**WEATHER PATTERN ANALYSIS USING DATA MINING**

Archan Roychoudhury

**Abstract** Artificial Neural Networks (ANN) is widely being used to classify and also predict weather patterns. Back propagation algorithms using ANN gives accurate predictions for noisy data which have been rolled up during pre-processing. Here we will implement the back propagation algorithm using MATLAB R2017a to simulate and test an ANN for weather pattern analysis.

**1.Introduction**

Accurate weather forecasting has become a very essential and impactful study in computer science and data analytics as all modern industries, communications and transport sectors depend on correct analysis of historic data to give accurate predictions of weather and climate.

This can be a very difficult process as weather data can be very noisy and inaccurate. Collected data may present several challenges as they may be daily, monthly, for six months, etc. That is why it is important to pre-process and roll-up the data to decrease noise.

Backward Propagation which is short for "backward propagation of errors," uses gradient descent for supervised learning inANN. Having an ANN and an error function allows the algorithm to calculate the gradient of the error function with respect to the weights of interconnected nodes in different hidden layers of the ANN.

Here backwards means that the calculation of the gradient takes from the back to the front that ensures that the gradients of the layers in the ANN which are in the final layer of weights are calculated before the gradients of the initial layers.

The unfinished calculations from the forward layers are used as inputs for the previous layers. The flow of error information backward helps in accurate calculation of gradient in consecutive layers rather than to calculate the gradient of each layer without any prior knowledge.

The algorithm uses a bottom-up approach in every step to generate 1 output for every n inputs.

**2.Literature Survey**

Neural Networks is a paradigm for processing large sets data. Neural Network modelling represents a human brain in which each Neurons represent a particular parameter based on which our data is to be analysed. There are numerous algorithms under Neural Networks like Back propagation algorithm, Fuzzy clustering, Radial basis function etc. Here we consider the algorithm Back propagation due to the noise present in the dataset. Back propagation falls under supervised learning method as it needs to know the kind of output needed for each input value. The basics of this method were derived by Henry J. Kelley in 1960 and Arthur E. Bryson in 1961. This method began to lack its lustre during the 2000s but it returned after a decade for powerful GPU base computing system. The working and the details of Back Propagation can be understood from Medskar and Liebowitz (1994) and Russel and Norvig (1995).

Chattopadhyay and Chattopadhyay (2008a),Ch.Jyotsana,Syam Prasad ,Vagdhan Kumar,BMusala & N.Raja(2012), and also Chao Xi Erihe (2013) have utilised feed-forward backward propagation method to calculate the gradients of ANN to predict weather patterns.

**3.Work Methodology**

Here we have used a feed-forward neural network with one input layer, one output layer and one or more hidden layers. The neurons inside every layer is interconnected with weighted synapses. This technique is most commonly used in weather pattern analysis as it can support any number of neurons for any number of layers.

Mostly weather pattern analysis is carried out with one intermediate hidden layer to map linear functions. However, as weather data is truly non-linear we have taken the approach of testing with more than one hidden layer to map discontinuous functions.

**3.1. Data Specification:**

The dataset used is taken from <http://www.weatherbase.com/weather/weather.php3?s=595983&cityname=Vellore-India>.

The dataset gives the average values of several weather parameters of the Vellore area. The data is a collection of 102 years of weather of Vellore.

Since, historical data tends to be inaccurate and noisy the data for the whole 102 years is already rolled-up to a monthly and annual average basis.

The parameters that we use here are:

|  |
| --- |
| Avg Temp(f) |
| Avg High Temp(f) |
| Avg Low Temp(f) |
| Avg Precipitation(in) |
| Avg Days with Precipitation |
| Avg Vapour Pressure(Hg) |
| Avg Length of Day(Hours) |
| Avg Cloud Cover (%) |
| Avg Daily Solar Radiation(kWh/m2) |

For the input data we write the data to an excel sheet of 9\*12 values for the 12-month average data values of the 9 parameters used.

For the target data we use the annual averages of the parameter values and write it to an 9\*1 excel sheet.

We import these as csv files to MATLAB R2017a.

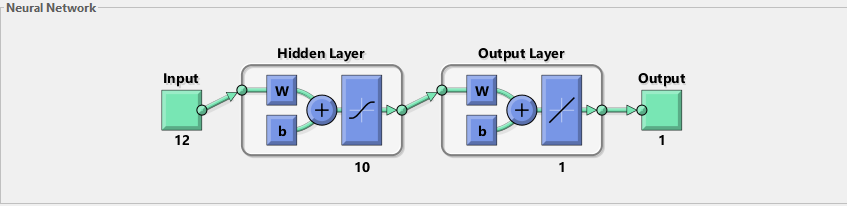
**3.2. Neural network modelling:**

The most important part is to create the ANN for which we use the “nntool” (neural network tool) which gives a simple graphic interface forpattern recognition and classification to fit curves and analyse regressions using the various concepts of neural networks.

The input data is represented as q input vectors of a column matrix. The target q vectors are stored in another matrix.

The procedure requires us to select Training data required to adjust the network error with time (which we set to 70%), Validating data required to generalize data and stop training (which we set to 15%) and Testing data which gives an independent network performance metric (which we set to 15%).

Here we create the ANN with 12 inputs and 1 inputs. In the middle we have presented 10 hidden layers (neurons) and 1 output layer.



Here we use Bayesian Regularization to train the ANN which minimizes the mean squared error and weight of every step, and then determines the correct combination and produces a network which generalizes smoothly.

This technique is used as it is effective for small datasets with noise and difficult outliers.

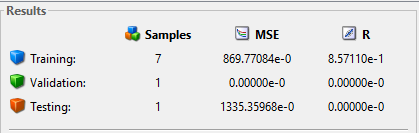
**4.Results**

The 2 ways by which we have analysed the performance of the ANN:

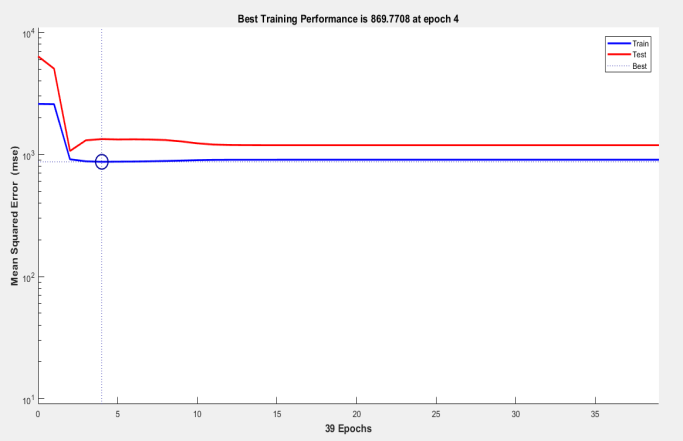
Mean Squared Error which gives the average squared difference between inputs and targets.

Regression values which gives the correlation between the input and targets.

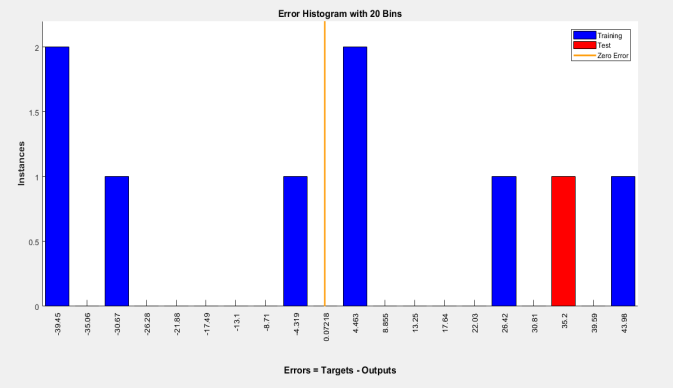
After training the data we get

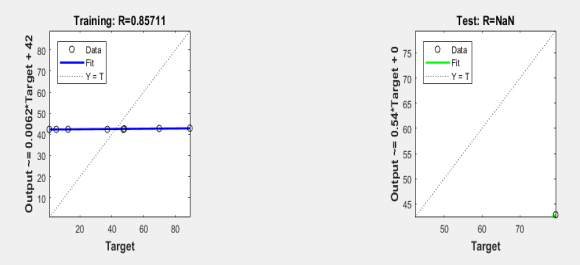


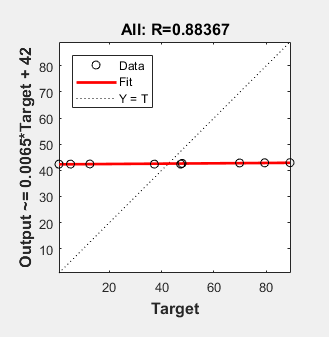
After 39 iterations the best performance graph comes out as



The error histogram quantifies the averaged differences of targets and inputs.



The regression analysis gives the following results



**5.Conclusion**

By creating and testing the ANNformed by the parameters defined from the weather dataset of Vellore acquired over the internet we can conclude the following:

Since the regression coefficient between the 9 parameters of the 12 months data and the annual average data is so high, we can say that the weather of Vellore area shows minimal change over the whole year.

The only 3 outlying parameters which have a weak regression value are Precipitation, Days with precipitation and Cloud Cover. This means that the only seasonal change occurs during the monsoon months.

We have used back propagation to arrive at a satisfactory result by eliminating errors inductively.

**6.Reference**

[1] T. Hall, H.E. Brooks, and C.A. Doswell III (Jun. 1999), “Precipitation forecasting using aneural network,” *Weather and Forecasting*, vol. 14, no. 3, pp. 338-345.

[2] R.J. Kugliowski and A.P. Barros (Dec. 1998), “Localized precipitation forecasts from anumerical weather prediction model using artificial neural networks,” *Weather and*

*Forecasting*, vol. 13, no. 4, pp. 1194-1204.

[3] J. Harris (Dec. 2011), “Neural Networks,” class powerpoint slides for CSCI 7130, ComputerScience Department, Georgia Southern University.

[4] J. Heaton (2010), *Introduction to Encog 2.5: Revision 3-November*, Heaton Research, Inc.

[5] S. Russell and P. Norvig(2010), *Artificial Intelligence: A Modern Approach, ThirdEdition*, Prentice Hall.

[6] I.H. Witten and E. Frank (2005), *Data Mining: Practical Machine Learning Tools andTechniques, Second Edition*, Morgan Kaufmann Publishers.

[7] J. Kennedy and R.C. Eberhart (1995), “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948.