A New Weather Forecasting Technique using Back Propagation Neural Network with Modified Levenberg-Marquardt Algorithm for Learning

I.Kadar Shereef [1], Dr. S. Santhosh Baboo[2]

[1]Research Scholar, Dravidian University
Head, Department of Computer Applications
Sree Saraswathi Thyagaraja College,Pollachi.Coimbatore,Tamil Nadu,India.

[2]Reader, PG and Research department of Computer Science
Dwaraka Doss Goverdhan Doss Vaishnav College
Chennai,Tamil Nadu.India.

Abstract-Temperature warnings are essential forecasts since they are utilized to guard life and property. Temperature forecasting is the kind of science and technology to approximate the temperature for a future time and for a given place. Temperature forecasts are performed by means of gathering quantitative data regarding the in progress state of the atmosphere. The author in this paper utilized a neural network-based technique for determining the temperature in future. The Neural Networks package consists of various kinds of training or learning techniques. One such technique is Back Propagation Neural Network (BPN) technique. The main advantage of the Back Propagation Neural Network technique is that it can reasonably estimated a large class of functions. This technique is more efficient than numerical differentiation. The simple meaning of this term is that the proposed technique has ability to confine the complex relationships among several factors that contribute to assured temperature. The proposed idea is tested using the real time dataset. In order to further improve the prediction accuracy, this paper uses Modified Levenberg-Marquardt (LM) Algorithm for Neural Network learning. In modified LM, the learning parameters are modified. The proposed algorithm has good convergence and also it reduces the amount of oscillation in learning procedure. The proposed technique is compare with the usage of BPN and the practical working of meteorological department. The experimental result shows that the proposed technique results in better accuracy of prediction when compared to the conventional technique of weather prediction.

Keywords--- Multi Layer Perception, Temperature Forecasting, Back propagation, Artificial Neural Network, Modified Levenberg-Marquardt Algorithm

1. INTRODUCTION

THE enormous computational is necessary to resolve the equations that represents the atmosphere, error concerned in measuring the initial conditions, and an imperfect understanding of atmospheric procedures because of chaotic nature [8, 20] of the atmosphere. This indicates that forecasts turn out to be less precise as the dissimilarity in current time and the time for which the forecast is performed (the range of the forecast) increases. The use of ensembles and model helps narrow the error and pick the most likely outcome.

Various proves involved in temperature prediction are

a. Data collection(atmospheric pressure, temperature, wind speed and direction, humidity, precipitation),
b. Data assimilation and analysis
c. Numerical weather prediction
d. Model output post processing

A neural network [1] is a dominant data modeling technique that has the capability to capture and symbolize complex input /output relationships. The inspiration for the growth of neural network is obtained from the aspiration to realize an artificial system that could carry out intelligent works related to those carry out by the human brain. Neural network look like the human brain in the following manners:

a. A neural network acquires knowledge through learning
b. A neural network's knowledge is stored within interneuron connection strengths known as synaptic weights

The exact supremacy and merits of neural networks [12] occurs in the capability to symbolize both linear and non linear relationships straightforwardly from the data being modeled. Conventional linear models are simply insufficient when it approaches for true modeling data that consists of non linear features.

A neural network model is a formation that can be altered to result in a mapping from a provided set of data to characteristics of or relationships between the data. The model is modified, or trained, with the help of collection of data from a provided source as input, usually referred to as the training set. When the training phase completed successful, the neural network will be capacity to carry out classification, estimation, prediction, or simulation on new data from the same or similar sources.

An Artificial Neural Network (ANN) [2, 4, 5] is a data processing model that is motivated by the manner biological nervous systems like the brain, process those data. The main constituent of this model is the new structure of the data processing system. It consists of a large number of extremely interrelated processing elements (neurons) functioning together in order resolve particular problems. ANNs, like people, be trained by illustrations. An ANN is constructed for some application like pattern recognition or data classification, by means of a learning process. Learning in biological systems provides alterations to the synaptic relation that occurs among the neurons.

A back propagation network [9] contains at least three layers (multi layer perception):

- An input layer
- At least one intermediate hidden layer
- An output layer

In distinction to the Interactive Activation and Competition (IAC) Neural Networks and Hopfield Networks, relation weights in a back propagation network are single way. Normally, input units are linked in a feed-forward manner with input units completely linked to units in the hidden layer and hidden units completely linked to units in the output layer. An input pattern is transmitted forward to the output units by means of the intervening input-to-hidden and hidden-to-output weights when a Back Propagation network is cycled.

As the algorithm's name provides a meaning, the errors (and consequently the learning) propagate backwards from the output nodes towards the inner nodes. Therefore precisely it can be explained, back propagation is utilized to compute the gradient of the error of the network with regard to the network's adjustable weights. This gradient is forever utilized in a simple stochastic gradient descent technique to identify weights that reduces the error. Regularly the term back propagation is mentioned in a more common means in order to mention the complete process surrounding both the computation of the gradient and its utilization in stochastic gradient manner. Back propagation frequently permits fast convergence on acceptable local minima for error in the type of networks to which it is suited.

The projected Temperature Prediction System which utilizes BPN Neural Network and [13-16] modified LM algorithm [22] is evaluated with the help of the dataset from [17]. The results are contrasted with practical temperature prediction outcome [18, 19]. This system supports the meteorologist to forecast the expectation weather effortlessly and accurately.

The remainder section of this paper is organized as follows. Section 2 discusses various temperature predicting systems with various learning algorithms that were earlier proposed in literature. Section 3 explains the proposed work of developing An Efficient Temperature Prediction System using BPN Neural Network with modified LM algorithm. Section 4 illustrates the results for experiments conducted on sample dataset in evaluating the performance of the proposed system. Section 5 concludes the paper with fewer discussions.

2. Related work

Several works were performed related to the temperature prediction system and BPN network conventionally. Some of the works summarized below.

Y.Radhika *et al.*, [3] presents an application of Support Vector Machines (SVMs) for weather prediction. Time series data of every day maximum temperature at place is considered to forecast the maximum temperature of the successive day at that place according to the every day maximum temperatures for a period of earlier n days referred to as organize of the input. Significance of the system is practical for different spans of 2 to 10 days with the help of optimal values of the SVM kernel.

Mohsen Hayati *et.al,* [5] studied about Artificial Neural Network based on MLP was trained and tested using ten years (1996-2006) meteorological data. The outcome suggests that MLP network has the lesser prediction error and can be recognized as a better technique to model the short-term temperature forecasting [STTF] systems. Brian A. Smith *et.al,*[6] aims at creating a ANN models with lesser average prediction error by means of enhancing the number of distinct observations utilized in training, adding together extra input expressions that explain the date of an observation, raising the duration of prior weather data considered in all observation, and reexamining the number of hidden nodes utilized in the network. Models were generated to predict air temperature at hourly intervals from one to 12 hours before it happens. The entire ANN model, containing a network architecture and set of associated parameters, was calculated by instantiating and training 30 networks and computing the mean absolute error (MAE) of the resulting networks for few set of input patterns.

Arvind Sharma *et.al,* [7] briefly provided the way of the various connectionist models could be created with the help of various learning techniques and then examines whether they can afford the necessary level of performance, that are adequately good and robust so as to afford a reliable prediction model for stock market indices.

Mike O'Neill [11] considers two major practical concerns: the relationship among the amounts of training data and error rate (equivalent to the attempt to collect training data to create a model with provided maximum error rate) and the transferability of models' expertise among various datasets (equivalent to the helpfulness for common handwritten digit recognition).Henry A. Rowley reduces the complicated work of manually choosing nonface training illustrations, that must be preferred to period the entire space of nonface images. Simple heuristics, like utilizing the detail that faces infrequently overlie in images, can additional enhance the accuracy. Contrasting with more than a few other state-of-the-art face detection techniques, it can be observed that the proposed system has better performance by means of detection and false-positive rates.

3. ANN APPROACH

A. *Phases in Back propagation Technique*

The back propagation [10] learning technique can be separated into two phases:

- Propagation
- Weight Update

Phase 1: Propagation

Each propagation includes the following process:

1. Forward propagation of a training pattern's input is provided by means of neural network for the purpose of producing the propagation's output activations.
2. Back propagation of the output activations propagation by means of the neural network with the help of training pattern's target for the purpose creating the deltas of every output and hidden neurons.

Phase 2: Weight Update

For each weight-synapse:

1. Multiply its input activation and output delta to obtain the gradient of the weight.
2. Bring the weight in the direction of the gradient by means of adding a proportion of it from the weight.

This proportion bangs on the speed and quality of learning; it is known as learning rate. The indication of the gradient of a weight assigns where the error is increasing; this is main reason for the weight to be updated in the reverse direction.

The phase 1 and phase 2 is continual until the performance of the network is acceptable.

B. *Modes of Learning*

There are fundamentally two kinds of learning to select from, one is on-line learning and the other is batch learning. Every propagation is followed straight away by means of a weight update in online learning [21]. In batch learning, much propagation happens before weight updating carried out. Batch learning requires extra memory capacity, but on-line learning needs more updates.

C. *Algorithm*

Actual algorithm for a 3-layer network (only one hidden layer) is provided below:

1. Initialize the weights in the network (often randomly)
2. Do
3. For each example e in the training set
   O = neural-net-output (network, e) ; forward pass
   T = teacher output for e
4. Calculate error (T - O) at the output units
5. Compute delta_wh for all weights from hidden layer to output layer ; backward pass
6. Compute delta_wi for all weights from input layer to hidden layer ; backward pass continued
7. Update the weights in the network
8. Until all examples classified correctly or stopping criterion satisfied
9. Return the network

D. *Back Propagation Neural Network*

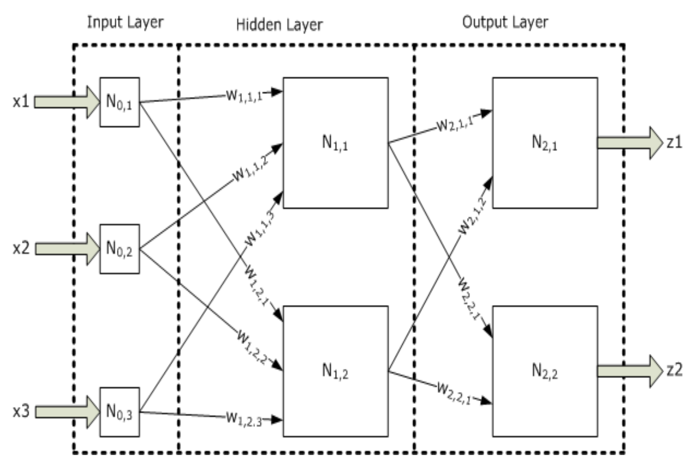

Fig1: A Back Propagation Neural Network Architecture

In the fig.1,

1. The output of a neuron in a layer migrates to every neuron in the subsequent layer.
2. All the neuron contains its own input weights.
3. The weights for the input layer are implicit to be 1 for all input. Alternatively it can be said as, input values are not altered.
4. The output of the NN is resulted by applying input values to the input layer, passing the output of all the neuron to the subsequent layer as input.
5. The Back Propagation Neural Network must have minimum of an input layer and an output layer. It is

permitted to have zero or supplementary hidden layers.

The number of neurons in the input layer is determined by the obtainable number of probable inputs. The number of neurons in the output layer is based on the number of preferred outputs. The number of hidden layers and in all the hidden layer number of neurons cannot be better defined in advance, and might alter per network arrangement and kind of data. Usually the addition of a hidden layer may permit the network to be trained more complex patterns, but at the equivalent time reduces its performance. A network configuration can have a one hidden layer, but more than one hidden layers can be provided if the network is not trained as it is expected.

The following conditions are to be analyzed for input to BPN,

- Atmospheric Pressure
- Atmospheric Temperature
- Relative Humidity
- Wind Velocity and
- Wind Direction

Back propagation is an iterative procedure that begins with the last layer and migrates backwards by the layers until the first layer is accomplished. Imagine that for every layer, the error in the output of the layer is determined previously. If the error of the output is determined, then it is not difficult to compute alterations for the weights, for the purpose of reducing that error. The difficulty is that the error in the output of the very last layer only can be observed [11].

Back propagation provides the techniques to find the error in the output of a previous layer by providing the output of a current layer as response. The process is consequently iterative: preliminary at the last layer and computing the modifications in the weight of the last layer. After that compute the error in the output of the prior layer and repeat the process.

The back propagation equations are provided below. The equation (1) represents the way to compute the partial derivative of the error E^P regarding the activation value y^i at the *n-th* layer.

Initialize the procedure by calculating the partial derivative of the error because of a single input image pattern regarding the outputs of the neurons on the last layer. The error occurred because of the single pattern is computed as below:

$$E_n^p = \frac{1}{2}\sum(x_n^i - T_n^i)^2 \qquad (1)$$

Where:

E_n^p represents the error because of a single pattern *P* at the last layer *n*; T_n^i represents the target output at the last layer (i.e., the desired output at the last layer) and x_n^i is the actual value of the output at the last layer.

Provided equation (1), then taking the partial derivative results in:

$$\frac{\partial E_n^p}{\partial x_n^i} = x_n^i - T_n^i \qquad (2)$$

Equation (2) gives us a starting value for the back propagation process. The numeric values are used for the quantities on the right side of equation (2) in order to calculate numeric values for the derivative. Using the numeric values of the derivative, the numeric values for the changes in the weights are calculated, by applying the following two equations (3) and then (4):

$$\frac{\partial E_n^p}{\partial y_n^i} = G(x_n^i)\frac{\partial E_n^p}{\partial x_n^i} \qquad (3)$$

where $G(x_n^i)$ is the derivative of the activation function.

$$\frac{\partial E_n^p}{\partial w_n^{ij}} = x_{n-1}^j \frac{\partial E_n^p}{\partial y_n^i} \qquad (4)$$

Subsequently, using equation (2) once more and also equation (3), the error for the previous layer is computed, with the help of following equation:

$$\frac{\partial E_{n-1}^p}{\partial x_{n-1}^k} = \sum_i w_n^{ik} \frac{\partial E_n^p}{\partial y_n^i} \qquad (5)$$

The values resulted from equation (5) are utilized as starting values for the computation on the directly preceding layer. This is the single most significant point in understanding back propagation. Otherwise it can be said that, it is taken the numeric values resulted from equation (5), and utilize them in a repetition of equations (3), (4) and (5) for the instantly preceding layer.

Simultaneously, the values resulted from equation (4) suggests the range to alter the weights in the current layer n, that was the entire reason of this gigantic exercise. Especially, the value of each weight is updated based on the following equation:

$$(w_n^{ij})_{new} = (w_n^{ij})_{old} - eta.(\frac{\partial E_n^p}{\partial w_n^{ij}}) \qquad (6)$$

where *eta* represents the learning rate, characteristically a small number like 0.0005 and will be decreased steadily during training.

The learning can be enhanced to improve the performance of prediction system. For this reason, this paper uses Modified Levenberg-Marquardt algorithm for learning phase of BPN

E. Modified Levenberg-Marquardt algorithm

A Modified Levenberg-Marquardt algorithm is used for training the neural network.

Considering performance index is $F(w) = e^T e$ using the Newton method we have as:

$$W_{K+1} = W_K - A_K^{-1}.g_K \tag{7}$$

$$A_k = \nabla^2 F(w)|_{w=w_k} \tag{8}$$

$$g_k = \nabla F(w)|_{w=w_k} \tag{9}$$

$$[\nabla F(w)]_j = \frac{\partial F(w)}{\partial w_j} = 2\sum_{i=1}^{N} e_i(w).\frac{\partial e_i(w)}{\partial w_j} \tag{10}$$

The gradient can write as:

$$\nabla F(x) = 2J^T e(w) \tag{11}$$

Where

$$J(w) = \begin{bmatrix} \frac{\partial e_{11}}{\partial w_1} & \frac{\partial e_{11}}{\partial w_2} & \cdots & \frac{\partial e_{11}}{\partial w_N} \\ \frac{\partial e_{21}}{\partial w_1} & \frac{\partial e_{21}}{\partial w_2} & \cdots & \frac{\partial e_{21}}{\partial w_N} \\ & . & & \\ & . & & \\ & . & & \\ \frac{\partial e_{KP}}{\partial w_1} & \frac{\partial e_{KP}}{\partial w_2} & \cdots & \frac{\partial e_{KP}}{\partial w_N} \end{bmatrix} \tag{12}$$

$J(w)$ is called the Jacobian matrix.

Next we want to find the Hessian matrix. The k, j elements of the Hessian matrix yields as:

$$\begin{aligned} [\nabla^2 F(w)]_{k,j} &= \frac{\partial^2 F(w)}{\partial w_k \partial w_j} \\ &= 2\sum_{i=1}^{N} \left\{ \frac{\partial e_i(w)}{\partial w_k}\frac{\partial e_i(w)}{\partial w_j} \right. \\ &\quad \left. + e_i(w).\frac{\partial^2 e_i(w)}{\partial w_k \partial w_j} \right\} \end{aligned} \tag{13}$$

The Hessian matrix can then be expressed as follows:

$$\nabla^2 F(w) = 2J^T(W).J(W) + S(W) \tag{14}$$

$$S(w) = \sum_{i=1}^{N} e_i(w).\nabla^2 e_i(w) \tag{15}$$

If S(w) is small assumed, the Hessian matrix can be approximated as:

$$\nabla^2 F(w) \cong 2J^T(w)J(w) \tag{16}$$

Using (7) and (15) we obtain the Gauss-Newton method as:

$$\begin{aligned} W_{k+1} &= \\ W_k &- [2J^T(w_k)\cdot J(w_k)]^{-1} 2J^T(w_k)e(w_k) \\ &\cong W_k - [J^T(w_k)\cdot J(w_k)]^{-1} J^T(w_k)e(w_k) \end{aligned} \tag{17}$$

The advantage of Gauss-Newton is that it does not require calculation of second derivatives.

There is a problem the Gauss-Newton method is the matrix $H = J^T J$ may not be invertible. This can be overcome by using the following modification.

Hessian matrix can be written as:

$$G = H + \mu I \tag{18}$$

Suppose that the eigenvalues and eigenvectors of H are $\{\lambda_1, \lambda_2, \ldots \ldots, \lambda_n\}$ and $\{z_1, z_2, \ldots \ldots, z_n\}$.Then:

$$\begin{aligned} Gz_i &= [H + \mu I]z_i \\ &= Hz_i + \mu z_i \\ &= \lambda_i z_i + \mu z_i \\ &= (\lambda_i + \mu)z_i \end{aligned} \tag{19}$$

Therefore the eigenvectors of G are the same as the eigenvectors of H, and the eigen values of G are $(\lambda_i + \mu)$. The matrix G is positive definite by increasing μ until $(\lambda_i + \mu) > 0$ for all i therefore the matrix will be invertible.

This leads to Levenberg-Marquardt algorithm:

$$w_{k+1} = w_k - [J^T(w_k)J(w_k) + \mu I]^{-1} J^T(w_k) e(w_k) \quad (20)$$

$$\Delta w_k = [J^T(w_k)J(w_k) + \mu I]^{-1} J^T(w_k) e(w_k) \quad (21)$$

As known, learning parameter, μ is illustrator of steps of actual output movement to desired output. In the standard LM method, μ is a constant number. This paper modifies LM method using μ as:

$$\mu = 0.01 e^T e \quad (22)$$

Where e is a $k \times 1$ matrix therefore $e^T e$ is a 1×1 therefore $[J^T J + \mu I]$ is invertible.

Therefore, if actual output is far than desired output or similarly, errors are large so, it converges to desired output with large steps. Likewise, when measurement of error is small then, actual output approaches to desired output with soft steps. Therefore error oscillation reduces greatly.

4. Experimentation And Result

To experiment the proposed system a Madras Minambak, India (VOMM)[17] contains the real time observation of the weather for a particular period of time. For this experiment, an observation of 2010 year is taken. The dataset contains many attributes such as Temperature, Dew Point, Relative Humidity (RH), Wind Direction (DIR), Wind Speed (SPD) and Visibility (VIS).

4.1 Experimental Setup

TABLE 4.1

BPN with Modified LM	Seasons			
	Winter	Pre-Monsoon	South-West Monsoon	North-East Monsoon
Number of Hidden Neuron	6	6	6	6
Number of Epochs	150	150	150	150
Activation Function Used in Hidden Layer	Tan-sig	Tan-sig	Tan-sig	Tan-sig
Activation Function Used Output Layer	pure linear	pure linear	pure linear	pure linear

The experimental set up for this paper considers four seasonal variations. The available weather data were split into four seasons such as Winter (January-February), Pre-Monsoon (March-May), South-West Monsoon (June-September) and North-East Monsoon (October-December). This data is obtained from Indian Meteorological Department (IMD) [23]. In this experimental process, the missing values are obtained by the k-Nearest Neighbor algorithm.

Table 4.1 shows the various variables and parameters used for the BPN with Modified LM approach. The number of hidden neurons used in the present experimental observation is 6. Moreover, the number of iterations (epochs) taken is 150. The activation function used in Hidden and Output layer is Tan-sig and pure linear respectively for all the seasons considered.

Ten random days in each season are selected as unseen days. For Winter season, the unseen days chosen are 1/1/10, 2/1/10, 4/1/10, 18/1/10, 16/2/10, 20/2/10, 21/2/10, 23/2/10, 25/2/10 and 28/2/10. For Pre-Monsoon season, the unseen days chosen are 5/3/10, 8/3/10, 14/3/10, 27/3/10, 5/4/10, 10/4/10, 15/4/10, 18/5/10, 28/5/10 and 29/5/10.

For South-West Monsoon, the unseen days chosen are 6/6/10, 23/6/10, 29/6/10, 7/7/10, 19/7/10, 1/8/10, 20/8/10, 28/8/10, 2/9/10 and 27/9/10.For North-East Monsoon, the unseen days chosen are 1/10/10, 8/10/10, 28/10/10, 2/11/10, 15/11/10, 23/11/10, 29/11/10, 3/12/10, 14/12/10 and 25/12/10.

4.2 Performance Parameters

The performance of the proposed approaches are evaluated using the following parameters like

- Mean Squared Error (MSE)
- Minimum and Maximum Error and
- Prediction Accuracy

A. Mean Squared Error (MSE)

Table 4.2 shows the Mean Squared Error (MSE) comparison of the proposed approach and the existing approaches. The comparison is obtained for four seasons namely Winter, Pre-Monsoon, South-West Monsoon and North-East Monsoon.

Table 4.2
Mean Squared Error Comparison

Seasons	Mean Squared Error (Iterations =150)		
	BPN with Linear Learning	BPN with LM	BPN with Modified LM
Winter	1.3	0.84	0.083
Pre-Monsoon	1.21	0.75	0.071
South-West Monsoon	1.13	0.69	0.063
North- East Monsoon	1.45	0.89	0.098

For the South-West Monsoon season, the MSE obtained for the proposed BPN with LM approach is 0.063 which is very less than the MSE obtained by the existing approaches like BPN with LM and BPN with Linear Learning. South-West Monsoon season has the least MSE value.

The minimum and maximum error taken for four seasons are obtained and tabulated below table 4.3 , 4.4 and shows the minimum and maximum error comparison of the BPN approaches with various learning techniques.

Table 4.3: The Minimum error Comparison for four seasons

Seasons	Minimum Error		
	BPN with Linear Learning	BPN with LM	BPN with Modified LM
Winter	0.1025	0.0791	0.0093
Pre-Monsoon	0.0972	0.0725	0.0089
South-West Monsoon	0. 082	0.0611	0.0081
North-East Monsoon	0.1110	0.0897	0.0097

The minimum error obtained by the existing approaches such as BPN with Linear Learning and BPN with LM is higher when compared to the proposed BPN with Modified LM approach for all the seasons.

Table 4.4: The Maximum error Comparison for four seasons

Seasons	Maximum Error		
	BPN with Linear Learning	BPN with LM	BPN with Modified LM
Winter	2.292	1.465	0.6012
Pre-Monsoon	2.178	1.324	0.5712
South-West Monsoon	2.035	1.212	0.5392
North-East Monsoon	2.214	1.428	0. 6315

C. Prediction Accuracy

Prediction accuracy for the proposed approaches for each season is tabulated in Table 4.5.

When the prediction accuracy of the existing approach is compared with the proposed approach, the proposed BPN with Modified LM is observed to have higher prediction accuracy for seasons taken into consideration.

Table 4.5 Comparison of the Prediction Accuracy for Various Seasons

Seasons	Prediction Accuracy (%)		
	BPN with Linear Learning	BPN with LM	BPN with Modified LM
Winter	88.14	91.80	93.89
Pre-Monsoon	88.10	92.12	94.28
South-West Monsoon	88.97	92.61	94.87
North-East Monsoon	87.62	91.62	93.39

5. CONCLUSION

In this paper, back propagation neural network is used for predicting the temperature based on the training set provided to the neural network. Through the implementation of this system, it is illustrated, how an intelligent system can be efficiently integrated with a neural network prediction model to predict the temperature. This algorithm improves convergence and damps the oscillations. This method proves to be a simplified conjugate gradient method. When incorporated into the software tool the performance of the back propagation neural network was satisfactory as there were not substantial number of errors in categorizing. Back propagation neural network approach for temperature forecasting is capable of yielding good results and can be considered as an alternative to traditional meteorological approaches. This paper uses Modified Levenberg-Marquardt Algorithm for Learning. This approach is able to determine the non-linear relationship that exists between the historical data (temperature, wind speed, humidity, etc.,) supplied to the system during the training phase and on that basis, make a prediction of what the temperature would be in future.The proposed approach is evaluated on Madras Minambak, India (VOMM) dataset. The performance of the proposed approach is evaluated based on the parameters like Mean Squared Error, Minimum and Maximum Error and Prediction Accuracy. The results are obtained and the values are tabulated for the data set. The performance of the proposed approach outperforms the existing two approaches based on the results obtained.

REFERENCES

[1] Xinghuo Yu, M. Onder Efe, and Okyay Kaynak,” A General Back propagation Algorithm for Feedforward Neural Networks Learning,”

[2] R. Rojas: Neural Networks, Springer-Verlag, Berlin, 1996.

[3] Y.Radhika and M.Shashi,” Atmospheric Temperature Prediction using Support Vector Machines,” International Journal of Computer Theory and Engineering, Vol. 1, No. 1, April 2009 1793-8201.

[4] M. Ali Akcayol, Can Cinar, Artificial neural network based modeling of heated catalytic converter performance, Applied Thermal Engineering 25 (2005) 2341-2350.

[5] Mohsen Hayati, and Zahra Mohebi,” Application of Artificial Neural Networks for Temperature Forecasting,” World Academy of Science, Engineering and Technology 28 2007.

[6] Brian A. Smith, Ronald W. McClendon, and Gerrit Hoogenboom,”

Improving Air Temperature Prediction with Artificial Neural Networks" International Journal of Computational Intelligence 3;3 2007.
[7] Arvind Sharma, Prof. Manish Manoria," A Weather Forecasting System using concept of Soft Computing," pp.12-20 (2006)
[8] Ajith Abraham1, Ninan Sajith Philip2, Baikunth Nath3, P. Saratchandran4," Performance Analysis of Connectionist Paradigms for Modeling Chaotic Behavior of Stock Indices,"
[9] Surajit Chattopadhyay," Multilayered feed forward Artificial Neural Network model to predict the average summer-monsoon rainfall in India ," 2006
[10] Raúl Rojas," The back propagation algorithm of Neural Networks - A Systematic Introduction, "chapter 7, ISBN 978-3540605058
[11] Mike O'Neill," Neural Network for Recognition of Handwritten Digits," Standard Reference Data Program National Institute of Standards and Technology.
[12] Carpenter, G. and Grossberg, S. (1998) in Adaptive Resonance Theory (ART), The Handbook of Brain Theory and Neural Networks, (ed. M.A. Arbib), MIT Press, Cambridge, MA, (pp. 79–82).
[13] Ping Chang and Jeng-Shong Shih," The Application of Back Propagation Neural Network of Multi-channel Piezoelectric Quartz Crystal Sensor for Mixed Organic Vapours," Tamkang Journal of Science and Engineering, Vol. 5, No. 4, pp. 209-217 (2002).
[14] S. Anna Durai, and E. Anna Saro," Image Compression with Back-Propagation Neural Network using Cumulative Distribution Function," World Academy of Science, Engineering and Technology 17 2006.
[15] Mark Pethick, Michael Liddle, Paul Werstein, and Zhiyi Huang," Parallelization of a Backpropagation Neural Network on a Cluster Computer,"
[16] K.M. Neaupane, S.H. Achet," Use of backpropagation neural network for landslide monitoring," Engineering Geology 74 (2004) 213–226.
[17] http://weather.uwyo.edu/cgi-bin/wyowx.fcgi?TYPE=sflist&DATE=current&HOUR=current&UNITS=A&STATION=VOMM
[18] Grossberg, S ,"Adaptive Pattern Classification and Universal Recoding: Parallel Development and Coding of Neural Feature Detectors", Biological Cybernetics, 23, 121–134 (1976).
[19] Maurizio Bevilacqua, "Failure rate prediction with artificial neural networks," Journal of Quality in Maintenance Engineering Vol. 11 No. 3, 2005 pp. 279-294Emerald Group Publishing Limited 1355-2511
[20] Chowdhury A and Mhasawade S V (1991), "Variations in Meteorological Floods during Summer Monsoon Over India", Mausam, 42, 2, pp. 167-170.
[21] Gowri T. M. and Reddy V.V.C. 2008. Load Forecasting by a Novel Technique using ANN. ARPN Journal of Engineering and Applied Sciences. 3(2): 19-25.
[22] Amir Abolfazl Suratgar, Mohammad Bagher Tavakoli and Abbas Hoseinabadi, "Modified Levenberg-Marquardt Method for Neural Networks Training", World Academy of Science, Engineering and Technology, Pp. 46-48, 2005.
[23] http//: www.mausam.gov.in/WEBIMD/downloads/termglossory.pdf).

Lt. Dr. S. Santhosh Baboo, aged forty two, has around Nineteen years of postgraduate teaching experience in Computer Science, which includes Six years of administrative experience. He is a member, board of studies, in several autonomous colleges, and designs the curriculum of undergraduate and postgraduate programmes. He is a consultant for starting new courses, setting up computer labs, and recruiting lecturers for many colleges. Equipped with a Masters degree in Computer Science and a Doctorate in Computer Science, he is a visiting faculty to IT companies. It is customary to see him at several national/international conferences and training programmes, both as a participant and as a resource person. He has been keenly involved in organizing training programmes for students and faculty members. His good rapport with the IT companies has been instrumental in on/off campus interviews, and has helped the post graduate students to get real time projects. He has also guided many such live projects. Lt. Dr. Santhosh Baboo has authored a commendable number of research papers in international/national Conference/journals and also guides research scholars in Computer Science. Currently he is Senior Lecturer in the Postgraduate and Research department of Computer Science at Dwaraka Doss Goverdhan Doss Vaishnav College (accredited at 'A' grade by NAAC), one of the premier institutions in Chennai.

I.Kadar Shereef, done his Under-Graduation(B.Sc., Mathematics) in NGM College, Post-Graduation (MCA) in Trichy Jamal Mohamed College and Master of Philosophy Degree in Periyar University (distance education). He is currently pursuing his Ph.D., in Computer Science in Dravidian University, Kuppam, Andhra Pradesh. Also, he is working as a Lecturer, Department of BCA, Sree Saraswathi Thyagaraja College of Arts and Science, Pollachi. He is having more than two years of research experience and more than 6 years of teaching experience. His research interest includes Data mining, Climate Prediction, Neural Network and Soft Computing.