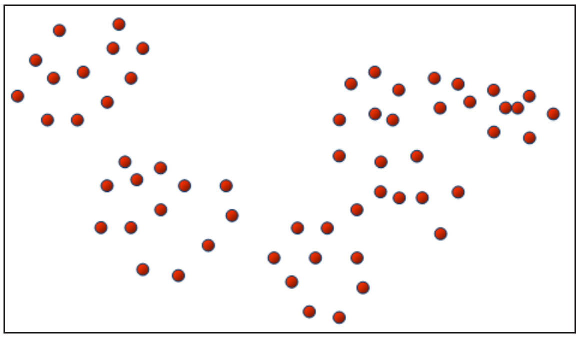
A) Data Clustering

Data clustering is the process of data in homologous aggregations, and it is branches out for searching of data. Clustering algorithm dividing the group of data into many aggregations, where the similarity between points within rhomb cluster bigger than similarity between two points within two different clusters.

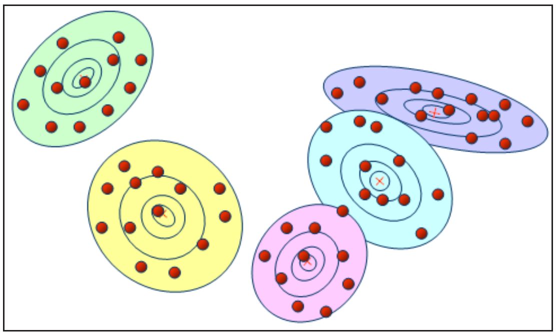
This forecasting requires calculating the distance between the temperature and all data in the database, which requires amount of time and computation, especially if we use data more than one year. To reduce the computation burden and time, we will introduce a novel approach that will reduce the required time and computation. The approach is based on using the K-means clustering algorithm [14], which to divide the feature vectors of all daily readings of the weather into several homogenous groups based on a set of features/attributes.

After dividing the database into groups, we will compute the center of each group. There are many different methods of grouping, in this research we use non-hierarchical that are moved the objects from one cluster to another cluster according to one of the criteria.

To illustrate this idea, we have a database contains 8 readings, each reading has 5 features daily, therefore we have 2920 records (objects) and 5 attributes in one year. Then we assume the number of clusters are 30 for example, after that we divide all the data into 30 groups. We will be finding each cluster containing 73 objects. Using this technique, we do not only reduce the time requirement, but also increase the research accuracy. Figure 5 (a) and (b) show the data before and after clustering



(a)



(b)

Figure 5. (a) Weather data are distributed before applying clustering algorithm.(b) after applying data clustering

The distances of each vector from the K cluster centers are computed, and the vector is assigned to the cluster having its center at a minimum distance.

V. EXPERIMENTAL RESULTS

We start with input data representing the 5 variables listed in table 1 for one year as shown in table 2. A small sample of the output value is shown in figure 6. The RBF consists of three layer, input layer, hidden layer and output layer. Number of neurons in the input layer is decided by the size of the input vector. In our case it is 5 neurons. The output layer is one neuron representing the air temperature. There is no specific rule to determine the exact number of neurons in hidden layer for the network architecture. To find out how many neurons in hidden layer that can possibly provide better performance in terms of least error of forecasting. We start from a network with one hidden layer in which there are two neurons, by increasing the number of neurons, we consider the performance error of the network. Fig. 8,9 and 10 show the difference between the predicted values and true values using different number of hidden layer neurons. Fig. 8 shows the difference using 22 neurons, fig. 9 using 49 neurons and fig. 10 using 56 neurons. As shown in fig. 11, the optimal structure is reached when the LMS is minimum when number of hidden neurons is 49. It can also be seen from fig. 11 that the error does not reduce significantly when number of neurons increases. The minimum error has been obtained for 49 neurons in the hidden layer.

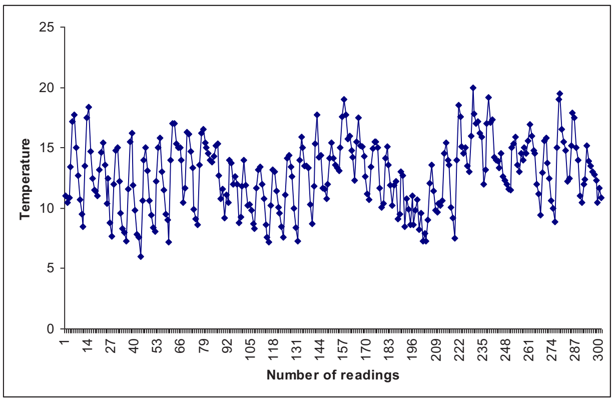


Figure 6. Sample output data

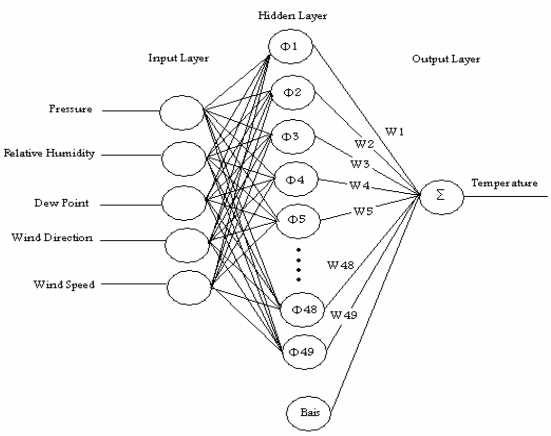


Figure 7. The optimal structure of RBF

The optimal structure for developed RBF neural network, for obtaining with minimum prediction error is shown in Table 3.

TABLE 2. input/output data for the year 2006 which is used for raining

Date	S. NO	Hour	Input (Parameters) Air P. S. level	Relativity Humidity	Dew Point Temp	Wind Direction	Wind Speed	Output Air Temp
01/01	1	00:00	23.70	84.00	8.40	230.00	3.00	11.00
	2	03:00	23.10	86.00	8.30	0.00	0.00	10.50
	3	06:00	22.70	79.00	7.40	190.00	3.00	10.90
	4	09:00	23.30	81.00	10.10	190.00	5.00	13.40
	5	12:00	20.80	64.00	10.30	200.00	10.00	17.20
	6	15:00	19.60	63.00	10.70	190.00	7.00	17.70
	7	18:00	19.50	73.00	10.30	170.00	6.00	15.00
	8	21:00	19.40	86.00	10.40	190.00	7.00	12.70
02/01	9	00:00	19.00	96.00	10.10	180.00	6.00	10.70
	10	03:00	18.10	93.00	8.50	200.00	8.00	9.50
	11	06:00	17.90	96.00	7.90	190.00	6.00	8.50
	12	09:00	18.80	77.00	9.60	210.00	9.00	13.50
	13	12:00	17.90	50.00	6.90	290.00	6.00	17.50
	14	15:00	16.70	50.00	7.90	300.00	3.00	18.40
	15	18:00	16.80	72.00	9.70	0.00	0.00	14.70
	16	21:00	17.20	85.00	10.00	0.00	0.00	12.50
03/01	17	00:00	15.70	84.00	8.90	0.00	0.00	11.50
	18	03:00	15.20	84.00	8.80	210.00	4.00	11.40
	19	06:00	14.70	83.00	8.10	230.00	4.00	11.00
	20	09:00	16.00	64.00	6.60	260.00	7.00	13.20
	21	12:00	14.00	56.00	5.80	250.00	3.00	14.60
	22	15:00	13.60	56.00	6.60	320.00	4.00	15.40
	23	18:00	14.10	76.00	9.50	310.00	11.00	13.60
	24	21:00	14.30	87.00	8.40	260.00	8.00	10.40
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
31/12	2913	00:00	35.1	80.0	12.4	30.0	6.0	15.8
	2914	03:00	35.4	93.0	12.2	0.0	0.0	13.3
	2915	06:00	35.1	93.0	12.0	0.0	0.0	13.1
	2916	09:00	38.0	72.0	12.5	60.0	10.0	17.5
	2917	12:00	37.5	62.0	10.6	30.0	8.0	18.0
	2918	15:00	37.3	69.0	11.0	20.0	10.0	16.7
	2919	18:00	38.1	66.0	10.0	50.0	8.0	16.4
	2920	21:00	38.7	68.0	10.0	40.0	6.0	15.8

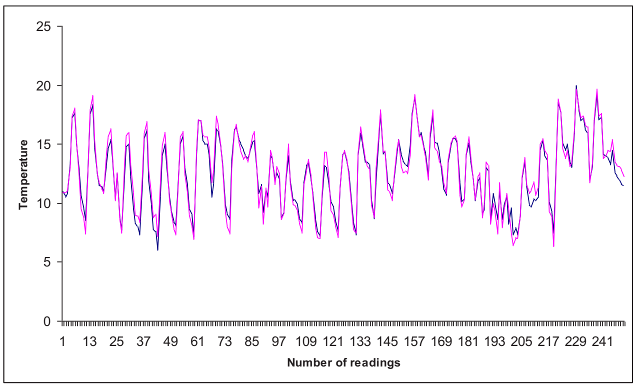


Figure 8. Comparison between observed and predicted values using 22 hidden neurons for 250 samples

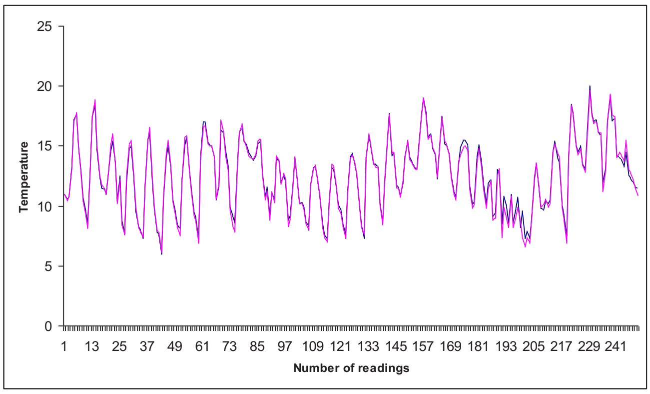


Figure 9. Comparison between observed and predicted values using 49 hidden neurons for 250 samples

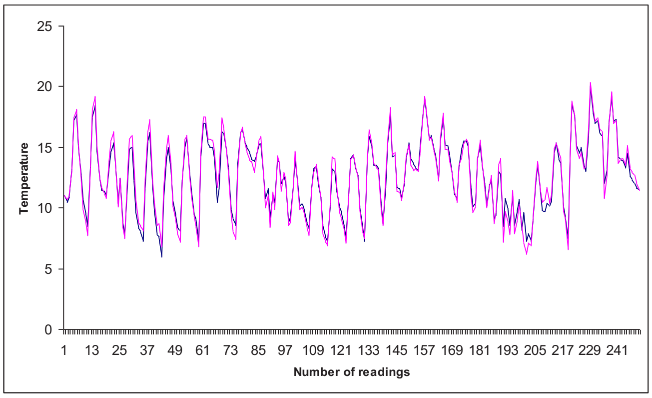


Figure 10. Comparison between observed and predicted values using 56 hidden neurons for 250 samples

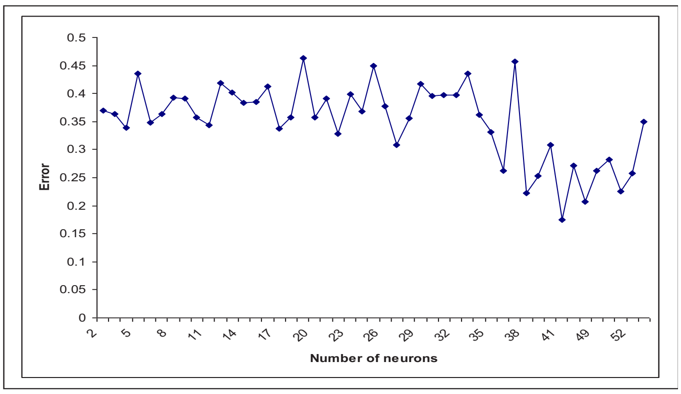


Figure.11 Change of MSE as number of neurons is increased

As can be seen from table that best results are obtained when the number of hidden neurons is 49.

TABLE 3. The optimal structure of RBF with minimum training error

Type		Number of units
Input Layer	1. Pressure	5
	2. Relative Humidity	
	3. Dew Point	
	4. Wind Direction	
	5. Wind Speed	
Hidden Layer	Guassian Function	49
Output Layer	1. Air Temperature	1

TABLE 4. Values of LMS for different number of neurons in the hidden layer

Number of neurons	Mean Square Error	Number of neurons	Mean Square Error
2	7.618259	27	0.368681
3	9.525264	28	0.449582
4	4.399589	29	0.377412
5	0.369212	30	0.30756
6	0.363826	31	0.355904
7	0.339054	32	0.417651
8	0.435077	33	0.395352
9	0.348618	34	0.396723
10	0.363548	35	0.396512
11	0.392096	36	0.435077
12	0.391086	37	0.361726
13	0.357324	38	0.330974
14	0.34359	39	0.262964
15	0.41814	40	0.456326
16	0.401203	41	0.222475
17	0.383049	47	0.253309
18	0.385165	48	0.308456
19	0.412778	49	0.174462
20	0.336892	50	0.270717
21	0.357714	51	0.206607
22	0.462961	52	0.262809
23	0.357827	53	0.281783
24	0.391772	56	0.226002
25	0.32866	57	0.257331
26	0.398196	60	0.350429

VI. CONCLUSIONS

In this paper we have presented an efficient method for the forecasting of air temperature based on the use of observations of various meteorological variables. RBF ANN was used as a function approximator to predict the air temperature. One year data was used as a training set and another year was used for testing the trained ANN. Experimental results have shown that RBF ANN can be trained effectively with smallest number of neurons without compromising the performance. The RBF can achieve good learning performance in very short training time and was also able to learn from the labeled learning pattern and generalize to the test data even though test data was hidden during training stage. Results have also shown that a good approximation with average MSE of 0.17446 using 49 neurons in the hidden layer.

REFERENCES

[1] I. Maqsood, M. R. Khan and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Comput. & Applic., Vol. 13,2004, pp. 113-122*

[2] S. Haykin. Neural Networks, A Comprehensive Foundation, New York, Macmillan Publishing., 1994

[3] C. Devi, B. Reddy, K. Kumar, B. Reddy,N. Nayak, "ANN Approach for Weather Prediction using Back Propagation," *International Journal of Engineering Trends and Technology*- Vol. No.3- 2012, pp.19-23

[4] Mohsen Hayati, and Zahra Mohebi, "Application of Artificial Neural Networks for Temperature Forecasting," *International Journal of Engineering and Applied Sciences, Vol.4 No.3, 2008,pp:164-168,*

[5] Moro QI, Alonso L, Vivaracho CE," Application of neural networks to weather forecasting with local data," *Proceedings of the 12th IASTED international conference on applied informatics*, Annecy, France, May 1994, pp 68-70

[6] G. Zhang and M. Y. Hu, "Neural Network Forecasting of the British Pound/US Dollar Exchange Rate," *OMEGA:Int. Journal of Management Science*, Vol. 26, pp: 495-506, 1998.

[7] J. Yao and C.L. Tan, "A case study on using neural networks to perform technical forecasting of forex," *Neurocomputing*, vol. 34, pp. 79-98, 2000.

[8] J. H.Wang and J. Y. Leu, "Stock Market Trend Prediction Using ARIMA-based Neural Networks," Proc. of IEEE Int. Conf. on Neural Networks, vol. 4, pp. 2160-2165, 1996.

[9] Ch.Jyosthna Devi , B.Syam Prasad Reddy, K.Vagdhan Kumar,B. Musala Reddy, N.Raja Nayak, " ANN Approach for Weather Prediction using Back Propagation," International Journal of Engineering Trends and Technology- Vol.(3). No.1, 2012

[10] A.K. Jain and R.C. Dubes, "Algorithms for Clustering Data", Prentice Hall, Englewood Cliffs, NJ, 1988

[11] McCormick-Goodhart, M.H., "The Allowable Temperature and Relative Humidity Range for the Safe Storage of Photographic Materials," Journal of the Society of Archivists, Vol. 17, No. 1, 7-21, 1996.

[12] Boopathi, G. , Arockiasamy S. " Image ompression: Wavelet transform using radial basis function (RBF) neural Network," (INDICON) pp:340-344, Kochi, India Dec 7-9, 2012

[13] E.-D. Virginia, "Biometric identification system using a radial basis network," in *Proc 34th Annu. IEEE Int.Carnahan Conf. Security Technol.*, 2000, pp. 47–51.

[14] I. Almerhag, I.El-Fegh and A. Dulla, " A Modified K-means Clustering Algorithm for Gray Image Segmentation," Proceedings of ACIT'2010, vol. 1, pp: 205-209, Benghazi, Libya,2010

[15] Shimshoni Y, Intrator N, "Classification of seismic signals by integrating ensembles of neural networks," *IEEE Trans Signal Proces, Vol.* 46 No.5, pp:1194-1201

Idirs El-Feghi Born in Misurata Libya 1959 Obtained BSc form University of Portland (1983) in Computer engineering, MSc. and PhD. from University of Windsor, Canada, (1999) and(2003) in Electrical and Computer engineering. Currently he is a professor at University of Tripoli, Tripoli-Libya. His area of interest include Image Processing, Artificial Intelligence, Optimization and Data Mining.