ANN Approach for Weather Prediction using Back Propagation

Ch.Jyosthna Devi [#1], B.Syam Prasad Reddy[#2], K.Vagdhan Kumar[#3],B.Musala Reddy[#4],N.Raja Nayak[#5]

Department Of Computer Science and Engineering, KLCE, Vaddeswaram, Guntur Dt.-522502, Andhra Pradesh, India

Abstract: **Temperature forecasting is important 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 at a given location. Temperature forecasts are made by collecting quantitative data about the current state of the atmosphere. A neural network can learn complex mappings from inputs to outputs, based solely on samples and require limited understanding from trainer, who can be guided by heuristics. In this paper, a neural network-based algorithm for predicting the temperature is presented .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 proposed idea is tested using the real time dataset. The results are compared with practical working of meteorological department and these results confirm that our model have the potential for successful application to temperature forecasting.**

Keywords:**FeedForwardNeuralNetwork,Temperature prediction,Back propagation,Training,ANN.**

I. INTRODUCTION

Weather simply refer to the condition of air on earth at a given place and time .The application of science and technology are to predict the state of the atmosphere in future time for a given location is so important due to its effectiveness in human life. Today, weather forecasts are made by collecting quantitative data about the current state of the atmosphere and using scientific understanding of atmospheric processes to project how the atmosphere will evolve. The chaotic nature of the atmosphere implies the need of massive computational power required to solve the equations that describe the atmospheric conditions. This is resulted from incomplete understanding of atmospheric processes which mean that forecasts become less accurate as the difference in time between the present moment and the time for which the forecast is being made increases. Weather is a continuous, data-intensive, multidimensional, dynamic and chaotic process and these properties make weather prediction [6] a big challenge. Generally, two methods are used for weather forecasting

(a) the empirical approach and

(b) the dynamical approach.

The first approach is based on the occurrence of analogs and is often referred by meteorologists as analog forecasting. This approach is useful for predicting local-scale weather if recorded data's are plentiful. The second approach is based on equations and forward simulations of the atmosphere and is often referred to as computer modeling. The dynamical approach is only useful for modeling large-scale weather phenomena and may not forecast short-term weather efficiently. Most weather prediction systems use a combination of empirical and dynamical techniques Artificial Neural Network (ANN) provides a methodology for solving many types of nonlinear problems that are difficult to be solved by traditional techniques .Most meteorological processes often exhibit temporal and spatial variability. They are suffered by issues of nonlinearity of physical processes, conflicting spatial and temporal scale and uncertainty in parameter estimates.

II. THE NEURAL NETWORK MODEL

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

- A neural network acquires knowledge through learning.
- A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights .The true power and advantage of neural networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics. In this paper, one model of neural network is selected among the main network architectures used in engineering. The basis of the model is neuron structure as shown in Fig. 1. These neurons act like parallel processing units. An artificial neuron is a unit that performs a simple mathematical operation on its inputs and

imitates the functions of biological neurons and their unique process of learning.

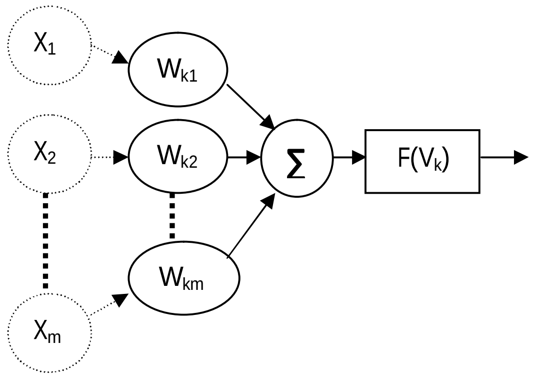


Fig.1 Neuron model

III PROPOSED APPROACH

The aim is to gather dataset consisting weather parameters like temperature, humidity, dew point, visibility ,atmospheric pressure ,sea level, wind speed, wind direction etc. and perform all required data preprocessing tasks .We have normalized the data using min-max normalization to scale the dataset into the range of 0 to 1.Dataset is gathered from a well known website www.weatherunderground.com [5] this normalized data is passed to the BPNN to train the neural network .Thus the neural network is trained by updating all the weights and biases as per the obtained errors in each iteration .Now pass the testing dataset as the network is trained the proposed BPNN model with the developed structure can perform good prediction with least error.

Like this all the neural network prediction algorithms works in weather forecasting .To improve the execution speed and accuracy of BPNN we are going to take "n" neural networks of different architecture and train them and then obtain the results from each network .Now calculate the mean values and treat them as the final prediction values .Now the BPNN predicts with least error and at high speed than before.

IV.ANN APPROACH

An Artificial Neural Network (ANN) [1] 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 new structure of the information processing system. It is composed of a huge number of highly interconnected processing elements (neurons) working together to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a particular application, such as pattern recognition or data classification, through a learning process. Learning in biological systems adds adjustments to the synaptic connections that exist between the neurons.

The ANN has capability to extract the relationship between the inputs and outputs of a process, without the physics being explicitly provided .Thus, these properties of ANN are well suited to the problem of weather forecasting [2] . The main purpose is to develop the most suitable ANN architecture and its associated training technique for weather prediction. This development will be based on using two different neural network architecture to demonstrate the suitable one for this application. Back Propagation (BPN) feed forward network and radial basis function network which were trained by differential evolution algorithm are the selected architectures in this study. The basic architecture of the both Radial Basis Functions (RBF) neural network and multilayer feed forward neural networks are given. Components of a modern weather forecasting system include the following modules: data collection, data assimilation and numerical weather prediction

A. Data collection: Observations of atmospheric pressure, temperature, wind speed, wind direction, humidity and precipitation are made near the earth's surface by trained observers, automatic weather stations. The World Meteorological Organization acts to standardize the instrumentation, observing practices and timing of these observations worldwide.

B. Data assimilation: During the data assimilation process, information gained from the observations is used in conjunction with a numerical model most recent forecast for the time that observations were made to produce the meteorological analysis. This is the best estimate of the current state of the atmosphere. It is a three dimensional representation of the distribution of temperature, moisture and wind .The features considered in this study are bar temperature, bar reading, sea level pressure, mean sea level pressure, dry bulb temperature, wet bulb temperature, due point temperature, vapor pressure, wind speed, humidity, cloudiness, precipitation, wind direction, wind speed and for prediction of rain. It is easy to implement and produces desirable forecasting result by training the given data set.

C. Numerical weather prediction: Numerical Weather Prediction (NWP) uses the power of computers to make a forecast. Complex computer programs, also known as forecast models, run on supercomputers and provide predictions on many atmospheric variables such as temperature, pressure, wind and rainfall. A forecaster examines how the features predicted by the computer will interact to produce the day's weather.

V.FEED FORWARD NEURAL NETWORKS

A typical neural network consists of layers. In a single layered network there is an input layer of source nodes and an output layer of neurons. A multi-layer network has in addition one or more hidden layers. A multi layer neural network is displayed in Fig. 1. Extra hidden neurons raise the network's ability to extract higher order statistics from (input) data. Furthermore a network is said to be fully connected if every node in each layer of the network is connected to every other node in the adjacent forward layer .The network "learns" by adjusting the interconnections (called weights) between layers. When the network is adequately trained, it is able to generalize relevant output for a set of input data. A valuable property of neural networks is that of generalization, whereby a trained neural network is able to provide a correct matching in the form of output data for a set of previously unseen input data.

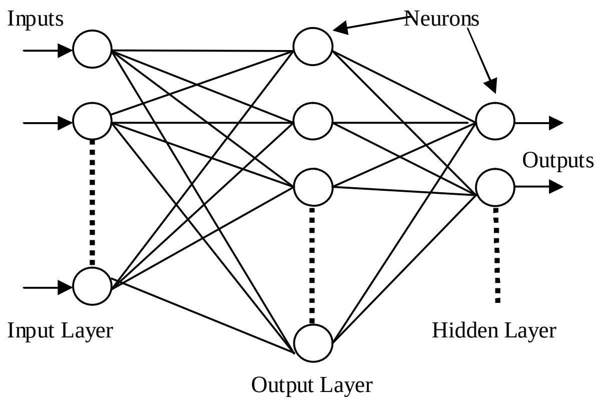


Fig.2 Feed forward Network

VI. BACK PROPAGATION

Back propagation is a common method of teaching artificial neural networks how to perform a given task. It is a supervised learning method, and is a generalization of the delta rule. It requires a teacher that knows, or can calculate, the desired output for any input in the training set. It is most useful for feed-forward networks The term is an abbreviation for "backward propagation of errors". Back propagation [3] requires that the activation function used by the artificial neurons (or "nodes") be differentiable.

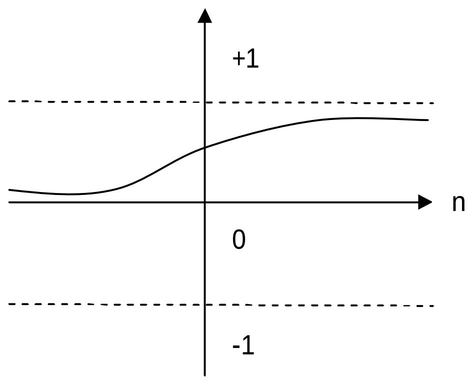


Fig.3. Log-Sigmoid Transfer Function

For better understanding, the back propagation 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. Backward propagation [4] 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 opposite direction of the gradient by subtracting 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 phase 1 and 2 until the performance of the network is trained. Once the network is trained, it will provide the desired output for any of the input patterns.

The network is first initialized by setting up all its weights to be small random numbers – say between –1 and +1. Next, the input pattern is applied and the output calculated (this is called the *forward pass*). The calculation gives an output which is completely different to what you want (the Target), since all the weights are random. We then calculate the *Error* of each neuron, which is essentially: *Target – Actual Output.* This error is then used mathematically to change the weights in such a way that the error will get smaller. In other words, the Output of each neuron will get closer to its Target (this part is called the *reverse pass*. The process is repeated again and again until the error is minimal. Let's do an example with an actual network to see how the process works. We'll just look at one connection initially, between a neuron in the output layer and one in the hidden layer,

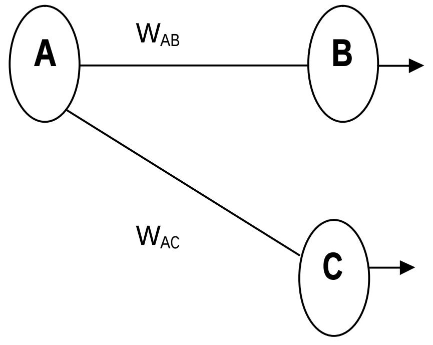


Fig.4 a single connection learning in a BPN

The connection we're interested in is between neuron A (a hidden layer neuron) and neuron B (an output neuron) and has the weight WAB. The diagram also shows another connection, between neuron A and C, but we'll return to that later. The algorithm works like this:

ALGORITHM:

1. First apply the inputs to the network and work out the output – remember this initial output could be anything, as the initial weights were random numbers.

2. Next work out the error for neuron B. The error is *What you want – What you actually get*, in other words:

ErrorB = OutputB (1-OutputB)(TargetB – OutputB)

The "*Output(1-Output)*" term is necessary in the equation because of the Sigmoid Function – if we were only using a threshold neuron it would just be *(Target – Output).*

3. Change the weight. Let W+ AB be the new (trained) weight and WAB be the initial weight. W+ AB = WAB + (ErrorB x OutputA)

Notice that it is the output of the connecting neuron (neuron A) we use (not B). We update all the weights in the output layer in this way.

4. Calculate the Errors for the hidden layer neurons. Unlike the output layer we can't calculate these directly (because we don't have a Target), so we *Back Propagate* them from the output layer This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors.

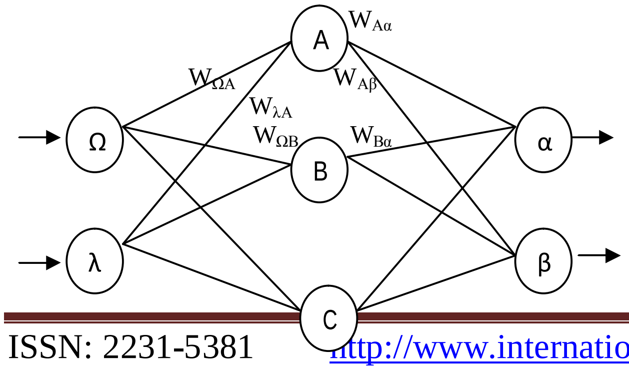
For example if neuron A is connected as shown to B and C then we take the errors from B and C to generate an error for A.

ErrorA = Output A (1 - Output A)(ErrorB WAB + ErrorC WAC)

Again, the factor "*Output (1 - Output)*" is present because of the sigmoid squashing function.

5. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers.

This may well have left some doubt in your mind about the operation, so let's clear that up by explicitly showing *all* the calculations for a full sized network with 2 inputs, 3 hidden layer neurons and 2 output neurons as shown in figure 3.4. W+ represents the new, recalculated, weight, whereas W represents the old weight.



$W_{\lambda B}$ $W_{B\beta}$

$W_{\Omega A}$ $W_{C\alpha}$

$W_{\lambda C}$

$W_{C\beta}$

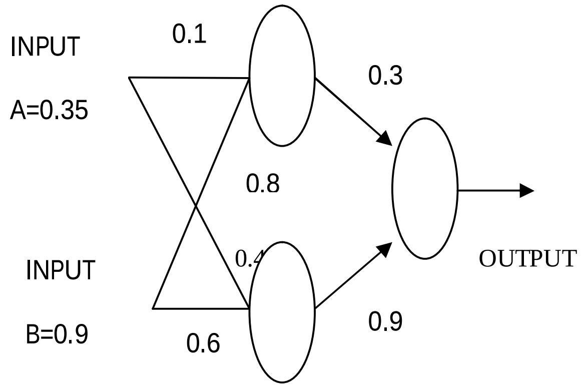
1. Calculate errors of output neurons
$\delta_\alpha = out_\alpha\ (1 - out_\alpha)\ (Target_\alpha - out_\alpha)$
$\delta_\beta = out_\beta\ (1 - out_\beta)\ (Target_\beta - out_\beta)$
2. Change output layer weights
$W^+_{A\alpha} = W_{A\alpha} + \eta\delta_\alpha\ out_A$
$W^+_{A\beta} = W_{A\beta} + \eta\delta_\beta\ out_A$
$W^+_{B\alpha} = W_{B\alpha} + \eta\delta_\alpha\ out_B$
$W^+_{B\beta} = W_{B\beta} + \eta\delta_\beta\ out_B$
$W^+_{C\alpha} = W_{C\alpha} + \eta\delta_\alpha\ out_C$
$W^+_{C\beta} = W_{C\beta} + \eta\delta_\beta\ out_C$

3. Calculate (back-propagate) hidden layer errors
$\delta A = out_A\ (1 - out_A)\ (\delta\alpha W_{A\alpha} + \delta\beta\ W_{A\beta})$
$\delta B = out_B\ (1 - out_B)\ (\delta\alpha\ W_{B\alpha} + \delta\beta\ W_{B\beta})$
$\delta C = out_C\ (1 - out_C)\ (\delta\alpha\ W_{C\alpha} + \delta\beta\ W_{C\beta})$
4. Change hidden layer weights
$W^+_{\lambda A} = W_{\lambda A} + \eta\delta_A\ in_\lambda$
$W^+_{\Omega A} = W^+_{\Omega A} + \eta\delta_A\ in_\Omega$
$W^+_{\lambda B} = W_{\lambda B} + \eta\delta_B\ in_\lambda$
$W^+_{\Omega B} = W^+_{\Omega B} + \eta\delta_B\ in_\Omega$
$W^+_{\lambda C} = W_{\lambda C} + \eta\delta_C\ in_\lambda$
$W^+_{\Omega C} = W^+_{\Omega C} + \eta\delta_C\ in_\Omega$

The constant η (called the learning rate, and nominally equal to one) is put in to speedup or slow down the learning if required.

To illustrate this let's do a worked Example

Consider the simple network below:



Assume that the neurons have a Sigmoid activation function and

(i) Perform a forward pass on the network.
(ii) Perform a reverse pass (training) once (target = 0.5).
(iii) Perform a further forward pass and comment on the

result.

Answer:(i)

Input to top neuron = (0.35x0.1)+(0.9x0.8)=0.755. Out = 0.68.

Input to bottom neuron = (0.9x0.6)+(0.35x0.4) = 0.68. Out = 0.6637.

Input to final neuron = (0.3x0.68)+(0.9x0.6637) = 0.80133. Out = 0.69.

(ii)Output error δ=(t-o)(1-o)o = (0.5-0.69)(1-0.69)0.69 = -0.0406.

New weights for output layer

w1+ = w1+(δ x input) = 0.3 + (-0.0406x0.68) = 0.272392.

w2+ = w2+(δ x input) = 0.9 + (-0.0406x0.6637) = 0.87305.

Errors for hidden layers:

δ1 = δ x w1 = -0.0406 x 0.272392 x (1-o)o = -2.406x10-3

δ2= δ x w2 = -0.0406 x 0.87305 x (1-o)o = -7.916x10-3

New hidden layer weights:

w3+=0.1 + (-2.406 x 10-3 x 0.35) = 0.09916.

w4+ = 0.8 + (-2.406 x 10-3 x 0.9) = 0.7978.

w5+ = 0.4 + (-7.916 x 10-3 x 0.35) = 0.3972.

w6+ = 0.6 + (-7.916 x 10-3 x 0.9) = 0.5928.

(iii)

Old error was -0.19. New error is -0.18205. Therefore error has reduced.

VII CONCLUSION

In this paper, how neural networks are useful in forecasting the weather and the working of most powerful prediction algorithm called back propagation algorithm was explained . A 3-layered neural network is designed and trained with the existing dataset and obtained a relationship between the existing non-linear parameters of weather. Now the trained neural network can predict the future temperature with less error.

REFFERENCES

[1] MohsenHayati, and Zahra Mohebi,"*Application of Artificial Neural Networks for Temperature Forecasting,*" World Academy of Science,Engineering and Technology 28 2007.

[2] Arvind Sharma, Prof. Manish Manoria," *A Weather Forecasting System using concept of Soft Computing,*" pp.12-20 (2006)

[3] Raúl Rojas," *The back propagation algorithm of Neural Networks - A Systematic Introduction,* "chapter 7, ISBN 978-3540605058

[4] K.M. Neaupane, S.H. Achet," Use of backpropagation neural network for landslide monitoring," Engineering Geology 74 (2004) 213–226.

[5] http://www.wunderground.com/history/airport/VABB/2010/9/1/CustomHistory.html?dayend=1&monthend=9&yearend=2011&req_city=NA&req_state=NA&req_statename=NA

[6] lai l.l. et al." Intelligent weather forecast" *Third international conference on machine learning and cybernetics, shanghai,* 2004.

[7] Mathur S. et al " A feature based neural network model for weather forecasting" *World academy of science, engineering and technology 34*,2007.

[8] Isa I.S. et al. "Weather forecasting using photovoltaic system and neural network" *Second international conference on computational intelligence, communication systems and networks,* 2010.

[9] Baboo S.S. and Shereef I.K. " An efficient weather forecasting system using artificial neural network" *International journal*

[10]Gill J. et al "Training back propagation neural networks with genetic algorithm for weather forecasting" *IEEE 8th international symposium on intelligent systems and informatics serbia,* 2010.

[11]" Data Mining: Concepts and Techniques" by *Jiawei Han and Micheline Kamber,* Morgan Kaufmann, 2001.