

An Efficient Temperature Prediction System using BPN Neural Network

Dr. S. Santhosh Baboo and I.Kadar Shereef

Abstract-Temperature warnings are important forecasts because they are used to protect life and property. Temperature forecasting is the application of science and technology to predict the state of the temperature for a future time and a given location. Temperature forecasts are made by collecting quantitative data about the current state of the atmosphere. In this paper, we present a neural network-based algorithm for predicting the temperature. The Neural Networks package supports different types of training or learning algorithms. One such algorithm is Back Propagation Neural Network (BPN) technique. The main advantage of the BPN neural network method is that it can fairly approximate a large class of functions. This method is more efficient than numerical differentiation. The simple meaning of this term is that our model has potential to capture the complex relationships between many factors that contribute to certain temperature. The results are compared with actual working of meteorological department and these results confirm that our model have the potential for successful application to temperature forecasting. Real time processing of weather data indicate that the BPN based weather forecast have shown improvement not only over guidance forecasts from numerical models, but over official local weather service forecasts as well.

Index Terms-- Multi layer perception, Temperature forecasting, Back propagation, Artificial Neural Network

I. INTRODUCTION

Due to chaotic nature[8] of the atmosphere, the massive computational power is required to solve the equations that describe the atmosphere, error involved in measuring the initial conditions, and an incomplete understanding of atmospheric processes. This means that forecasts become less accurate as the difference in current time and the time for which the forecast is being made (the range of the forecast) increases. The use of ensembles and model helps narrow the error and pick the most likely outcome.

Several steps to predict the temperature 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 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 network resemble the human brain in the following two ways:

- 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 true power and advantages of neural networks lies in the ability to represent both linear and non linear relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes true modeling data that contains non linear characteristics.

A neural network model is a structure that can be adjusted to produce a mapping from a given set of data to features of or relationships among the data. The model is adjusted, or trained, using a collection of data from a given source as input, typically referred to as the training set. After successful training, the neural network will be able to perform classification, estimation, prediction, or simulation on new data from the same or similar sources.

An Artificial Neural Network (ANN)[5] is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons.

A back propagation network [9] consists of at least three layers (multi layer perception): an input layer, at least one intermediate hidden layer, and an output layer. In contrast to the Interactive Activation and Competition (IAC) neural networks (IAC) and Hopfield networks, connection weights in a back propagation network are one way. Typically, units are connected in a feed-forward fashion with input units fully connected to units in the hidden layer and hidden units fully connected to units in the output layer. When a Back Propagation network is cycled, an input pattern is propagated forward to the output units through the intervening input-to-hidden and hidden-to-output weights.

As the algorithm's name implies, the errors (and therefore the learning) propagate backwards from the output nodes to the inner nodes. So technically speaking, back propagation is used to calculate the gradient of the error of the network with respect to the network's modifiable weights. This gradient is always used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Often the term "back propagation" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient

descent. Back propagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited.

II. RELATED WORK

Many works were done related to the temperature prediction system and BPN network. They are summarized below.

Y.Radhika and M.Shashi [3] presents an application of Support Vector Machines (SVMs) for weather prediction. Time series data of daily maximum temperature at location is analyzed to predict the maximum temperature of the next day at that location based on the daily maximum temperatures for a span of previous n days referred to as order of the input. Performance of the system is observed over various spans of 2 to 10 days by using optimal values of the 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 results show that MLP network has the minimum forecasting error and can be considered as a good method to model the short-term temperature forecasting [STTF] systems. Brian A. Smith *et.al*, [6] focused on developing ANN models with reduced average prediction error by increasing the number of distinct observations used in training, adding additional input terms that describe the date of an observation, increasing the duration of prior weather data included in each observation, and reexamining the number of hidden nodes used in the network. Models were created to predict air temperature at hourly intervals from one to 12 hours ahead. Each ANN model, consisting of a network architecture and set of associated parameters, was evaluated by instantiating and training 30 networks and calculating the mean absolute error (MAE) of the resulting networks for some set of input patterns.

Arvind Sharma *et.al*, [7] briefly explains how the different connectionist paradigms could be formulated using different learning methods and then investigates whether they can provide the required level of performance, which are sufficiently good and robust so as to provide a reliable forecast model for stock market indices. Experiment results reveal that all the connectionist paradigms considered could represent the stock indices behavior very accurately.

Mike O'Neill [11] focus on two major practical considerations: the relationship between the amounts of training data and error rate (corresponding to the effort to collect training data to build a model with given maximum error rate) and the transferability of models' expertise between different datasets (corresponding to the usefulness for general handwritten digit recognition).

Henry A. Rowley eliminates the difficult task of manually selecting nonface training examples, which must be chosen to span the entire space of nonface images. Simple heuristics,

such as using the fact that faces rarely overlap in images, can further improve the accuracy. Comparisons with several other state-of-the-art face detection systems are presented; showing that our system has comparable performance in terms of detection and false-positive rates.

III. ANN APPROACH

A. Phases in Backpropagation Technique

The back propagation [10] learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

Each propagation involves the following steps:

1. Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.

2. Back propagation of the propagation's output activations through the neural network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight Update

For each weight-synapse:

1. Multiply its output delta and input activation to get the gradient of the weight.

2. Bring the weight in the direction of the gradient by adding a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the learning rate. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction.

Repeat the phase 1 and 2 until the performance of the network is satisfactory.

B. Modes of Learning

There are basically two modes of learning to choose from, one is on-line learning and the other is batch learning. In on-line learning, each propagation is followed immediately by a weight update. In batch learning, much propagation occurs before weight updating occurs. Batch learning requires more memory capacity, but on-line learning requires more updates.

C. Algorithm

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

```

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

D. Back Propagation Neural Network

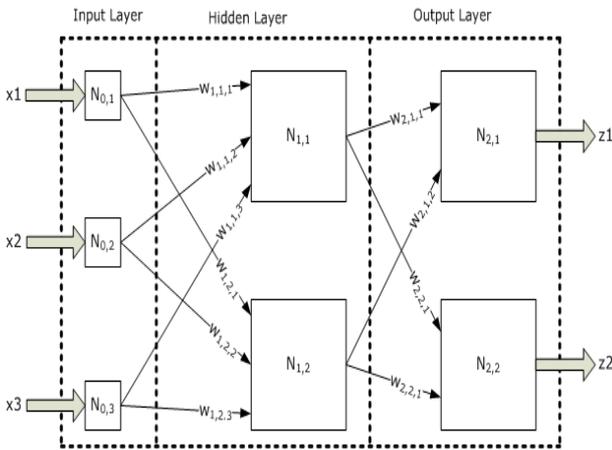


Fig1: A Back Propagation Neural Network

In the fig.1,

1. The output of a neuron in a layer goes to all neurons in the following layer.
2. Each neuron has its own input weights.
3. The weights for the input layer are assumed to be 1 for each input. In other words, input values are not changed.
4. The output of the NN is reached by applying input values to the input layer, passing the output of each neuron to the following layer as input.
5. The Back Propagation NN must have at least an input layer and an output layer. It could have zero or more hidden layers.

The number of neurons in the input layer depends on the number of possible inputs we have, while the number of neurons in the output layer depends on the number of desired outputs. The number of hidden layers and how many neurons in each hidden layer cannot be well defined in advance, and could change per network configuration and type of data. In general the addition of a hidden layer could allow the network to learn more complex patterns, but at the same time decreases its performance. A network

configuration can have a single hidden layer, but more hidden layers can be added if the network is not learning well.

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

- (i) Atmospheric Pressure
- (ii) Atmospheric Temperature
- (iii) Relative Humidity
- (iv) Wind Velocity and
- (v) Wind Direction

Back propagation is an iterative process that starts with the last layer and moves backwards through the layers until the first layer is reached. Assume that for each layer, we know the error in the output of the layer. If we know the error of the output, then it is not hard to calculate changes for the weights, so as to reduce that error. The problem is that we can only observe the error in the output of the very last layer [11].

Back propagation gives us a way to determine the error in the output of a prior layer by giving the output of a current layer as feedback. The process is therefore iterative: starting at the last layer and calculating the changes in the weight of the last layer. Then calculate the error in the output of the prior layer. Repeat.

The back propagation equations are given below. The first equation shows how to calculate the partial derivative of the error E^p with respect to the activation value y^i at the n -th layer. In the code, you will see a variable named **dErr_wrt_dYn[ii]**.

Start the process off by computing the partial derivative of the error due to a single input image pattern with respect to the outputs of the neurons on the last layer. The error due to a single pattern is calculated as follows:

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

where :

E_n^p is the error due to a single pattern P at the last layer n ;
 T_n^i is 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.

Given equation (1), then taking the partial derivative yields:

$$\frac{\partial E_n^p}{\partial x_n^i} = x_n^i - T_n^i \tag{2}$$

Equation (2) gives us a starting value for the back propagation process. We use the numeric values 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, we calculate the numeric values for the changes in the weights, by applying the following two equations (3) and then (4):

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

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

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

Then, using equation (2) again and also equation (3), we calculate the error for the previous layer, using the following equation (5):

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

The values we obtain from equation (5) are used as starting values for the calculations on the immediately preceding layer. This is the single most important point in understanding back propagation. In other words, we take the numeric values obtained from equation (5), and use them in a repetition of equations (3), (4) and (5) for the immediately preceding layer.

Meanwhile, the values from equation (4) tell us how much to change the weights in the current layer n, which was the whole purpose of this gigantic exercise. In particular, we update the value of each weight according to the formula:

$$(w_n^{ij})_{new} = (w_n^{ij})_{old} - eta \cdot \left(\frac{\partial E_n^p}{\partial w_n^{ij}} \right) \quad (6)$$

where *eta* is the "learning rate", typically a small number like 0.0005 and will be decreased gradually during training.

IV. EXPERIMENTATION AND RESULT

From conducted experiments we find the following key changes in the atmospheric pressure signature that can be related to dynamic states of atmospheric conditions and for meaningful short duration weather prediction.

Fig.2 shows the graph for Indian climate in different time interval. We tested our algorithm by using those data as training set and obtained the related temperature. The training vs error graph is shown in fig.3.

From the conducted experiment we can examine that the temperature peaks during the month of May, its range is 41 degree centigrade. The temperature falls to the dead end during the month of January. The minimum temperature is 6 °C. The wet days occur during the month of November. The speed of the wind remains constant with slight changes throughout the annum. The relative humidity is exceeding 43°C during the month of August. The humidity remains constant in the month of March and April at the same time its the least scale of humidity.

The least average temperature occurs during the month of December and its peek occurs in the month of June and July. The average sunlight lies in the range of 5-10 °C throughout the entire year. The precipitation is maximum (20 cm) during the month of mid August.

The back propagation Training-Error graph explains that the error is high when the iteration is less and vice versa. In the above graph it explains that when the iteration count is below 1000 the sum squared error is maximum (i.e. 0.26) and when the count reaches 5000 the error value is merely 0.

This results that for more accurate results, the iteration count should be high. The most useful measure of model performance, however, comes from evaluating the sequence of 12 predictions leading to severe freeze events such as those in February and March.

The present considerations on the meteorological time series leads to two possible interpretations of the global climate: one is that a low-dimensional climate attractor may exist and that the climate dynamics may have altered. The other temperature variations may be colored noise with many degrees of freedom. The latter interpretation would lead to the following forecast of the future trend in the climate: the global temperature would begin to decrease at some point in the future, since colored noise has a zero mean in a long time period. Within the framework of the present study we can dismiss neither of the interpretations. The present work is still in a preliminary stage.

The distribution of prediction errors across all horizons is centered near zero, while the variance of these error distributions increases relative to horizon length. The increased divergence between predicted and observed temperatures at longer horizons is apparent in the plots. As prediction horizon increases, so does deviation from the line of perfect fit. The trend holds, specifically, in cases where a model fails to predict freezing temperatures.

At the other extreme, the use of a logistic activation function in the output node, and the inverse of the scaling function to convert the output to a temperature, placed an upper bound on the model predictions. Because the scaling range was smaller than the output range of a logistic node, this bound was several degrees higher than the 20 °C threshold used to select observations for the development and evaluation sets. As a result, models were constrained from predicting temperatures above 25 °C. At temperatures near 25 °C, models are more likely to under predict. As the prediction horizon increases, the number of observed temperatures above this threshold increases.

Another experiment was done to see performance of overall system. Artificial neural network was trained by using 200 training data. Training process results the weight and bias of

the network. This indicates inverse kinematic model of robot manipulator. For testing, the network was tested by using data

that are not used for training process. And experimental result

shows that artificial neural network can model the inverse kinematic of robot manipulator with average error of 2.16%.

The performance of artificial neural network is indicated by the RMSE value. Experiment was done in various training parameter value of artificial neural network, i.e., various numbers of hidden layers, various number of neuron per layer, and various value of learning rate. There were 200 training data used in this research for training the artificial neural network. The training data were created by running the robot manually and measuring the position of each link directly.

The success of the 24-hour model makes intuitive sense as such a history is capable of generalizing over trends associated with the familiar daily cycle. The decision in previous research to use six hours of prior data was likely

due to the method of increasing the duration by short increments until evaluation errors began to increase. Because that work relied on single-network evaluations and found that a network with eight hours of prior information

was less accurate than a six-hour network, the experiment was stopped before exploring longer durations of prior data. The results of the research reported herein suggest that the use of multiple network evaluation can avoid such errors.

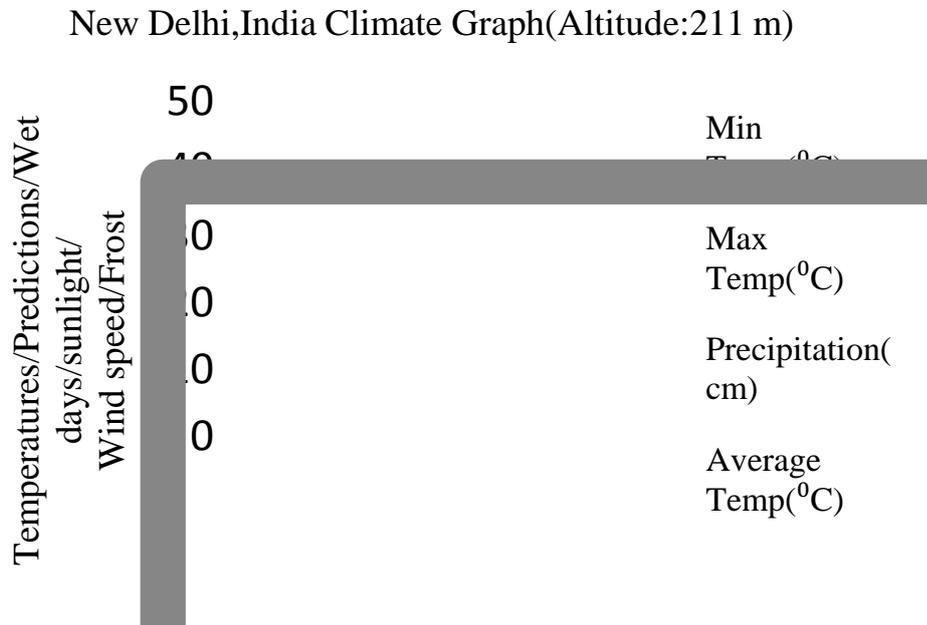


Fig.2: India Climate Graph

TABLE 1: INDIA TEMPERATURES IN FAHRENHEIT

| Temperatures in India (by month) | Average min. Temp in India(°F) | Average max Temp in India(°F) | Average Temperatures(°F) |
|----------------------------------|--------------------------------|-------------------------------|--------------------------|
| January | 45 | 70 | 57 |
| February | 50 | 75 | 63 |
| March | 59 | 86 | 73 |
| April | 70 | 97 | 83 |
| May | 81 | 106 | 93 |
| June | 84 | 104 | 94 |
| July | 81 | 95 | 94 |

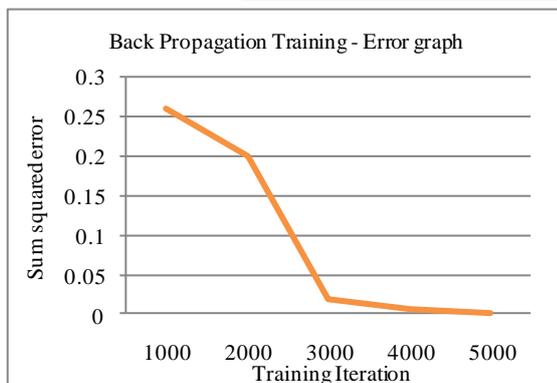


Fig.3: Back propagation Training-Error graph

V. 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, we illustrate 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 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.

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] An Introduction to Back-Propagation Neural Networks by Pete McCollum
- [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,"
- [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,"
- [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] <http://www.climatetemp.info/india/>
- [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.



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 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, and 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 one year of research experience and more than 5 years of teaching experience. His research interest includes Data mining, Climate Prediction, Neural Network and Soft Computing.