

# Identification of the best hidden layer size for three-layered neural net in predicting monsoon rainfall in India

Surajit Chattopadhyay and Goutami Chattopadhyay

## ABSTRACT

In the present research, long-range prediction of average summer monsoon rainfall over India has been attempted through three layered artificial neural network models. The study is based on the summer monsoon data pertaining to the years 1871–1999. Nineteen neural network models have been developed with variable hidden layer size. Total rainfall amounts in the summer monsoon months of a given year have been used as input and the average summer monsoon rainfall of the following year has been used as the desired output to execute a supervised backpropagation learning procedure. After a thorough training and test procedure, a neural network with eleven nodes in the hidden layer is found to be the most proficient in forecasting the average summer monsoon rainfall of a given year with the said predictors. Finally, the performance of the eleven-hidden-nodes three-layered neural network has been compared with the performance of the asymptotic regression technique. Ultimately it has been established that the eleven-hidden-nodes three-layered neural network has more efficacy than asymptotic regression in the present forecasting task.

**Key words** | artificial neural network, asymptotic regression, hidden nodes, India, prediction, summer monsoon rainfall

**Surajit Chattopadhyay** (corresponding author)  
Department of Information Technology,  
Pailan College of Management and Technology,  
Bengal Pailan Park, Kolkata 700104,  
West Bengal,  
India  
Postal address: 1/19 Dover Place,  
Kolkata 700019, West Bengal  
India  
E-mail: surajit\_2008@yahoo.co.in

**Goutami Chattopadhyay**  
Department of Atmospheric Sciences,  
University of Calcutta,  
Kolkata 700019, West Bengal,  
India  
Current address: 1/19 Dover Place,  
Kolkata 700019, India

## INTRODUCTION

Stochastic weather models are statistical models whose output bears a resemblance to the weather data to which they have been fitted (Wilks 1999). Such models are available for various hydrological systems (e.g. Williams *et al.* 1985; Pickering *et al.* 1988). The stochastic approach elucidates the nonlinearity in the data and calls for prior assumptions concerning the data distribution (Milionis & Davis 1994). Several authors have recognized the chaotic behavior of atmosphere (e.g. Elsner & Tsonis 1992; Men *et al.* 2004; Varotsos & Cracknell 2004; Varotsos 2005; Varotsos *et al.* 2005; Varotsos & Kirk-Davidoff 2006). Thus, no prior assumptions should be made while dealing with chaotic atmospheric processes (Wilks 1991). Alternative mathematical methods are, therefore, required in developing models for atmospheric phenomena. In the last few decades, artificial neural networks (ANN) have

emerged as an alternative mathematical tool for understanding atmospheric processes. Hu (1964) initiated the implementation of ANN methodology in weather forecasting. Unlike the stochastic modeling techniques, artificial neural networks make no prior assumptions concerning the data distribution and they are capable of modeling highly nonlinear relationships and can be trained to accurately generalize when presented with a new dataset (Nagendra & Khare 2006). The ANNs are parallel computational models, comprised of densely organized adaptive processing units. The vital characteristic of neural networks is their adaptive nature that makes the ANN techniques very alluring in application domains for solving problems where the internal physical processes are highly complex and nonlinear (Gardner & Dorling 1998; Nagendra & Khare 2006). Over the last few decades, ANNs have opened up new

avenues to the forecasting task involving geophysical phenomena (Gardner & Dorling 1998; Hsieh & Tang 1998). The present study being an application of ANN to rainfall forecasting, the authors of this paper have gone through scores of noteworthy papers where ANN has been applied to rainfall time series from different points of view. Some of them are mentioned here. Zhang & Scofield (1994) applied ANN to estimate rainfall and to recognize cloud merger from satellite data. Michaelides *et al.* (1995) compared the performance of ANN with multiple linear regressions in estimating missing rainfall data over Cyprus. Kalogirou *et al.* (1997) implemented ANN to reconstruct the rainfall time series over Cyprus. Lee *et al.* (1998) applied Artificial Neural Networks in rainfall prediction by splitting the available data into homogeneous subpopulations. Wong *et al.* (1999) constructed fuzzy rule bases with the aid of self-organizing maps (SOM) and backpropagation neural networks and then developed a predictive model with the help of the rule base for rainfall over Switzerland using spatial interpolation. Chaotic behaviour of rainfall has been investigated by Sivakumar *et al.* (1999) and Sivakumar (2000, 2001).

The genesis of rainfall is a complex phenomenon. It has a close association with several other atmospheric variables. A plethora of evidence based on observational and modeling studies has established a strong relationship between changes in the slowly varying boundary conditions at the Earth's surface (e.g. sea surface temperature, land surface conditions) (Xue & Shukla 1998). The anthropogenic activity plays an important role in the gradual building up of atmospheric pollution, which is strongly associated with precipitation (Cartalis & Varotsos 1994; Jacovides *et al.* 1994; Kondratyev & Varotsos 2001a, b; Varotsos *et al.* 2001). For instance, aerosol particles reduce the penetration of solar radiation to the surface, suppressing precipitation. In addition, monsoon acts as a dynamical mechanism for transporting atmospheric pollutants between geographical regions (i.e. from Asia to the Mediterranean region). Several stochastic models have been attempted to forecast the occurrence of rainfall, to investigate its seasonal variability, to forecast monthly/yearly rainfall over some given geographical areas. Daily precipitation occurrence has been viewed through Markov chains by Chin (1977). Gregory *et al.* (1993) applied a chain-dependent stochastic model, called a

Markov chain model, to investigate inter-annual variability of area average total precipitation. Wilks (1998) applied mixed exponential distribution to simulate precipitation amount at multiple sites exhibiting realistic spatial correlation.

The Indian economy is primarily based on agriculture, and agricultural processes are heavily dependent upon rainfall in the summer monsoon period. So prediction of the Indian summer monsoon rainfall is quite imperative for successful agricultural practices over this country. Several methods have been adopted to date to forecast the summer monsoon rainfall over India. Hastenrath (1988) developed a statistical model using the regression method to predict the Indian summer monsoon rainfall anomaly. Rajeevan (2001) discussed the problems and prospects for the prediction of the Indian summer monsoon and revealed that Indian summer monsoon predictability exhibits epochal variations. Gadgil *et al.* (2005) investigated the causes of failure in the prediction of the Indian summer monsoon and expected the dynamical models to generate better prediction only after the problem of simulating year-to-year variation of the monsoon is addressed. Kishtawal *et al.* (2003) assessed the feasibility of a nonlinear technique based on a genetic algorithm, an Artificial Intelligence technique for the prediction of summer rainfall over India. Guhathakurta (2006) implemented the ANN technique to predict rainfall over a state (Kerala) of India, but he confined his study to within this state. The present contribution deviates from the study of Guhathakurta (2006) in the sense that, instead of choosing a particular state, the authors implement back-propagation ANN to forecast the average summer monsoon rainfall over the whole country and an aroma of newness further lies in the fact that here various multilayer ANN models are attempted to find out the best fit.

In an earlier work (Chattopadhyay 2007) the first author of the present paper developed an ANN in the form of a multilayer perceptron model to forecast the summer monsoon rainfall with some other meteorological parameters and indices as predictors. The aroma of newness in the present study lies in the fact that, instead of developing an ANN model with a predetermined ANN architecture, it has tested several possible ANN models with variable hidden layer sizes. All the models have been tested rigorously using statistical procedures. Another newness in

the approach is that, instead of using different meteorological parameters as predictors, it has used the summer monsoon rainfall data of past years as predictors. The authors have attempted this in order to get rid of the task of collecting several other relevant meteorological parameters and to develop a model that can be simpler than other complicated numerical weather prediction models. The ANN models developed in this paper are based on supervised backpropagation learning where the “desired output” would be the average summer monsoon rainfall of a given year and the learning would aim to minimize the difference between model output and the actual values of the desired output. The input matrix would consist of the rainfall amounts of the summer monsoon months of a particular year. The details of the implementation procedure are described in the subsequent sections.

## MATERIALS AND METHODS OF DEVELOPING THE PREDICTIVE MODEL

The ANNs have recently become an important alternative tool to conventional methods in modeling complex nonlinear relationships. In the recent past, ANNs have been applied to model large data with large dimensionality (i.e. Gevrey *et al.* 2003; Nagendra & Khare 2006). Most of the ANN studies spoke to the problem allied with pattern recognition, forecasting and comparison of the neural network with other traditional approaches in ecological and atmospheric sciences. However, the step-by-step procedure involved in the development of ANN-based models is less discussed (Nagendra & Khare 2006). This paper develops an ANN model step-by-step to predict the average rainfall over India during the summer monsoon by exploring the data available at the website <http://www.tropmet.res.in> run by the Indian Institute of Tropical Meteorology.

The ANN approach has several advantages over conventional phenomenological or semi-empirical models, since they require a known input dataset without any assumptions (Gardner & Dorling 1998; Nagendra & Khare 2006). It exhibits rapid information processing and is able to develop a mapping of the input and output variables. Such a mapping can subsequently be used to predict desired outputs as a

function of suitable inputs (Nagendra & Khare 2006). A multilayer neural network can approximate any smooth, measurable function between input and output vectors by selecting a suitable set of connecting weights and transfer functions or activation functions (Kartalopoulos 1996; Gardner & Dorling 1998; Nagendra & Khare 2006).

The model building process consists of four sequential steps:

- (i) selection of the input and output for the supervised backpropagation learning,
- (ii) selection of the activation function,
- (iii) training and testing of the model,
- (iv) testing the goodness of fit of the model.

The advent of the backpropagation algorithm (BP), the adaptation of the steepest descent method, opened up new avenues of application of multilayered ANN for many problems of practical interest (see, for example, Sejnowski & Rosenberg 1987; Kamarthi & Pittner 1999; Perez & Reyes 2001). A multilayer ANN contains three basic types of layer: input layer, hidden layer(s) and output layer. Basically the backpropagation learning involves propagation of error backwards from the output layers to the hidden layers in order to determine the update for the weights leading to the units in the hidden layer(s). The nonlinear relationship between input and output parameters in any network requires a function, known as the activation function, which can appropriately connect and/or relate the corresponding parameters (Nagendra & Khare 2006).

The weight updating in the BP algorithm can be mathematically written as (Trentin and Giuliani 2001)

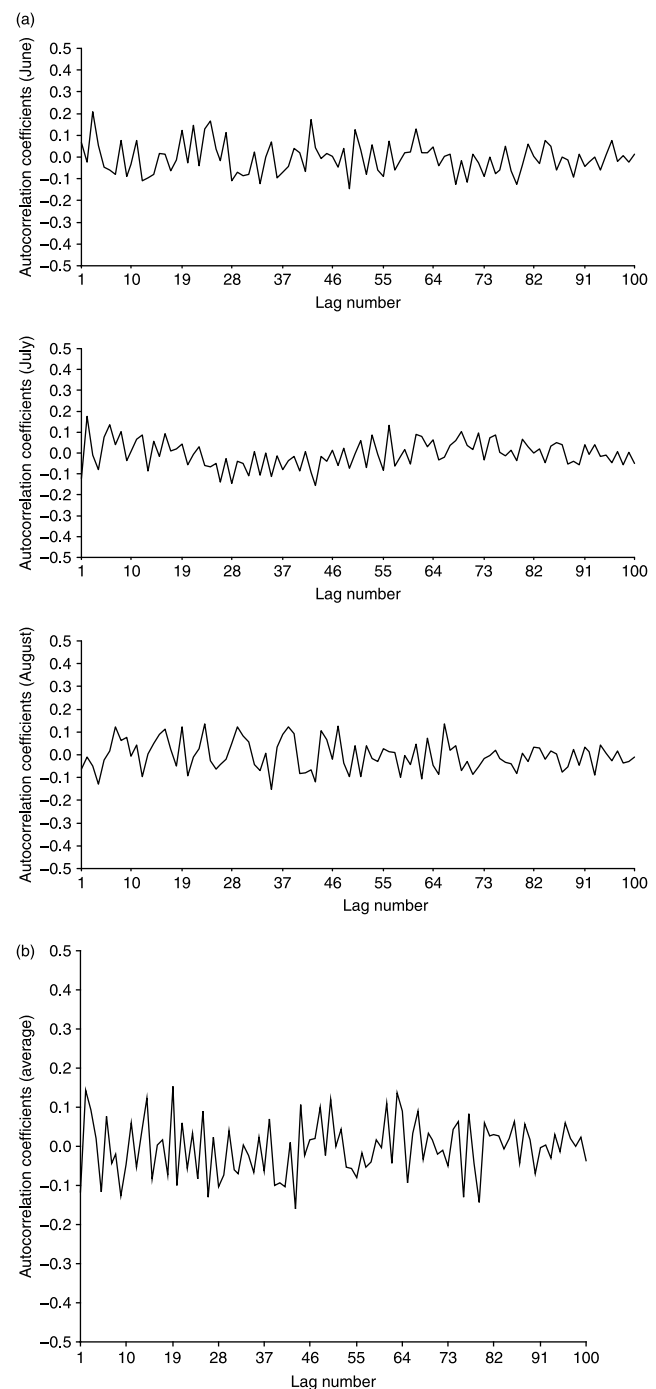
$$\Delta w_{ij}(t+1) = \eta(t+1)\delta_i(t+1)o_j(t+1) + \rho\Delta w_{ij}(t) \quad (1)$$

Equation (1) is used to compute the entity of weight change at step  $(t+1)$ . Here  $\eta(t)$  is the learning rate at time  $t$ ,  $\rho$  is the momentum,  $w_{ij}$  are the connection weights,  $\delta_i(t)$  is the usual delta factor for unit  $i$  as obtained from direct application of the delta rule at time  $t$  and  $o_j(t)$  is the output from the unit  $j$  at the same time. In general, the parameters  $\eta$  and  $\rho$  are called the learning parameters. They are used to speed up or to slow down the convergence of error (Nagendra & Khare 2006).

In India, the months June, July and August are identified as the summer monsoon months. Thus, the present study

explores the data of these three months corresponding to the years 1871–1999. From these 129 years, the last year is deleted because that would not lead to any prediction. Thus, there would be  $(128 \times 3 = 384)$  months in our modeling problem. For each month, there would be a time series of homogenized rainfall data with 128 entries. It is interesting to see that the time series are not pairwise correlated. The mutual Pearson correlation values are  $-0.06$  (June–July),  $-0.01$  (June–August) and  $-0.01$  (July–August). Thus, all the correlation values are too small, indicating that the relationships are not linear. Thus, the necessity of implementing ANN in the prediction problem is felt highly relevant. In the next step, the autocorrelation functions are derived for each summer monsoon month within the study period. The autocorrelation functions are presented in Figure 1(a). This figure shows that all the autocorrelations (computed up to 100 lags) are much below 1 and above  $-1$ . This indicates that the corresponding time series exhibit no persistence. The autocorrelation function of the average summer monsoon rainfall time series is presented in Figure 1(b) with autocorrelation coefficients up to 100 lags. In this case too, the autocorrelation coefficients are at a significant distance from  $\pm 1$ . This again points out that the data displays no serial correlation or persistence. The aim of this paper is to develop a multilayer feedforward ANN model so that the average summer monsoon rainfall of a given year can be predicted using the rainfall data of the summer monsoon months of the immediately previous year. Thus, the input matrix would consist of four columns of which the first three columns would correspond to the summer monsoon months' rainfall of year  $n$  and the fourth column would correspond to the average summer monsoon rainfall of the year  $(n + 1)$ . Basically, the fourth column would correspond to the 'desired output' in the supervised backpropagation learning procedure (Kartalopoulos 1996; Kamarthi & Pittner 1999; Yegnanarayana 2000). The first 75% data (i.e. 96 rows out of 128 rows) are taken as the training set and the remaining 25% data (i.e. 32 rows out of 128 rows) are taken as the test set or validation set.

On-line learning and batch learning are two standard learning schemes for the BP algorithm. In the first type of learning the weights of the network are updated immediately after the presentation of each pair of input and target patterns. In the other learning all the pairs of patterns in the



**Figure 1** | Schematic showing the autocorrelation function corresponding to the yearly average summer monsoon rainfall between 1871 and 1999.

training sets are treated as a batch, and the network is updated after processing of all training patterns in the batch (Kamarthi & Pittner 1999). In either case the vector  $w_k$  contains the weights computed during the  $k$ th iteration and

the output error function  $E$  is a multivariate function of the weights in the network (Kamarthi & Pittner 1999; Chattopadhyay 2007):

$$E(w_k) = \begin{cases} E_p(w_k)[\text{on-line}] \\ \frac{\sum_p E_p(w_k)[\text{batch}]}{p} \end{cases} \quad (2)$$

where  $E_p(w_k)$  denotes the half-sum-of-squares error functions of the network output for a certain input pattern  $p$ . The purpose of the supervised learning (or training) is to find out a set of weights that can minimize the error  $E$  over the complete set of training pairs. Every cycle in which each one of the training patterns is presented once to the neural network is called an epoch.

When the sigmoid function is adopted as the activation function, the BP algorithm becomes 'backpropagation for the sigmoid Adaline' (Widrow & Lehr 1990). In this method the input matrix is multiplied by the weight matrix and the product is used as the variable for the sigmoid activation function. For example, at epoch  $k$ , the sigmoid nonlinearity is produced as (Chattopadhyay 2007)

$$f(W_k X_k) = \frac{1}{1 + e^{-\left(\sum_i w_i x_i\right)}} \quad (3)$$

where  $W_k = [w_1 \ w_2 \dots w_n]$  and  $[X_k = x_1 \ x_2 \dots x_n]^T$  are the weight matrix and the transpose of the input matrix, respectively, at epoch  $k$ .

After training or learning the ANN with the BP algorithm with sigmoid nonlinearity, an ultimate weight matrix is obtained. This weight matrix is applied to another set of independent inputs to examine the efficiency of the model. This phase is called the testing or validation phase.

After developing the model through training and testing, the goodness of fit of the model is examined statistically. The overall prediction error (PE) is measured as (Perez & Reyes 2001)

$$PE = \frac{\langle |y_{\text{predicted}} - y_{\text{actual}}| \rangle}{\langle y_{\text{actual}} \rangle} \times 100 \quad (4)$$

where  $\langle \rangle$  implies the average over the whole test set.

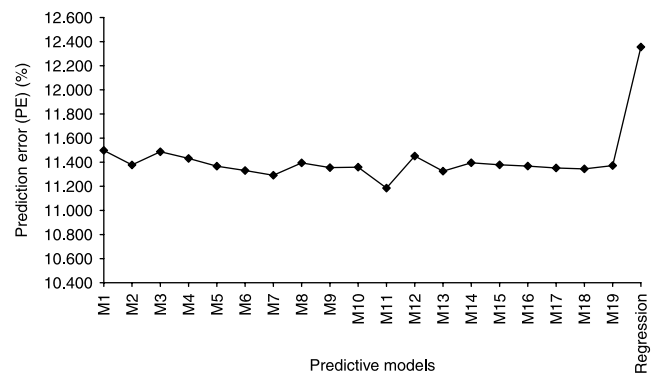
The predictive model is identified as a good one if the  $PE$  is sufficiently small, i.e. close to 0. The model with minimum  $PE$  is identified as the best prediction model.

## IMPLEMENTATION DETAILS AND THE RESULTS

Details of the input and output variables are presented in the previous section. The learning rate  $\eta$  is taken as 0.9 and the momentum is taken as 0.2. Three-layered feedforward ANNs would be designed now. The problem is to find out the number of hidden nodes producing the best model. Since the number of adjustable parameters in a one-hidden-layer feedforward neural network with  $n_i$  input units,  $n_o$  output units and  $n_h$  hidden units is  $[n_o + n_h(n_i + n_o + 1)]$  (Perez *et al.* 2000) for  $n_i = 3$ ,  $n_o = 1$  and 96 training cases, it is not possible to use an  $n_h$  greater than 19.

Now, 19 three-layered feedforward ANN models with  $n_h = 1, 2, 3, \dots, 19$  and  $\eta = 0.9$  are generated. Model  $M_k$  would imply the three-layered feedforward ANN with  $k$  nodes in the hidden layered and trained through on-line backpropagation learning using the methodology explained in the earlier section. In all the 19 models the initial weights are chosen randomly from  $-0.5$  to  $+0.5$  (Pal & Mitra 1999). After each training iteration/epoch the network is tested for its performance on the validation dataset. The training process is stopped when the performance reaches the maximum on the validation dataset (Sarle 1997; Gardner & Dorling 1998; Haykin 2001; Nagendra & Khare 2006).

After training and testing, the  $PE$  (see Equation (8)) values are computed for each model. The results are schematically presented in Figure 2.



**Figure 2** | Schematic showing the prediction error over the whole test set corresponding to 19 different three-layered ANN models and the nonlinear regression model. The figure shows minimum prediction error by  $M_{11}$ .

The result shows that model  $M_{11}$  produces the lowest prediction error among the 19 possible predictive models. After training through 500 epochs the final weight matrix for  $M_{11}$  is found to be

Hdn1_Nrn1	- 1.3427	- 0.1057	- 1.5891	- 0.7324
Hdn1_Nrn2	- 2.0527	- 0.1494	- 0.2275	- 0.4997
Hdn1_Nrn3	- 2.1007	- 0.2049	- 0.1358	- 0.4894
Hdn1_Nrn4	- 0.2157	- 0.1697	- 2.8283	- 1.2286
Hdn1_Nrn5	- 1.1849	- 0.9757	- 0.9198	- 0.7404
Hdn1_Nrn6	- 1.0083	- 0.3428	- 1.4079	- 1.1690
Hdn1_Nrn7	- 1.0306	- 0.7357	- 2.0758	- 0.2146
Hdn1_Nrn8	- 1.7027	- 0.3139	- 0.8909	- 0.3798
Hdn1_Nrn9	- 1.1130	- 0.5681	- 1.0294	- 1.1448
Hdn1_Nrn10	- 0.8666	- 0.9057	- 1.7507	- 0.4722
Hdn1_Nrn11	- 1.6605	- 0.2804	- 0.6678	- 0.6393

Now, the weight matrix presented above would be applied to the test cases to examine the goodness of fit of the model. In Figure 3, the performance of  $M_{11}$  is pictorially presented. This figure shows that in 10 out of 32 test cases the prediction error is below 5%. This means that in 31.25% test cases the absolute prediction error is below 5%. In 14 out of 32 cases, that is, in 46.88%, the absolute prediction error is below 10%. In 23 out of 32 cases, that is, in 71.88% cases, the absolute prediction error is below 15%. Thus, it can be said that if  $\pm 15\%$  error is allowed, then prediction yield is 0.72. In long-range meteorological prediction,  $\pm 15\%$  prediction error is acceptable. Moreover, prediction yield is significant (0.47) when prediction error is  $\pm 10\%$ . Furthermore, the overall prediction error (PE) is small (11.2%). It can therefore be concluded that  $M_{11}$  is an acceptable predictive model for predicting summer monsoon rainfall one year in advance. The components of  $M_{11}$  are presented in Table 1.

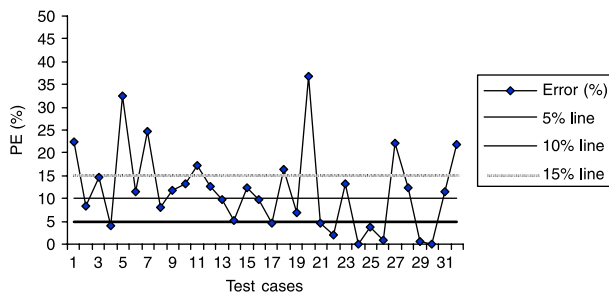


Figure 3 | Schematic showing the absolute prediction errors produced by model  $M_{11}$  in the test cases.

Table 1 | Basic network components of the  $M_{11}$  model

Network architecture			
Number of inputs	3	Number of outputs	1
Number of hidden layers	1	Hidden layer sizes	11
Learning rate	0.9	Initial wt range ( $0 \pm w$ )	0.5
Momentum	0.2		
Training options			
Total no. rows in our data	128	No. of training cycles	500
		Training mode	On line
Save network weights	With least training error		
Training/validation set	Partition data into training/validation set		

## PREDICTION USING NONLINEAR REGRESSION TECHNIQUE

A nonlinear regression equation in the form of asymptotic regression is derived in this section. The regression parameters are estimated using the Levenberg–Marquardt algorithm. The equation is adopted in the form

$$\hat{y}_{\text{predicted}} = b_0 + b_1 \exp(b_2 x) + b_3 \exp(b_4 y) + b_5 \exp(b_6 z) \quad (5)$$

In this equation  $x$ ,  $y$  and  $z$  imply the amounts of rainfall in the months of June, July and August, respectively. On the basis of the same training set as in ANN, the regression parameters  $b_i (i = 1, 2, \dots, 6)$  come out to be 0.1, 0.02, -0.04, 0.02, -0.06 and 0.02, respectively. The regression constant  $b_0$  is 206.9. After inserting these values into Equation (5) and applying to the test cases the overall prediction error (PE) is found to be 12.4%, which is less than the prediction errors produced by the ANN models (see Figure 2). Thus, it can be concluded that asymptotic regression is less acceptable than the ANN models.

## CONCLUSION

In this paper the autocorrelation structure of the time series pertaining to summer monsoon rainfall over India has been reviewed. It has been found that, in all the summer monsoon months, the total rainfall time series exhibit almost no serial correlation. Then the usefulness of implementing artificial

neural networks has been explained as a predictive tool for the average summer monsoon rainfall over India. Nineteen three-layered neural network models have been generated with variable hidden layer sizes. With the weight matrices available after training the networks, the test cases have been examined for validation of the models.

Prediction abilities of different neural network models have been tested using statistical procedures. Percentage errors of prediction yielded by those models have been used as a statistical measure of suitability of the predictive model to the time series under consideration. It has been observed that percentage error of prediction varies with variation in the hidden layer size. This error attains its minimum value in the case of the neural network model with eleven nodes in the hidden layer, that is, model  $M_{11}$ . Consequently the hidden layer with eleven nodes is identified as the best-hidden layer within a three-layered neural network producing minimum prediction error in the case of the average summer monsoon rainfall in India. But the supremacy of the  $M_{11}$  model cannot be established unless it is tested against a nonlinear regression model applied to the same time series as that of the neural network. The nonlinear regression in the form of asymptotic regression has been implemented in the training set as explained in the previous section. Then the validation process has been followed for the same test cases as in the case of neural networks. Finally it has been observed that asymptotic regression does not perform better than the neural network models. The conclusion, therefore, would be that the three-layered neural network model with eleven nodes in the hidden layer and trained with backpropagation learning can be a better alternative to the traditional regression approach in order to predict the average rainfall in the Indian summer monsoon. Though the study indicates the efficacy of neural networks in forecasting average summer monsoon rainfall over India, the average forecasting error is still in the neighborhood of 11%. This error may be narrowed by implementing hybrid soft computing techniques.

## ACKNOWLEDGEMENTS

The authors wish to express their sincere thanks to the anonymous reviewers for their valuable comments to enhance the quality of this paper.

## REFERENCES

- Cartalis, C. & Varotsos, C. 1994 Surface ozone in Athens Greece, at the beginning and at the end of the 20th-century. *Atmos. Environ.* **28** (1), 3–8.
- Chattopadhyay, S. 2007 Feed forward artificial neural network model to predict the average summer monsoon rainfall in India. *Acta Geophys.* **55** (3), 369–382.
- Chin, E. 1977 Modeling daily precipitation occurrence process with Markov chain. *Wat. Res. Res.* **13**, 949–956.
- Elsner, J. B. & Tsonis, A. A. 1992 Non-linear prediction, chaos, and noise. *Bull. Am. Meteorol. Soc.* **73**, 49–60.
- Gadgil, S., Rajeevan, M. & Nanjundiah, R. 2005 Monsoon prediction - why yet another failure? *Curr. Sci.* **88**, 1389–1499.
- Gardner, M. W. & Dorling, S. R. 1998 Artificial neural network (multilayer perceptron) - a review of applications in atmospheric sciences. *Atmosph. Environ.* **32**, 2627–2636.
- Gevrey, M., Dimopoulos, I. & Lek, S. 2003 Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecol. Model.* **160**, 249–264.
- Gregory, J. M., Wigley, T. M. L. & Jones, P. D. 1995 Application of Markov models to area average daily precipitation series and inter annual variability in seasonal totals. *Clim. Dyn.* **8**, 299–310.
- Guhathakurta, P. 2006 Long-range monsoon rainfall prediction of 2005 for the districts and sub-division Kerala with artificial neural network. *Curr. Sci.* **90**, 773–779.
- Hastenrath, S. 1988 Prediction of Indian summer monsoon rainfall: further exploration. *J. Clim.* **1**, 298–304.
- Haykin, S. 2001 *Neural Networks: A Comprehensive Foundation*, 2nd edn. Pearson Education, New Delhi.
- Hsieh, W. W. & Tang, T. 1998 Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bull. Am. Meteorol. Soc.* **79**, 1855–1869.
- Hu, M. J. C. 1964 Application of the ADALINE System to Weather Forecasting. *Masters Thesis, Technical Report 6775-1*, Standard Electronic Lab, Stanford, C.A.
- Jacovides, C. P., Varotsos, C., Kaltounides, N. A., Petrakis, M. & Lalas, D. P. 1994 Atmospheric turbidity parameters in the highly polluted site of Athens basin. *Renew. Energy* **4** (5), 465–470.
- Kalogirou, S. A., Neocleous, C., Constantinou, C. N., Michaelides, S. C. & Schizas, C. N. 1997 A time series construction of precipitation records using artificial neural networks. In: *Proceedings of EUFIT '97 Conference*, 8–11 September, Aachen, Germany. pp 2409–2413.
- Kamarthi, S. V. & Pittner, S. 1999 Accelerating neural network training using weight extrapolation. *Neural Networks* **12**, 1285–1299.
- Kartalopoulos, S. V. 1996 *Understanding Neural Networks and Fuzzy Logic - Basic Concepts and Applications*. Prentice Hall, New Delhi.
- Kishtawal, C. M., Basu, S., Patadia, F. & Thapliyal, P. K. 2003 Forecasting summer rainfall over India using Genetic

- Algorithm. *Geophys. Res. Lett.* **30**, doi:10.1029/2003GL018504.
- Kondratyev, K. Y. & Varotsos, C. A. 2001a Global tropospheric ozone dynamics – Part I: Tropospheric ozone precursors – Part II: Numerical modelling of tropospheric ozone variability. *Environ. Sci. Pollut. Res.* **8** (1), 57–62.
- Kondratyev, K. Y. & Varotsos, C. A. 2001b Global tropospheric ozone dynamics – Part II: Numerical modelling of tropospheric ozone variability – Part I: Tropospheric ozone precursors. *Environ. Sci. Pollut. Res.* **8** (2), 113–119.
- Lee, S., Cho, S. & Wong, P. M. 1998 Rainfall prediction using artificial neural network. *J. Geog. Inf. Decision Anal.* **2**, 233–242.
- Men, B., Xiejing, Z. & Liang, C. 2004 Chaotic analysis on monthly precipitation on hills region in Middle Sichuan of China. *Nature Sci.* **2**, 45–51.
- Michaelides, S. C., Neocleous, C. C. & Schizas, C. N. 1995 Artificial neural networks and multiple linear regression in estimating missing rainfall data. In: *Proceedings of the DSP95 International Conference on Digital Signal Processing, Limassol, Cyprus*. pp 668–673.
- Milionis, A. E. & Davis, T. D. 1994 Regression and stochastic models for air pollution. *Part I: Review comments and suggestions.* *Atmos. Environ.* **28**, 2801–2810.
- Nagendra, S. M. S. & Khare, M. 2006 Artificial neural network approach for modelling nitrogen dioxide dispersion from vehicular exhaust emissions. *Ecol. Model.* **190**, 99–115.
- Pal, S. K. & Mitra, S. 1999 *Neuro-Fuzzy Pattern Recognition*. Wiley Interscience, New York.
- Perez, P. & Reyes, J. 2001 Prediction of particulate air pollution using neural techniques. *Neural Comput. Appl.* **10**, 165–171.
- Perez, P., Trier, A. & Reyes, J. 2000 Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago. *Chile. Atmos. Environ.* **34**, 1189–1196.
- Pickering, N. B., Stedinger, J. R. & Haith, D. A. 1988 Weather input for nonpoint source pollution models. *J. Irrig. Drainage* **114**, 674–690.
- Rajeevan, M. 2001 Prediction of Indian Summer Monsoon: status, problems and prospects. *Curr. Sci.* **81**, 1451–1457.
- Sarle, W. 1997 *Neural Network Frequently Asked Questions*. Available at: <ftp://ftp.sas.com/pub/neural/FAQ.html>.
- Sejnowski, T. J. & Rosenberg, C. R. 1987 Parallel networks that learn to pronounce English text. *Complex Syst.* **1**, 145–168.
- Sivakumar, B. 2000 Chaos theory in hydrology: important issues and interpretations. *J. Hydrol.* **227**, 1–20.
- Sivakumar, B. 2001 Rainfall dynamics in different temporal scales: a chaotic perspective. *Hydrol. Earth Syst. Sci.* **5**, 645–651.
- Sivakumar, B., Liong, S. Y., Liaw, C. Y. & Phoon, K. K. 1999 Singapore rainfall behavior: chaotic? *J. Hydrol. Engng., ASCE* **4**, 38–48.
- Trentin, E. & Giulani, D. 2001 A mixture of recurrent neural networks for speaker normalization. *Neural Comput. Appl.* **10**, 120–135.
- Varotsos, C. 2005 Power-law correlations in column ozone over Antarctica. *Int. J. Remote Sensing* **26**, 3333–3342.
- Varotsos, C. & Cracknell, A. P. 2004 New features observed in the 11-year solar cycle. *Int. J. Remote Sensing* **25**, 2141–2157.
- Varotsos, C., Kondratyev, K. Y. & Efstathiou, M. 2001 On the seasonal variation of the surface ozone in Athens. *Greece. Atmos. Environ.* **35** (2), 315–320.
- Varotsos, C. & Kirk-Davidoff, D. 2006 Long-memory processes in ozone and temperature variations at the region 60 degrees S-60 degrees N. *Atmos. Chem. Phys.* **6**, 4093–4100.
- Varotsos, C., Ondov, J. & Efstathiou, M. 2005 Scaling properties of air pollution in Athens, Greece and Baltimore, Maryland. *Atmos. Environ.* **39**, 4041–4047.
- Widrow, B. & Lehr, M. A. 1990 30 years of adoptive neural networks; perceptron, madaline, and back propagation. *Proc. IEEE* **78**, 1415–1442.
- Wilks, D. S. 1991 Representing serial correlation of meteorological events and forecasts in dynamic decision-analytic models. *Mon. Weather Rev.* **119**, 1640–1662.
- Wilks, D. S. 1998 Multisite generalization of a daily stochastic precipitation generation model. *J. Hydrol.* **210**, 178–191.
- Wilks, D. S. 1999 Interannual variability and extreme-value characteristics of several stochastic daily precipitation models. *Agric. Forest Meteorol.* **93**, 153–169.
- Williams, J. R., Nicks, A. D. & Arnold, J. G. 1985 Simulator for water resources in rural basins. *J. Hydraul. Engng.* **111**, 970–986.
- Wong, K. W., Wong, P. M., Gedeon, T. D. & Fung, C. C. 1999 *Rainfall Prediction Using Neural Fuzzy Technique*. Available at: <http://www.it.murdoch.edu.au/~wong/publications/SIC97.pdf>
- Xue, Y. & Shukla, J. 1998 Model simulation of the influence of global SST anomalies on Sahel rainfall. *Mon. Weather Rev.* **126**, 2782–2792.
- Yegnanarayana, B. 2000 *Artificial Neural Networks*. Prentice-Hall, New Delhi.
- Zhang, M. & Scofield, A. R. 1994 Artificial neural network techniques for estimating rainfall and recognizing cloud merger from satellite data. *Int. J. Remote Sensing* **16**, 3241–3262.