ISSN:2249-5789
Parag P Kadu et al, International Journal of Computer Science & Communication Networks,Vol 2(1), 61-64

Temperature Prediction System Using Back propagation Neural Network : An Approch

Parag.P.Kadu
M.Tech. IV Sem,Department of CSE,
Govt. College of Engineering,
Amravati, India.
paragkadu24@gmail.com

Prof.Kishor P.Wagh
Asst. Prof, Department of CSE,
Govt. College of Engineering,
Amravati,India.
kishorwagh2000@yahoo.com

Dr.Prashant N. Chatur
Head,Department of CSE,
Govt.College of Engineering,
Amravati,india.
chatur.prashant@gcoea.ac.in

Abstract

This paper utilizes artificial neural networks for temperature forecasting. Our study based on back propagation neural network which is trained and tested based on dataset provided. In formulating the ANN-based predictive model; three-layer network has been constructed. Suitable air temperature predictions can provide farmers and producers with valuable information when they face decisions regarding the use of mitigating technologies such as orchard heaters or irrigation. The research presented in this thesis developed artificial neural networks models for the prediction of air temperature. In this paper, back propagation neural network is used for temperature forecasting. The technical milestones, that have been achieved by the researchers in this field has been reviewed and presented in this paper. From the past decades there are various models are developed for weather forecasting using artificial neural network, and by using soft computing, which are discussed in this paper. Artificial neural networks and the back propagation algorithm used for temperature forecasting in general are explained.

1. Introduction

Air temperatures prediction is of a concern in environment, industry and agriculture. The climate change phenomenon is as the first environmental problem in the world threatening the human beings. The industrial activities are so effective in this problem and cause the global warming which the world has been faced with, lately. Knowing the variability of ambient temperature is important in agriculture because extreme changes in air temperature may cause damage to plants and animals [4].Air temperature forecasting is useful in knowing the probability of tornado, and flood occurrence in an area. Prediction of the energy consumption, soil surface temperature and solar-radiation are also related to ambient air temperature forecasting.

Artificial Neural Network (ANN), a component of soft computing, is highly suitable for the situations where the underlying processes exhibit chaotic features. The concept of ANN is originated from the attempt to develop a mathematical model capable of recognizing complex patterns on the same line as biological neuron work. It is useful in the situations where underlying processes / relationships display chaotic properties. ANN does not require any prior knowledge of the system under consideration and are well suited to model dynamical systems on a real-time basis. It is, therefore, possible to set up systems so that they would adapt to the events which are observed and for this, it is useful in real time analyses, e.g., weather forecasting, different fields of predictions, etc.

ANN provides a methodology for solving many types of non-linear problems that are difficult to solve by traditional techniques. Most meteorological processes often exhibit temporal and spatial variability, and are further plagued by issues of non-linearity of physical processes, conflicting spatial and temporal scale and uncertainty in parameter estimates. With ANN, there exists the capability to extract the relationship between the inputs and outputs of a process. Thus, these properties of ANN are well suited to the problem of weather forecasting under consideration [2].

One type of network sees the nodes as ‘artificial neurons’. These are called artificial neural networks .An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signals received are strong enough, the neuron is activated and emits a signal though the axon. This signal might be sent to another synapse, and might activate other neurons. The complexity of real neurons is highly abstracted when modeling artificial neurons. These basically consist of inputs, which are multiplied by weights, and then computed by a

mathematical function which determines the activation of the neuron [1].Another function computes the output of the artificial neuron. ANN combines artificial neurons in order to process information.

The higher a weight of an artificial neuron is, the stronger the input which is multiplied by it will be. Weights can also be negative, so we can say that the signal is inhibited by the negative weight. Depending on the weights, the computation of the neuron will be different. By adjusting the weights of an artificial neuron we can obtain the output we want for specific inputs. But when we have an ANN of hundreds or thousands of neurons, it would be quite complicated to find by hand all the necessary weights. But we can find algorithms which can adjust the weights of the ANN in order to obtain the desired output from the network. This process of adjusting the weights is called learning or training [Fig. 1].

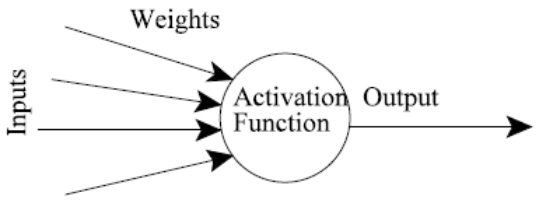

Fig.1 An Artificial Neuron.

2.Materials and Methods

A. Data Sets

The data used in this study are daily and monthly for air temperature prediction were collected from wireless kit and wunderground.com, with the help of these parameters, forecast temperature using statistica software as a platform. The different sensors like rain sensor, wind sensor, and thermo-hydro sensor records different parameters like rainfall, wind, temperature and humidity. The recorded data is present in the form of datasheet [9]. This data set is send for pre-processing and then to the statistica software. These data were considered in three different data sets including training, test and validation in artificial neural network.

B. Back Propagation Learning

The simple perceptron is just able to handle linearly separable or linearly independent problems. By taking the partial derivative of the error of the network with respect to each weight, we will learn a little about the direction the error of the network is moving. In fact, if we take the negative of this derivative (i.e. the rate change of the error as the value of the weight increases) and then proceed to add it to the weight, the error will decrease until it reaches local minima. This makes sense because if the derivative is positive, this tells us that the error is increasing when the weight is increasing. The obvious thing to do then is to add a negative value to the weight and vice versa if the derivative is negative. Because the taking of these partial derivatives and then applying them to each of the weights takes place, starting from the output layer to hidden layer weights, then the hidden layer to input layer weights (as it turns out, this is necessary since changing these set of weights requires that we know the partial derivatives calculated in the layer downstream), this algorithm has been called the back propagation algorithm [5]. A neural network can be trained in two different modes: online and batch modes. The number of weight updates of the two methods for the same number of data presentations is very different. The online method weight updates are computed for each input data sample, and the weights are modified after each sample. An alternative solution is to compute the weight update for each input sample, but store these values during one pass through the training set which is called an epoch. At the end of the epoch, all the contributions are added, and only then the weights will be updated with the composite value. This method adapts the weights with a cumulative weight update, so it will follow the gradient more closely. It is called the batch-training mode. Training basically involves feeding training samples as input vectors through a neural network, calculating the error of the output layer, and then adjusting the weights of the network to minimize the error.

The previously mentioned back-propagation learning algorithm works for feed-forward networks with continuous output. Training starts by setting all the weights in the network to small random numbers. Now, for each input example the network gives an output, which starts randomly. We measure the squared difference between this output and the desired output—the correct class or value. The sum of all these numbers over all training examples is called the total error of the network. If this number was zero, the network would be perfect, and the smaller the error, the better the network. By choosing the weights that minimize the total error, one can obtain the neural network that best solves the problem at hand. This is the same as linear regression, where the two parameters characterizing the line are chosen such that the sum of squared differences between the line and the data points is minimal. This can be done analytically in linear regression, but there is no analytical solution in a feed-forward neural network with hidden units. In back-propagation, the weights and thresholds are changed each time an example is presented, such that the error gradually becomes smaller. This is repeated, often hundreds of times, until the error no longer changes. In back-propagation, a numerical optimization technique called gradient descent makes the math particularly simple; the form of the equations gave rise to the name of this method. There are some learning parameters (called learning rate

and momentum) that need tuning when using back-propagation, and there are other problems to consider. For instance, gradient descent is not guaranteed to find the global minimum of the error, so the result of the training depends on the initial values of the weights. However, one problem overshadows the others: that of over-fitting [3]. Over-fitting occurs when the network has too many parameters to be learned from the number of examples available, that is, when a few points are fitted with a function with too many free parameters. Although this is true for any method for classification or regression, neural networks seem especially prone to over parameterization. A network that over fits the training data is unlikely to generalize well to inputs that are not in the training data. There are many ways to limit over-fitting (apart from simply making small networks), but the most common include averaging over several networks, regularization and using methods from Bayesian statistics. To estimate the generalization performance of the neural network, one needs to test it on independent data, which have not been used to train the network. This is usually done by cross-validation, where the data set is split into, for example, and ten sets of equal size. The network is then trained on nine sets and tested on the tenth, and this is repeated ten times, so all the sets are used for testing. This gives an estimate of the generalization ability of the network; that is, its ability to classify inputs that it was not trained on. To get an unbiased estimate, it is very important that the individual sets do not contain examples that are very similar.

C.Statistica Software

I. **Select the Variables for the Analysis**

Statistica data miner distinguishes between categorical and continuous variables and dependent and predictor. Categorical variables are those that contain information about some discrete quantity. Continuous variables are measured on a continuous scale [10]. Dependent variables are those we want to predict. Predictor variables are those that we want to use for prediction or classification.

II. **Feature Selection and Variable Screening**

The Feature selection and variable screening module will automatically select subsets of variables from extremely large data files or databases connected for in-place processing .The module can handle a practically unlimited number of variables. Literally millions of input variables can be scanned to select predictors for regression or classification. Specifically, the program includes several options for selecting variables that are likely to be useful or useful or informative in specific subsequent analysis [11]. The unique algorithms implemented in the feature selection and variable screening module will select continuous or categorical predictor variables that show a relationship to the continuous or categorical dependent variables of interest, regardless of weather that relationship is simple or complex.

III. **Factor Analysis**

The Factor Analysis module contains a wide range of statistics and options, and provides a comprehensive implementation of factor analytic techniques with extended diagnostics and a wide variety of analytic and exploratory graphs. It performs principal components and common and hierarchical factor analysis and can handle extremely large analysis problems.

IV. **Statistica Neural Network**

Statistica Neural Network is a comprehensive, state-of-the-art, powerful and extremely fast neural networks data analysis package that contains the following features: integrated pre and post processing, including data selection, nominal-value encoding, scaling, normalization and missing value substitution with interpretation for classification, regression and time series problems[11]. Statistica Neural Network has numerous facilities to aid in selecting appropriate network architecture. Statistica Neural Network statistical and graphical feedback includes bar charts, matrices and graphs of individual and overall case errors, vital statistics such as regression error ratios, which are all automatically calculated.

3.Training and Testing Neural Network

The best training procedure is to compile a wide range of examples (for more complex problems, more examples are required), which exhibit all the different characteristics of the problem. To create a robust and reliable network, in some cases, some noise or other randomness is added to the training data to get the network familiarized with noise and natural variability in real data [7]. Poor training data inevitably leads to an unreliable and unpredictable network. Usually, the network is trained for a prefixed number of epochs or when the output error decreases below a particular error threshold. Special care is to be taken not to over train the network. By overtraining, the network may become too adapted in learning the samples from the training set, and thus may be unable to accurately classify samples outside of the training set.[Fig. 2]

A. Choosing the Number of Neurons

The number of hidden neurons affects how well the network is able to separate the data. A large number of hidden neurons will ensure correct learning, and the network is able to correctly predict the data it has been trained on, but its performance on new data, its ability to generalize, is compromised [5]. With too few hidden neurons, the network may be unable to learn the relationships amongst the data and the error will fail to fall below an acceptable level. Thus, selection of the number of hidden neurons is a crucial decision.

B. Choosing the Intial Weights

The learning algorithm uses a steepest descent technique, which rolls straight downhill in weight space until the first valley is reached. This makes the choice of initial starting point in the multidimensional weight space critical. However, there are no recommended rules for this selection except trying several different starting weight values to see if the network results are improved.

C. Choosing the Learning Rate

Learning rate effectively controls the size of the step that is taken in multidimensional weight space when each weight is modified. If the selected learning rate is too large, then the local minimum may be overstepped constantly, resulting in oscillations and slow convergence to the lower error state[6]. If the learning rate is too low, the number of iterations required may be too large, resulting in slow performance.

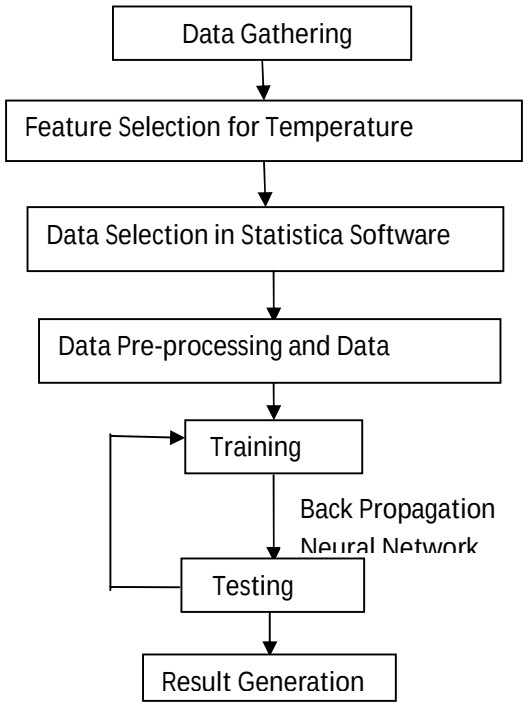

Figure 2: Flowchart of temperature prediction system

4. Result and Discussion

The obtained results indicate that satisfactory prediction accuracy has been achieved through back propagation neural network. A back propagation neural network with gradient descent method minimizes the error rate and it is a promising approach for temperature forecasting.

Mean, minimum and maximum air temperatures were considered as input and output of the network. Three different data sets were extracted from the input and target data for training, validation and test phases. While training set consists of 50 percent of data to build the model and determine the parameters such as weights and biases, validation data set includes 25 percent to measure the performance of network by holding constant parameters. Finally, 25 percent of data is used to increase the robustness of model in the test phase.

The validation and testing phases are very important due to misleading of small error in the training phase. If the network is not trained well due to the irrelevant data of the individual cases such as over fitting, it leads to the small error in the training set and makes large error during validation and test phases. While the purpose of training phase is based on learning, it is not a good metric for the performance of network in validation phase.

5. Conclusion

Neural-networks-based ensemble models were developed and applied for hourly temperature forecasting. The experimental results show that the ensemble networks can be trained effectively without excessively compromising the performance. The ensembles can achieve good learning performance because one of the ensemble's members is able to learn from the correct learning pattern even though the patterns are statistically mixed with erroneous learning patterns.

6. References

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

[2]R.A. Pielke, "A comprehensive meterological modeling system RAMS," *Meteorology and Atmosphe-ric Physics, Springer*- Verlag Vol. 49, 69-91p,1992.

3]B.A. Smith " Improving air temperature prediction with rtificial neural networks" *International journal of omputational intelligence,* 2007.

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

[5]Senduru Srinivasulu, "Extracting Spatial Semantics in Association Rules for Weather Forecasting Image", *Research Scholar Department of Information Technology, Sathyabama University Chennai, India* IEEE 2010

[6]S.S. Baboo and I.K Shereef. " An efficient weather forecasting system using artificial neural network" *International journal of environmental science and development,* vol. 1, no. 4, 2010.

[7]Y.Wang and S.Banavar "Convective Weather Forecast Accuracy Analysis at center and sector levels", *NASA Ames Research center, Maffett Field, Califonia.*

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

[9]Weather.com,http://www.weather.com

[10]AccuWeather.com,http://www.accuweather.com

[11]Statistica SANN-v9_training_manual.