6

Data Mining Based on Neural Networks for Gridded Rainfall Forecasting

Kavita Pabreja
Birla Institute of Technology and Science,
Pilani, Rajasthan,
Maharaja Surajmal Institute (GGSIP University),
New Delhi,
India

1. Introduction

The application of neural networks in the data mining has become wider. Although neural networks may have complex structure and long training time but they have high acceptance ability for noisy data and high accuracy. Artificial neural network (ANN), has emerged during last decade as an analysis and forecasting tool in the field of weather. In this chapter the data mining based on neural networks has been used to forecast daily rainfall over Indian region. ANN has been trained for forecasting the rainfall of current year based on previous year's rainfall for the months of June to September. The ANN hence trained has demonstrated promising results.

2. Literature review

ANN has been applied in a few weather forecasting cases in the past. A neural network, using input from the Eta Model and upper air soundings, has been developed by Hall et al.,1999 for the probability of precipitation (PoP) and quantitative precipitation forecast (QPF) for the Dallas–Fort Worth, Texas, area. Forecasts from two years were verified against a network of 36 rain gauges. The resulting forecasts were remarkably sharp, with over 70% of the PoP forecasts being less than 5% or greater than 95%.

A neuro-fuzzy system has been used for rainfall forecasting using data from 1893-1933 as training set and 1934-1980 as test set. ANN has shown outstanding forecasting performance in many other weather related forecasts (Hayati & Mohebi, 2007; Chattopadhyay, 2007; Paras et al., 2007; Collins & Tissot, 2008). In this chapter, we have tried to forecast rainfall based on only the latitude and longitude of previous year's rainfall datasets and the results were found to be very convincing.

3. About artificial neural network

An ANN is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an

interconnected group of artificial neurons, and it processes information using a connectionist approach to computation (Sivanandam et al., 2009). In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. For the configuration, there are network functions used for training and testing of the network, as explained in following sections.

3.1 Network function

The word '*network*'' refers to the inter–connections between the neurons in the different layers of each system. The most basic system has three layers. The first layer has input neurons which send data via synapses to the second layer of neurons and then via more synapses to the third layer of output neurons. More complex systems have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" which are used to manipulate the data in the calculations.

The layers network through the mathematics of the system algorithms. The network function $f(x)$ is defined as a composition of other functions $g_i(x)$, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables, as shown in Fig. 1.

3.2 Training and testing the network

In an Artificial Neural Network, the system parameters are changed during operation, normally called the training phase. After the training phase, the Artificial Neural Network parameters are fixed and the system is deployed to solve the problem at hand (the testing phase). The Artificial Neural Network is built with a systematic step-by-step procedure to optimize a performance criterion or to follow some implicit internal constraint, which is commonly referred to as the learning rule (Kosko, 2005). The input/output training data are fundamental in neural network technology, because they convey the necessary information to "discover" the optimal operating point. The nonlinear nature of the neural network processing elements (PEs) provides the system with lots of flexibility to achieve practically any desired input/output map, i.e., some Artificial Neural Networks are universal mappers.

An input is presented to the neural network and a corresponding desired or target response set at the output (when this is the case the training is called supervised). An error is composed from the difference between the desired response and the system output. This error information is fed back to the system and adjusts the system parameters in a systematic fashion (the learning rule). The process is repeated until the performance is acceptable. It is clear from this description that the performance hinges heavily on the data. If one does not have data that cover a significant portion of the operating conditions or if

they are noisy, then neural network technology is probably not the right solution. On the other hand, if there is plenty of data and the problem is poorly understood to derive an approximate model, then neural network technology is a good choice. In artificial neural networks, the designer chooses the network topology, the performance function, the learning rule, and the criterion to stop the training phase, but the system automatically adjusts the parameters. So, it is difficult to bring a priori information into the design, and when the system does not work properly it is also hard to incrementally refine the solution. But ANN-based solutions are extremely efficient in terms of development time and resources, and in many difficult problems artificial neural networks provide performance that is difficult to match with other technologies.

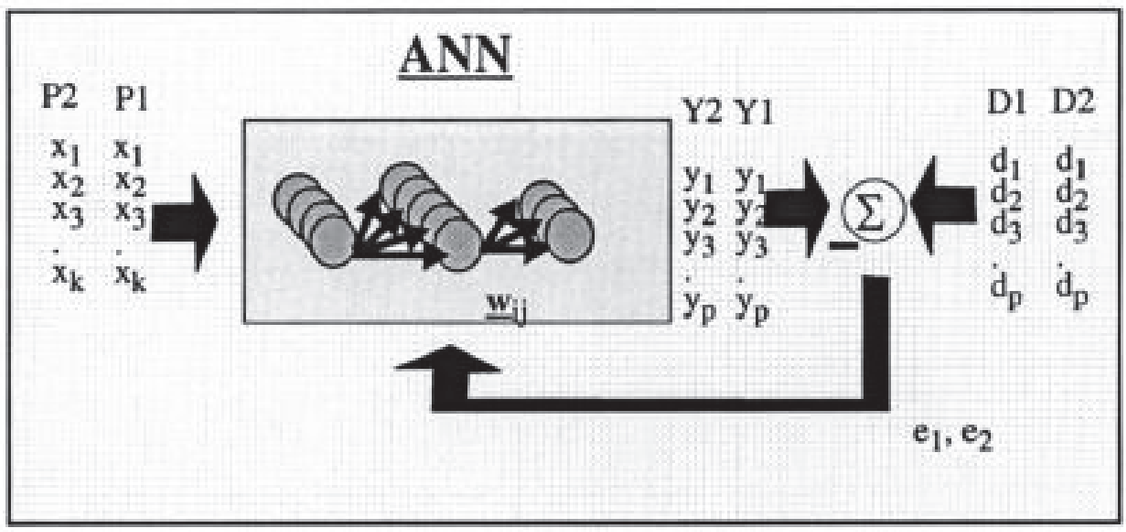


Fig. 1. Style of neural computation

3.3 MLP back propagation network

This is the most common neural network model, also known as supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using the historical data so that the model then can be used to produce the output when the desired output is unknown.

In this network, shown in Fig. 2, the input data are fed to input nodes and then they will pass to the hidden nodes after multiplying by a weight. A hidden layer adds up the weighted input received from the input nodes, associates it with the bias and then passes the result on through a nonlinear transfer function. The output node does the same operation as that of a hidden layer. This type of network is preferred as back propagation learning is a popular algorithm to adjust the interconnection weights during training, based upon the generalized delta rule proposed (Kosko, 2005).

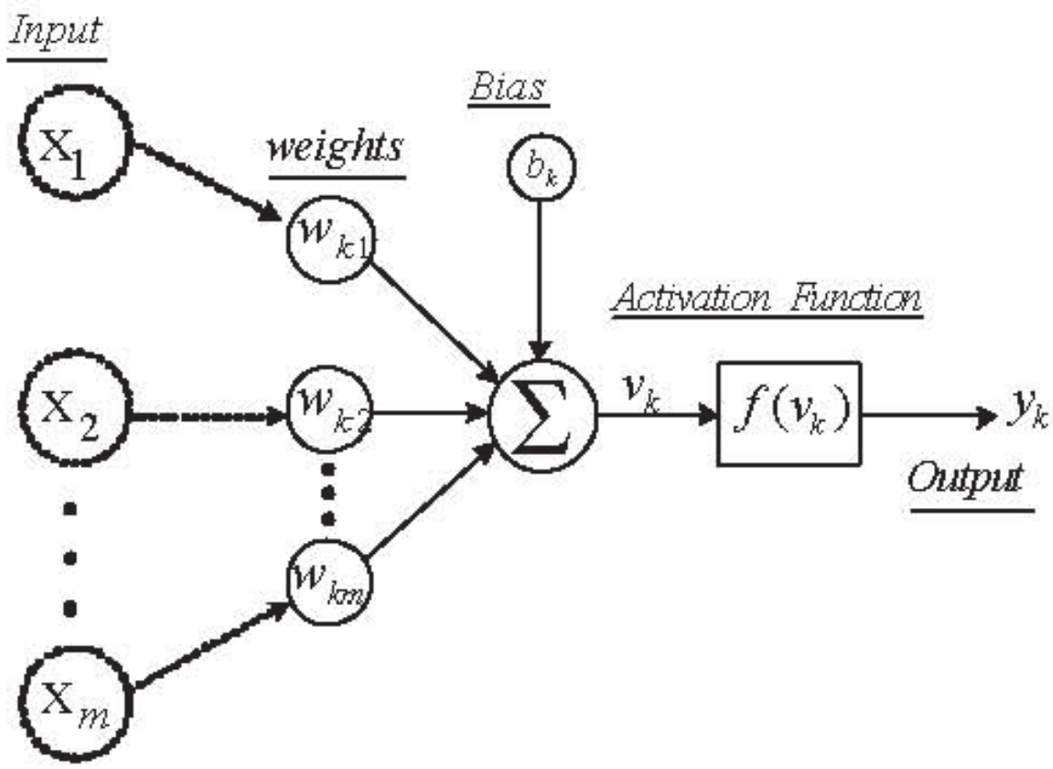


Fig. 2. Neuron Model

4. Case study of rainfall forecasting

4.1 Datasets used

A very high resolution (0.5° × 0.5°) daily rainfall (in mm) dataset for mesoscale meteorological studies over the Indian region has been provided by Indian Meteorological Department (IMD) and described by Rajeevan & Bhate(2009). The dataset is in .grd format, a control file describing the structure of .grd file has been provided. There is one .grd file for each year of rainfall.

This dataset consists of daily rainfall data for each year for the period 1984–2003. The data is for the geographical region from longitude 66.5 ^{o}E to 100.5 ^{o}E and latitude 6.5 ^{o}N to 38.5 ^{o}N for each day of the year. There are 4485 grid points readings every day and rainfall record for 122 days (June to September) per year are selected for analysis i.e 5,47,170 records out of a total of 16,37,025 records for one year of rainfall.

4.2 Data re-processing

Steps followed for pre-processing of the .grd so that the ANN can be trained and tested, are mentioned below:

1. The .grd file has been converted to .dat file using a FORTRAN (Formula Translator) programme. This dataset is very huge in size.
2. The .txt files have been exported to Excel worksheet and then to Access database. The data looks like as if a rectangular grid is filled with values of rainfall in mm.(a sample of year 1989 rainfall is shown in table 1).
3. A programme is written in Visual Basic so as to organize data in tabular format with rainfall mentioned at every grid point on each day, as shown in table 2.
4. Finally exporting the dataset into .xls format for analysis, by Matlab (Matrix Laboratory).

The daily rainfall dataset taken into consideration for the training of Neural Network is from longitude 70.5 °E to 90.0 °E and latitude 17.5°N to 37.0°N for the time period June to September for the years 1989 to 1992 as the focus is on Indian subcontinent only.

Longitude (°E)

Latitude(°N)	75	75.5	76	76.5	77	77.5	78	78.5	79	79.5
38.5	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
38	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
37.5	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
37	-999	-999	-999	-999	0	0	0	-999	-999	-999
36.5	-999	-999	-999	-999	4.4	1.2	0	-999	-999	-999
36	-999	-999	-999	9.8	8	1.5	0	1	-999	-999
35.5	-999	-999	-999	6.4	6.5	1	7.3	2.8	-999	-999
35	-999	-999	7.7	7.4	24.2	11.7	17.3	24.2	2	-999
34.5	-999	-999	7.1	10.7	6.2	7.1	27.8	16.2	10.6	5.8
34	-999	-999	0.4	16	0.3	1.8	0.8	7.7	7.9	0
33.5	-999	45.5	0	23.5	37.3	5.2	13.6	1.8	0	0
33	10.6	0.3	15	27.9	17	0	0	0	4.7	1.1
32.5	0	0	1.8	31.2	8.3	5.9	0.3	7.7	3.6	3.7
32	0	14.4	24.9	9.3	0	0	0.4	0	2.7	0
31.5	0	1.6	0.2	3.4	0	1.5	0	0	2.2	0
31	0	0	6	1.7	0	0	0	0	5	1.6
30.5	2.4	2.2	4.2	0	0	0	1.6	0	0	0

(Source: as a result of pre-processing rf1989.grd provided by IMD)

Table 1. Text file retrieved from .grd file for rainfall in 1989

Table1989					
S.No.	**Day#**	**Date**	**Latitude (°N)**	**Longitude (°E)**	**Rainfall (in mm)**
404	1	01-Jun-89	34.5	76	7.1
405	1	01-Jun-89	34.5	76.5	10.7
406	1	01-Jun-89	34.5	77	6.2
407	1	01-Jun-89	34.5	77.5	7.1
408	1	01-Jun-89	34.5	78	27.8
409	1	01-Jun-89	34.5	78.5	16.2
410	1	01-Jun-89	34.5	79	10.6
411	1	01-Jun-89	34.5	79.5	5.8

(Source: as a result of pre-processing rf1989.grd provided by IMD)

Table 2. Rainfall for year 1989 organized in tabular format

4.3 Technique used

ANN in this study was trained and simulated using Matlab 7.0 (matrix laboratory) designed and developed by Math Works Inc. For the training and testing of network, a two layer MLP

Day no.	Latitude	Longitude	Rainfall
1	35.5	76.5	6.4
1	35.5	77	6.5
1	35.5	77.5	1
1	35.5	78	7.3
1	35.5	78.5	2.8
1	35	76	7.7
1	35	76.5	7.4
1	35	77	24.2
1	35	77.5	11.7
1	35	78	17.3
1	35	78.5	24.2
1	35	79	2
1	34.5	76	7.1
1	34.5	76.5	10.7
1	34.5	77	6.2
1	34.5	77.5	7.1

(Source: as a result of pre-processing rf1989.grd provided by IMD)

Table 3. Sample of location-wise rainfall for year 1989

Back Propagation network has been used. The input dataset comprises of daynumber (day 1 corresponds to June 1, day 2 to June 2 and so on till day number 122 that corresponds to September 30), latitude and longitude. The output data corresponds to rainfall in mm. A sample of dataset is shown in table 3. From this table, columns 1 to 3 are used as input and column 4 is used as target.

Before training, the inputs and outputs have been scaled so that they fall in the range[-1,1]. The following code has been used at Matlab prompt:-

[pn1992,minp,maxp,tn1992,mint,maxt]=premnmx(linput,loutput)

The original network inputs and targets are given in the matrices linput and loutput. The normalized inputs and targets, pn1992 and tn1992, that are returned, will all fall in the interval [-1,1]. The vectors minp and maxp contain the minimum and maximum values of the original inputs, and the vectors mint and maxt contain the minimum and maximum values of the original targets.

4.4 Methodology

Different transfer functions for hidden and output layers were used to find the best ANN structure for this study. Transfer function used in hidden layer of the back propagation network is tangent-sigmoid while pure linear transfer function is used in output layer.

ANN developed for prediction of rainfall is trained with different learning algorithms, learning rates, and number of neurons in its hidden layer. The aim is to create a network which gives an optimum result. The network was simulated using 3 different Back propagation learning algorithms. They are Resilient Backpropagation (*trainrp*), Fletcher-Reeves Conjugate Gradient (*traincgf*) and Scale Conjugate Gradient (*trainscg*).

The Resilient Back propagation (*trainrp)* eliminates the effect of gradient with small magnitude. As magnitudes of the derivative have no effect on the weight update, only the sign of the derivative is used to determine the direction of the weight update. *Trainrp* is generally much faster than standard steepest descent algorithms, and require only a modest increase in memory requirements which suits network with sigmoidal transfer function.

Fletcher-Reeves Conjugate Gradient (*traincgf*) generally converges in fewer iteration than *trainrp,* although there is more computation required in each iteration. The conjugate gradient algorithms are usually much faster than variable learning rate back propagation, and are sometimes faster than *trainrp*. *Traincgf* also require only a little more storage than simpler algorithms, thus they are often a good choice for networks with a large number of weights.

The third algorithm, Scale Conjugate Gradient (*trainscg*) was designed to avoid the time-consuming line search. This differs from other conjugate gradient algorithm which requires a line search at each iteration. The *trainscg* routine may require more iteration to converge, but the number of computations in each iteration is significantly reduced because no line search is performed. *Trainscg* require modest storage.

4.5 Results

Daily rainfall data for 122 days in a year i.e. months June to September were chosen for training and testing. Networks were trained with data of year 1989 and tested using rainfall data of the year 1990. The training has been done using three different training functions as mentioned before: traincgf, trainrp and trainscg. Fig. 3 to Fig. 5 demonstrate the result of training with year 1989 dataset and testing with year 1990 datasets. The results are convincing and the network once trained has been tested with year 1990 datasets and the error comes out to be less than 0.005 in 5 epochs for training functions trainscg and traincgf. With trainrp function, it takes 35 iterations to train.

Another rainfall dataset is for the year 1991 and 1992, training with 1991 and testing with 1992. Fig. 6 to Fig. 8 demonstrate the result of training with year 1991 dataset and testing with year 1992 datasets. Here again, the results are convincing and the network once trained has been tested with year 1992 datasets and the error comes out to be less than 0.005 in 3 epochs for training functions trainscg and traincgf. With trainrp function, it takes 13 iterations to train.

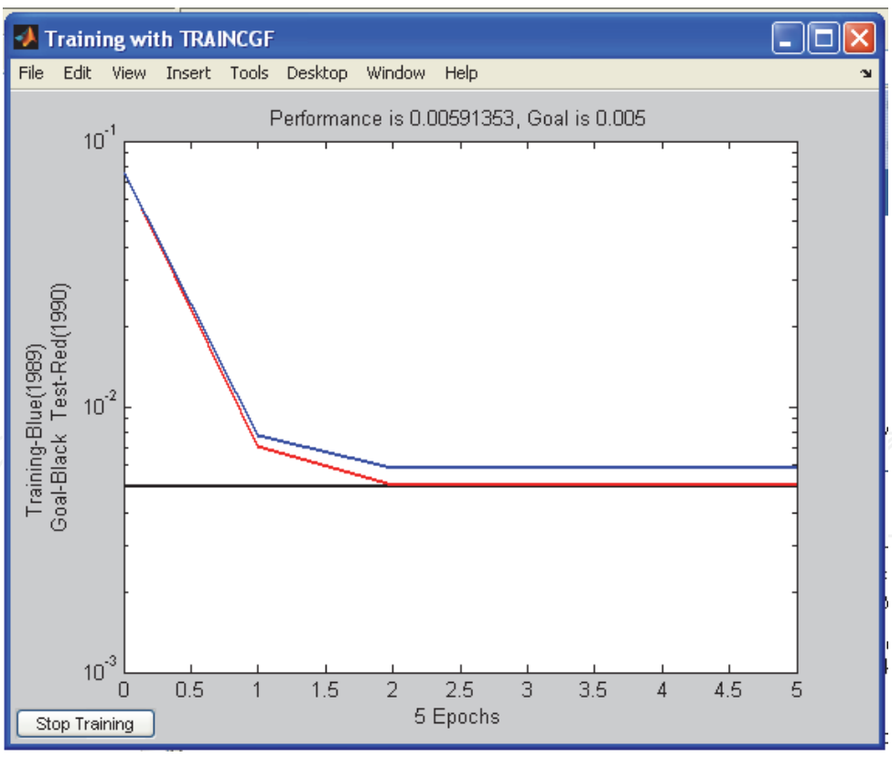


Fig. 3. Result of training ANN with Rainfall data of year 1989 and testing with Rainfall data of year 1990 using learning function traincgf

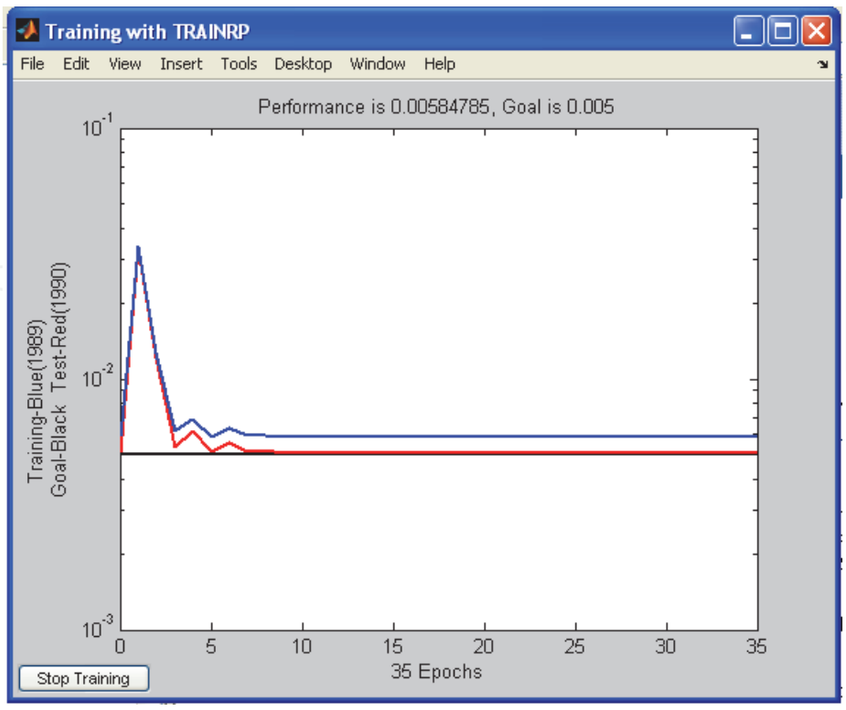


Fig. 4. Result of training ANN with Rainfall data of year 1989 and testing with Rainfall data of year 1990 using learning function trainrp

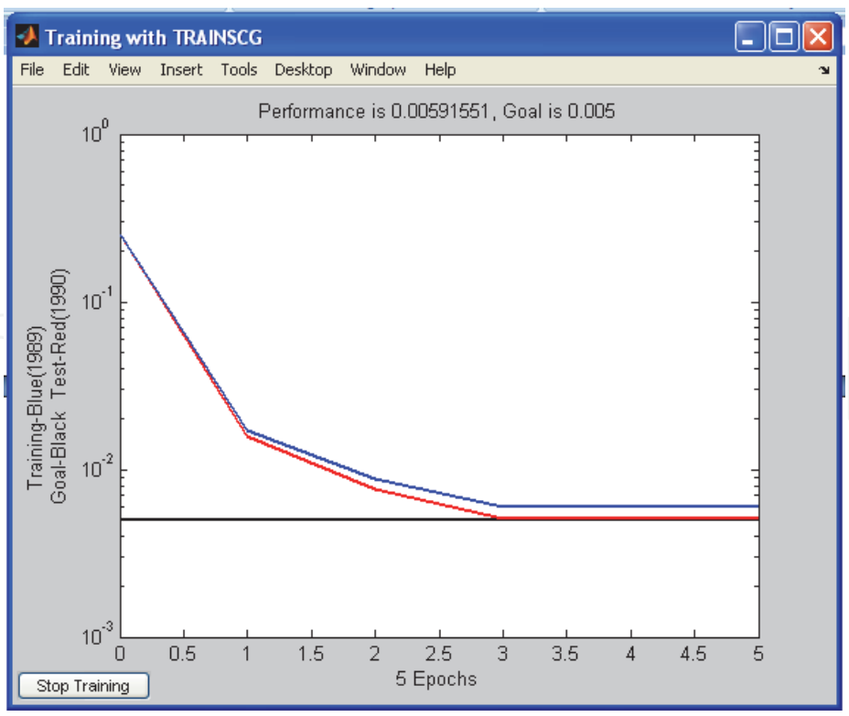


Fig. 5. Result of training ANN with Rainfall data of year 1989 and testing with Rainfall data of year 1990 using learning function trainscg

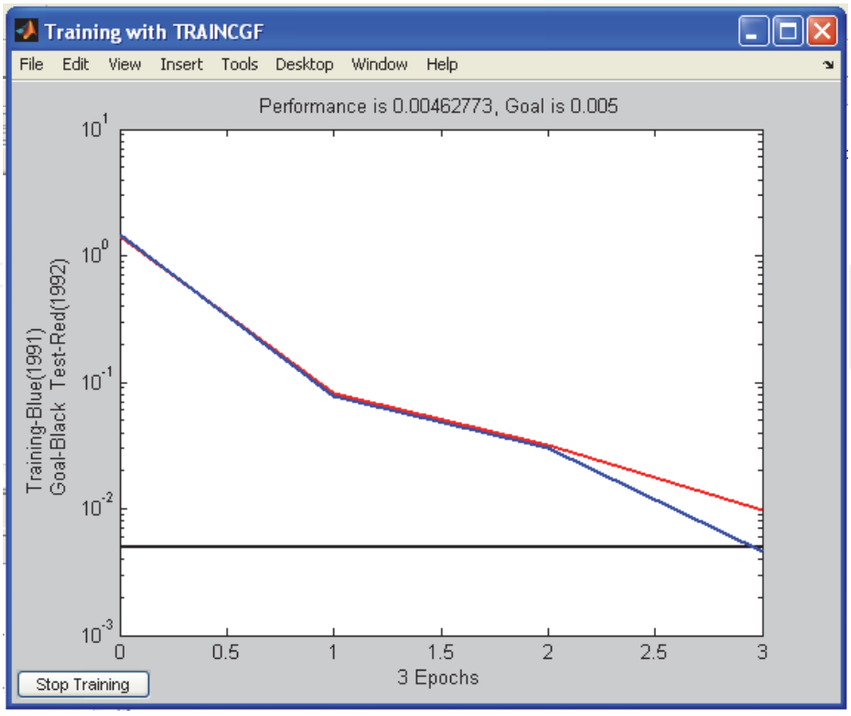


Fig. 6. Result of training ANN with Rainfall data of year 1991 and testing with Rainfall data of year 1992 using learning function traincgf

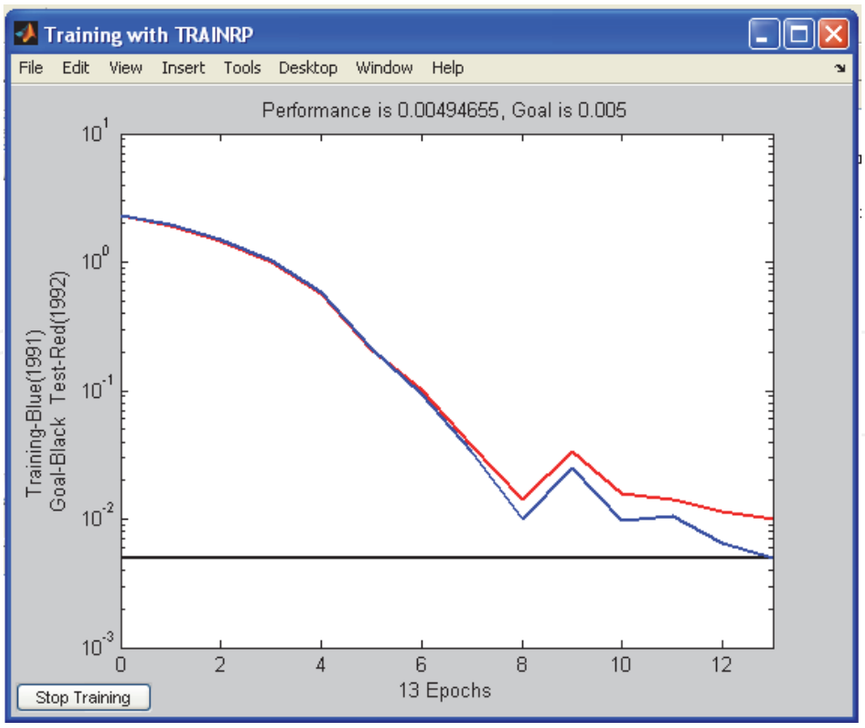


Fig. 7. Result of training ANN with Rainfall data of year 1991 and testing with Rainfall data of year 1992 using learning function trainrp

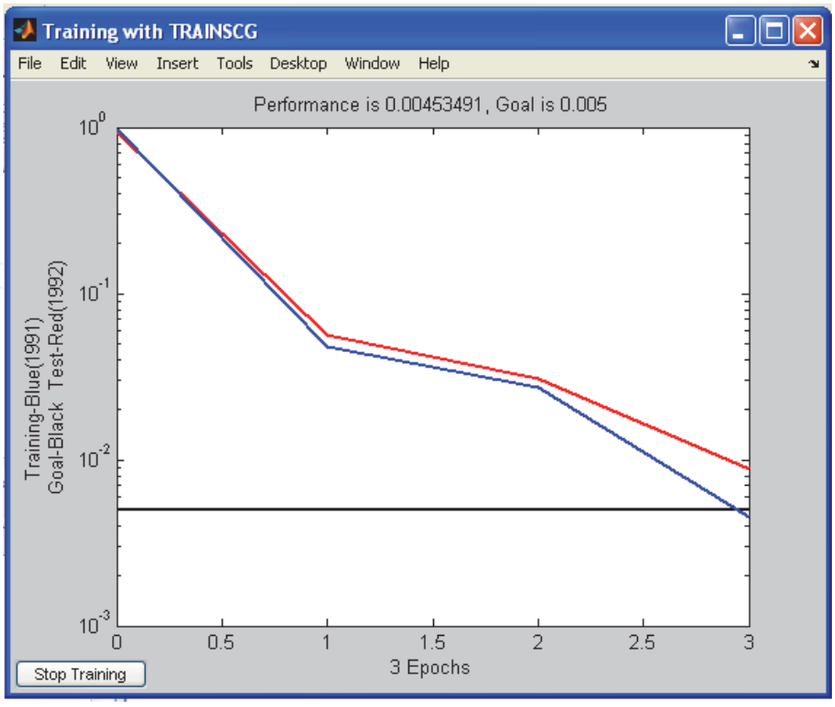


Fig. 8. Result of training ANN with Rainfall data of year 1991 and testing with Rainfall data of year 1992 using learning function trainscg

5. Conclusion

It is concluded that ANN has demonstrated promising results and is very suitable for solving the problem of rainfall forecasting. Using only the input parameters as gridded location, the ANN has been trained to predict Rainfall. This study has clearly brought out that Data Mining techniques when applied rigorously can help in providing advance information for forecast of sub-grid phenomenon.

6. Acknowledgement

This study is based on the datasets made available by courtesy of Indian Meteorological Department, India. The author is thankful for the support extended by IMD. Also, the author thanks Dr. Rattan K. Datta, Former Advisor – Deptt. of Science & Technology, Former President - Indian Meteorological Society and Computer Society of India, for his motivation and guidance.

7. References

Chattopadhyay S.(2007). Multilayered feed forward Artificial Neural Network model to predict the average summer-monsoon rainfall in India, *Journal Acta Geophysica*, Vol. 55, No.3, 2007, pp. 369-382.

Collins W., Tissot P.(2008). Use of an artificial neural network to forecast thunderstorm location, *Proceedings of the Fifth Conference on Artificial Intelligence Applications to Environmental Science,* Published in Journal of AMS., San Antonio, TX, January, 2008.

Hall T., Brooks H.E., Doswell C.A.(1999). Precipitation Forecasting Using a Neural Network, *Weather and Forecasting*, Vol. 14, 1999, pp.338-345.

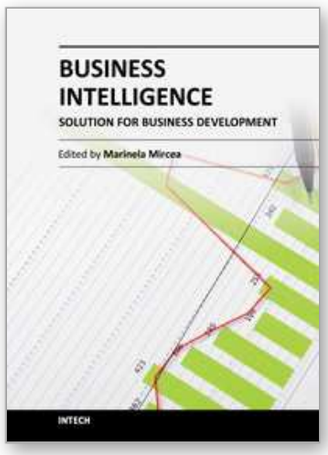
Hayati M., Mohebi Z.(2007). Temperature forecasting based on neural network approach, *World Applied Sciences Journal*. Vol. 2, No. 6, 2007, pp. 613-620.

Kosko B.(2005). *Neural Networks and Fuzzy Systems*, Prentice Hall of India Ltd., 2005.

Paras, Mathur S., Kumar A., Chandra M. (2007). A Feature Based Neural Network Model for Weather Forecasting. *World Academy of Science, Engineering and Technology,* Vol. 34, 2007, pp. 66-73.

Rajeevan M., Bhate J.(2009). A high resolution daily gridded rainfall dataset (1971–2005) for mesoscale meteorological studies, *Current Science*, Vol. 96, No. 4, February 2009.

Sivanandam S.N., Sumathi S., Deepa S.N.(2009). *Introduction to Neural Networks using Matlab*, Tata McGraw Hill Education Private Ltd., 2009.



Business Intelligence - Solution for Business Development
Edited by Dr. Marinela Mircea

ISBN 978-953-51-0019-5
Hard cover, 108 pages
Publisher InTech
Published online 01, February, 2012
Published in print edition February, 2012

The work addresses to specialists in informatics, with preoccupations in development of Business Intelligence systems, and also to beneficiaries of such systems, constituting an important scientific contribution. Experts in the field contribute with new ideas and concepts regarding the development of Business Intelligence applications and their adoption in organizations. This book presents both an overview of Business Intelligence and an in-depth analysis of current applications and future directions for this technology. The book covers a large area, including methods, concepts, and case studies related to: constructing an enterprise business intelligence maturity model, developing an agile architecture framework that leverages the strengths of business intelligence, decision management and service orientation, adding semantics to Business Intelligence, towards business intelligence over unified structured and unstructured data using XML, density-based clustering and anomaly detection, data mining based on neural networks.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen