IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. IV (May-Jun. 2014), PP 55-58
www.iosrjournals.org

Efficient Weather Prediction By Back-Propagation Algorithm

Manisha Kharola[1] and Dinesh Kumar[2]
[1]*M.Tech Scholar of computer science & Engineering , SRCEM, Palwal*
[2]*Department of computer science & Engineering , SRCEM, Palwal*

Abstract: *Artificial Neural Networks (ANNs) have been applied extensively to both regress and classify weather phenomena. While one of the core strengths of neural networks is rendering accurate predictions with noisy data sets, there is currently not a significant amount of research focusing on whether ANNs are capable of producing accurate predictions of relevant weather variables from small-scale, imperfect datasets. Our paper makes effort to use back propagation algorithm to train the network. So, that it can help in predicting the future weather.*
Keywords: *ANN, Back Propagation, Weather Prediction.*

I. Introduction

Weather process is a dynamic and non linear phenomenon. There is a need to apply statistical post-processing techniques on modeled forecast fields to improve the prediction quality and value. India is an agricultural country so weather prediction plays a very important role. If we knew that what will be the weather in future then we can work according to that. The weather prediction should be accurate and more precise.

An Artificial Neural Network (ANN)is an information –processing paradigm that is inspired by the way biological nervous system such as the brain, process information. It is composed of a large number of highly interconnected processing elements(neurons) working in parallel to solve the specific problems. The weight on the connections encodes the knowledge of a network. Each neuron has the local memory and the output of each neuron depends upon only the input signals arriving at the neuron and value in neuron’s memory. The intelligence of a neural network emerges from the collective behaviour of neurons, each of which performs only very limited operations. Even though each individual neuron works slowly, they can still quickly find the solutions by working in parallel. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. Using ANN has following benefits:-

- No linearity- The answer from the computational neuron can be linear or not.
- Adaptive learning- The ANN is capable of determine the relationship between the different examples which are presented to it without requiring a previous model.
- Self-organization- This property allows the ANN to distribute the knowledge in the entire network structure, there is no element with specific stored information.
- Fault tolerance- This can be shown in two senses: The first is related to the samples shown to the network, in which case it answers correctly even when the examples exhibit variability or noise; the second appears when in any of the elements of the network occurs a failure, which does not affect its functioning due to the way in which it stores information.

Back propagation is a method of training multilayer ANNs which use the procedure of supervised learning. Supervised algorithms are error-based learning algorithms which utilize an external reference signal(teacher) and generate an error signal by comparing the reference with the obtained output. Based on error signal, neural network modifies its synaptic connection weights to improve the system performance. In this scheme, it is always assumed that the desired answer is known a priori[2].

The traditional Back Propagation Neural Network(BPNN) Algorithm is generally used in solving many practical problems. The BPNN learns by calculating the errors of the output layer to find the errors in the hidden layers. Back-Propagating is highly appropriate for problems in which no relationship is found between the output and inputs. Due to its suppleness and learning capability, it has been effectively implemented in wide range of applications[3].

II. Literature Survey

Many works were done related to the weather forecasting system and BPNN. They are summarized below.

Singh, Bhambri and Gill[5] has highlighted on the temperature prediction is a temporal and time series based on process. Due to non linearity in climatic physics, neural network are suitable to predict these meteorological processes. Back Propagation integrated with genetic algorithm is the most important to train the neural networks. In this paper in order to show the dependence of temperature on a particular data series, a time series based temperature prediction model using integrated back propagation with genetic algorithm technique algorithm is proposed. In the proposed technique, the effect of under training and over training the system is also shown.

Aliev et al., 2008, have proposed, fuzzy recurrent neural network (FRNN) based time series forecasting method for solving forecasting problems[6]. Nekoukar et al., 2010, have used radial basis function neural network for financial time-series forecasting, and the result of their experiment shows the feasibility and effectiveness[7].

Y. Radhika and M.Shashi[8] presents an application of Support Vector Machines(SVMs) for weather prediction. Time series data of daily maximum temperature at location is studied 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 for various spans of 2 to 10 days by using optimal values of the kernel.

Wojtylak[9] analyses the fuzzy weather forecasts, which are computed in the system and used to forecast pollution concentrations and to investigate the effectiveness of forecasting pollution concentrations, putting the dependence between particular attributes, describing the weather forecast in order and proving the applicable fuzzy numbers in air pollution forecasting.

Khalid J. and Steve[10] in a study on ”Knowledge Based System for Weather Information Processing and Forecasting” have five components as Image Aequisition, Image Processing and Enhancement, Feature Extraction and Selection, Weather Knowledge Base and Weather Inference Engine(WINE). This model is designed for the physical observations from the satellite imagery and the meteorological information.

III. Back Propagation Algorithm

A. Overview of Algorithm

A Back-Propagation network consists of at least three layers of units: an input layer, at least one intermediate hidden layer and an output layer. 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 layers[11].

The back propagation method for training multilayer feed forward network is conducted in two phases. There is first a feed forward phase in which input patterns are presented and the network computes output values, producing an error value for each pattern at each output unit. This is followed by a feed backward phase in which error derivatives at each unit are computed and weight are adjusted to reduce errors.

In BPNN the input pattern is presented to the input layer of the network. These inputs are propagated through the network until they reach the output units. This forward pass produces the actual or predicted output pattern. Because back propagation is a supervised learning algorithm, the desired output are given as part of training vector. The actual network output are subtracted from the desired outputs and an error signal is produced. This error signal is then the basis for the back propagation step, where by the error are passed back through the neural network by computing the contribution of each hidden processing unit and deriving the corresponding adjustment needed to produce the correct output[2].

B. Phases in Back Propagation Technique

The back propagation [1] 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 is given through the neural network in order to generate the propagation's output activations.
2. Back propagation of the output activations propagation 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 input activation and output delta 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 impacts on the speed and quality of learning; it is called the learning rate. The sign of the gradient of a weight designates where the error is increasing; this is why the weight must be updated in the opposite direction.

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

C. Mathematical Analysis of Algorithm

Assume a network with N inputs and M outputs.

Weight Initialization

Set all the weights and node thresholds to small random numbers. But node thresholds is the negative of the weight from the bias unit.

Calculation of Activation

1. The activation level of an input unit is determined by the instances presented to the network.
2. The activation level Oj of hidden and output unit is determined by

   $$\mathbf{Oj = F\,(\textstyle\sum Wji\ Oj\text{-}\theta j)}$$

   Where **Wji** is the weight from an input **Oj**, **θj** is the node threshold, and **F** is activation function.

   $$\mathbf{F(a)=1/(1+eexp(-a))}$$

Weight Training

1. Start at the output units and work backward to the hidden layers recursively. Adjust weight by

   $$\mathbf{Wji(t+1)= Wji(t) + \Delta Wji}$$

   Where **Wji (t)** is the weight from unit i to unit j at time t and **ΔWji** is the weight adjustment.
2. The weight change is computed **by $\Delta Wji = \eta \delta j Oi$.**

   Where Δ is trial – independent learning rate**(0<Δ<1)** and **δj** is the error gradient at unit j. **η** is the learning rate.

   Convergence is sometime faster by adding a momentum term:

   $$\mathbf{Wji(t+1)= Wji(t)+ \eta \delta j Oi +\alpha\ [Wji(t) - Wji(t\text{-}1)]}$$
3. The error gradient is given by:

   For the output units:

   $$\mathbf{\delta j = Oj(1\text{-}\ Oj)(\ Tj\text{-}\ Oj\ )}$$

   where Tj is the desired output activation and Oj is actual output activation at output unit j.

   For the hidden units:

   $$\mathbf{\delta j = Oj(1\text{-}\ Oj) \textstyle\sum \delta k\ Wkj}$$

   where **δk** is error gradient at unit k to which a connection points from hidden unit j.
4. Repeat iterations until convergence in terms of the selected error criterion. An iteration include presenting an instance, calculating activations and modifying weights.

IV. Proposed Work

The steps of the proposed work are given:

A. Data Collection

In this step various sensors are used to collect the data like wind sensors, rain sensor, pressure sensor and temperature sensors. The data is collected repeatedly after short interval of time so that we have enough input for the process. Large dataset helps in increasing the accuracy of the output. After this the data is send for preprocessing.

B. Data Preprocessing and Data analysis

In this the data is preprocessed for the errors in it. Data is normalized. Data normalizing is the process of scaling data to fall within a smaller range. Normalizing help in speeding up the learning phase. Data mining is done to deal with the noisy data and the missing value.

C. Training

Training of the network is done with the help of back propagation algorithm. In this we have to choose learning rate and the momentum value. Learning rate control the speed of the network. As we increase the learning rate it will speed up the training. We are using tanh activation function because it provides more recognition and accuracy. We can use any number of hidden layers or can increase it if the network is not learning well. The input of the back propagation network is temperature, rainfall, humidity, pressure and precipitation.

D. Testing

In this testing is done by providing with various other dataset as input and getting the desired result. When we do not get accurate results than the network is again trained. Network is checked with different situations that can occur in future.

E. Output

When the testing step is complete the output result is provided. On the basis of this result we classify that how will be the future weather.

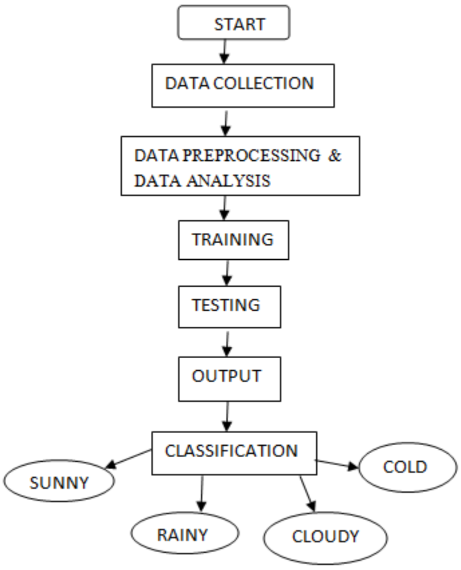

Figure 1. Process of Weather Prediction

V. Result And Conclusion

This paper provides a new method to predict the future weather with the help of back propagation training algorithm. It was found that the network learns very fast with back propagation algorithm. The results are more accurate for predicting the future weather. Back propagation is a gradient descent algorithm which learns by minimizing the error in the output by adjusting the weights in the network. Tanh activation function is used as it provides more accurate results when the number of iterations is increased. This is used for a small area but in future we can extend our research to greater extend.

References

[1] Raúl Rojas," The back propagation algorithm of Neural Networks - A Systematic Introduction, "chapter 7, ISBN 978-3540605058
[2] Saduf, Mohd Arif Wani,"Comparative study of Back Propagation Learning Algorithm for Neural Networks,"IJARCSSE,Volume:3,Issue:12,Dec2013.
[3] Shital Solanki,H.B.Jethva "A review on back propagation algorithms for Feedforward Networks" GRA Volume : 2 | Issue : 1 Volume : 2 | Issue : 1 | Jan 2013.
[4] Gunjan Goswami, "Introduction to Artificial Neural Network".
[5] Singh, Bhambri and Gill, "Time Series based Temperature Prediction using Back Propagation with Genetic Algorithm Technique",IJCSI,Volume:8,Issue5,No.3,Sep2011.
[6] Aliev R. A., Fazlollahi B., Aliev R. R., Guirimov B., 2008, "Linguistic time series forecasting using fuzzy recurrent neural network", Soft Comput, vol. 12, pp. 183–190.
[7] Nekoukar Vahab, Taghi Mohammad, Beheshti Hamidi, 2010, "A local linear radial basis function neural network for financial time-series forecasting", Springer Science, vol. 23, pp. 352–356.
[8] 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.
[9] D. Domanskam. Wojtylak, "Fuzzy Weather Forecast In Forecasting Pollution Concentrations".
[10] Siddiqui Khalid J. and Nugen Steve M., Knowledge Based System for Weather Information Processing and Forecas-ting, Department of Computer Science, SUNY at Fredonia, NY 14063, IEEE 1966.
[11] Shital Solanki and H.B Jethva,"A review on back propagation algorithm for Feedforward Networks",GRA,Volume:2,Issue:1,Jan 2013.