Meteorol. Atmos. Phys. 62, 225–240 (1997)

Meteorology and Atmospheric Physics

[1] Indian Institute of Tropical Meteorology (IITM), Pashan, Pune, India
[2] Chemical Engineering Division, National Chemical Laboratory (NCL), Pashan, Pune, India

Prediction of All India Summer Monsoon Rainfall Using Error-back-propagation Neural Networks

C. Venkatesan[1], **S. D. Raskar**[2], **S. S. Tambe**[2], **B. D. Kulkarni**[2], and **R. N. Keshavamurty**[1]

With 5 Figures

Received June 24, 1996
Revised November 6, 1996

Summary

In this paper, multilayered feedforward neural networks trained with the error-back-propagation (EBP) algorithm have been employed for predicting the seasonal monsoon rainfall over India. Three network models that use, respectively, 2, 3 and 10 input parameters which are known to significantly influence the Indian summer monsoon rainfall (ISMR) have been constructed and optimized. The results obtained thereby are rigorously compared with those from the statistical models. The predictions of network models indicate that they can serve as a potent tool for ISMR prediction.

1. Introduction and Motivation

Being a nation with an agriculture-based economy, the southwest monsoon which arrives with a remarkable regularity is an important natural phenomenon for India. Due to its linkage with the global climatic conditions, the southwest monsoon has received considerable international attention in recent years. One of the major scientific issues in meteorology pertains to long-range forecasting of the Indian summer monsoon rainfall (ISMR) which is vital for India economically and socially. Blanford was the first to attempt a forecast of the monsoon based on the Himalayan snow cover extent and thickness (Blanford, 1884). Later, Sir Gilbert Walker (1908, 1910, 1933) attempted the development of an objective method for forecasting ISMR. Since 1979, as a part of the detailed global climatic observation and the MONEX program, several studies addressing the issue of ISMR prediction have been performed and they resulted in the increased understanding of the phenomenon. These studies are based mainly on the statistical models and have empirically related the ISMR with various predictors (parameters) (Thapliyal, 1981, 1982; Shukla and Paolino, 1983; Mooley et al., 1986; Shukla and Mooley, 1987). Since the physical processes involved are complex and the character and the strengths of the interactions amongst them are not clearly known, the ISMR prediction is a challenging task. Consequently, only partial success has been achieved in ISMR prediction.

Most systems encountered in the real-world are nonlinear and simple linear or linearized models cannot capture the essence of the underlying phenomena. Notwithstanding this, the usage of linear or linearized models continues for want of better methodologies to deal with the nonlinear systems. The task of ISMR prediction falls into this category and is traditionally carried out using a variety of linear regression models (Bhalme et al., 1986; Gowariker et al., 1989, 1991;

Mooley and Shukla, 1987; Parthasarathy et al., 1988; Prasad, 1992; Shukla, 1987; Thapliyal, 1990; Verma, 1982).

Prediction essentially involves determining the functional relationship(s) between the input-output variables of a system. Once such relationship is established from the available input-output data, it can be used to estimate the outputs corresponding to the new inputs for which output values are not available. The traditional method of empirical modelling is to assume some form of a fitting function with unknown parameters and subsequently employ the regression or optimization techniques for their estimation. The ARMA (Auto Regressive Moving Average) type models (Tong, 1990) fall into this category. Here, a polynomial expansion is performed to render the model *linear-in-the-parameters*. However, in this approach, it becomes necessary to apriori determine the structure of the empirical model or which terms to include so that the problem reduces to one of estimating the parameters using a least-squares criterion. In recent years, a new mathematical tool in the form of artificial neural networks (ANNs) has become available for performing nonlinear function approximation, pattern recognition, data compression, noise reduction etc. In the ensuing paragraphs ANNs are discussed in brief.

2. Artificial Neural Networks

Artificial neural networks were originally proposed to mathematically model the brain functions such as learning, pattern recognition, classification, and generalization. In the last decade, ANNs have been used in almost all the branches of science, engineering, and technology including atmospheric sciences for performing the above mentioned tasks. Although used for a wide variety of applications, the real power of ANNs, particularly that of the feedforward multilayered neural networks, lies in performing nonlinear function approximation and classification. The feedforward neural networks (FFNNs) have been used, for example, in satellite data retrieval and interpretation of satellite imagery (Peak and Tag, 1992, 1994; Zhang and Scofield, 1994), process identification and control (Chen and Billings, 1992), fault detection and diagnosis (Venkatasubramanian et al., 1990) etc. The FFNNs have the ability to approximate any nonlinear functional relationship between a set of input-output variables and it has triggered numerous applications including those in the nonlinear identification and prediction of weather systems (Elsner and Tsonis, 1992; Peak and Tag, 1989; Tang et al., 1994; Derr and Slutz, 1994; McCann, 1992; Thomson, 1996). Grieger and Latif (1994) investigated the dynamics of the El Nino phenomenon using neural networks. Minns and Hall (1996) used FFNNs in hydrology as rainfall-runoff models. It is to be noted that unlike the traditional linear or nonlinear regression techniques, the FFNNs do not require an explicit functional form for performing input-output mapping. In this sense they can be viewed as "black-box" models.

To impart into an FFNN, the above mentioned capabilities, the neural network is made to learn the relationship between the input-output data via a procedure called *network training*. The set of such input-output patterns is known as the training set. The most popular FFNN training algorithm is the error-back-propagation (EBP) proposed by Rumelhart et al. (1986a, b) and Werbos (1974). The detailed description of the EBP technique can be found in many places (e.g., Bishop, 1994; Haykin, 1994; Zupan and Gasteiger, 1993; Freeman and Skapura, 1991; Beale and Jackson, 1990; Hecht-Nielsen, 1990; Muller and Reinhardt, 1990; Wasserman, 1989; Yoh-Han Pao, 1989; Lippmann, 1987); however, for the sake of completeness, the methodology is very briefly described below.

The EBP algorithm is based on a gradient descent technique known as the *generalized delta rule* (GDR) which minimizes the average-squared-error between the actual and network predicted outputs by moving down the error surface (Schalkoff, 1992; Parker, 1982). An EBP network consists of at least three layers of neurons (also called as *processing elements* or *nodes*), namely, an input layer, an output layer and a middle layer (known as *hidden* layer). Each neuron in a layer is fully connected to all the neurons of the consecutive layer and the strength of the connection is known as *weight*. At the beginning of network training, the weights are randomly initialized. Figure 1 shows a typical three-layer feedforward neural network. The input layer neurons do not perform any computa-

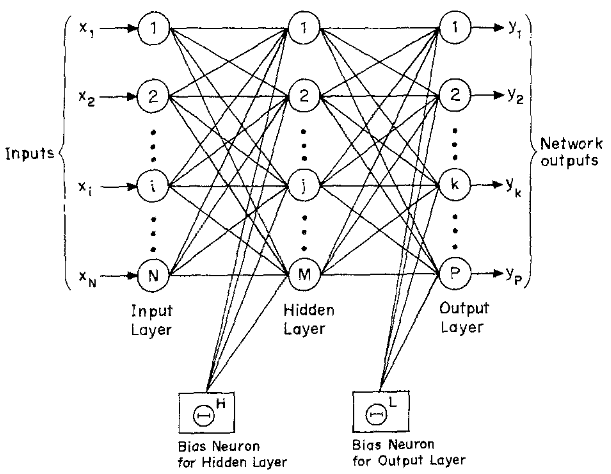

Fig. 1. A three layer feedforward neural network

tions and merely distribute the input values over all the neurons in the hidden layer. Each node in the hidden layer first computes the weighted-sum of its inputs which is passed through a nonlinear transfer function to arrive at the output. The common choice of the nonlinear transfer function is either the logistic sigmoid or hyperbolic tangent (*tanh*). The nonlinear function approximation ability of the EBP networks is due to the nonlinear transfer function used for computing the outputs of the hidden units. It is to be noted that the use of a nonlinear transfer function for the output layer nodes is optional. The output values of the hidden layer neurons serve as the inputs to the next (output) layer neurons. Since each neuron in a layer passes its output to every neuron in the next layer, the network is called a *feedforward network*. The output of the output layer nodes forms the network output. Subsequently, the network outputs are compared with the target (desired) values, and the error, termed as the *prediction error*, is evaluated. This error is used to correct, according to the GDR strategy, the connection weights between the hidden and output layers and, subsequently, those between the hidden and input layers. The changes in the weights are made by presenting the input-output patterns of the training data repeatedly until a prespecified error criterion is fulfilled. At this point the weights are said to be converged.

It is not enough that the EBP network predicts well only the outputs corresponding to the training set inputs. It is necessary that the trained network also predicts the outputs corresponding to the new (or novel) inputs which are not part of the training set. The network ability of accurately predicting the outputs of the novel input vectors is known as the *generalization* ability and is vital for the satisfactory function approximation performance by the network. To ensure that the network possesses good generalization capability, the available data is divided into two parts known as the *training* set and *test* set. The training set is used for network learning and the test set to simultaneously evaluate the generalization ability of the network.

The EBP algorithm sometimes suffers from problems such as slow convergence and entrapment into a local minimum of the error surface. According to one strategy (Rumelhart et al., 1986a) inclusion of the *momentum* term in the weight-adjustment (learning) equation speeds up the network training and also helps in avoiding a local minimum. While training the EBP network a precaution must be exercised to avoid what is known as *overfitting* phenomenon. Overfitting takes place if: (i) the network is trained over very large number of training iterations, and (ii) the hidden layer contains *more-than-necessary* number of neurons. The ill-effect of overfitting is that the network learns minute details (e.g., noise) in the training data at the cost of learning the smooth trend therein. Consequently, the network fails to generalize and makes erroneous predictions for the new inputs. To ensure that the network generalizes best and is not overfitted, following methodology is usually adopted: (i) stop the network training as soon as the root-mean-squared (RMS) error with respect to the test set reaches a minimum, (ii) explore the error surface rigorously so as to reach the deepest possible error minimum or the global minimum, (iii) study the effect of the number of hidden units on the network's generalization ability. The by-product of this strategy aimed at avoiding overfitting is, in the process, the problem of determining the optimal network structure is solved; which is essentially one of fixing the minimum number of hidden nodes required for addressing a specific mapping or classification task (Tambe et al., 1996). Several strategies to improve performance of the EBP learning algorithm have been proposed which are reviewed by Xu et al. (1992) and Huang and Huang (1991).

Elsner and Tsonis (1992) have reviewed the applications of artificial neural networks to the prediction problems arising in meteorology and have given examples of time series forecasting (also see Tsonis and Elsner, 1988, 1989). Navone and Ceccatto (1994) have shown that forecasting of the Indian monsoon can be performed via ANNs. Goswami and Srividya (1996) have conducted ANN-assisted ISMR prediction employing the ISMR time series of 120 years. This approach is similar to the classical stochastic time series analysis employed by Basu and Andharia (1992) wherein previous values of ISMR are used to predict its future value. In the present paper, the FFNNs trained using the EBP algorithm have been used for approximating the nonlinear relationship between the ISMR predictors and the ISMR. Specifically, three EBP networks that use two, three and ten input parameters (predictors) respectively, have been constructed, trained and optimized. In the following paragraphs, the construction of the data sets used for training and testing the networks is described followed by a discussion on the significance of the input parameters. The three network models hereafter are referred to as model-I (two parameters), model-II (three parameters), and model-III (ten parameters), respectively.

3. Data Organization

The selection of suitable input parameters for the prediction of ISMR is of critical importance and the early attempts were made by Blanford and Walker. Walker (1908, 1910) selected four parameters to develop a multiregression model for forecasting ISMR. Several attempts have been made since then to identify the suitable global and regional predictors for ISMR prediction (Hastenrath, 1986, 1987, 1995; Bhalme et al., 1986; Shukla, 1987; Parthasarathy et al., 1988, 1993; and Shukla and Mooley, 1987). Most of these are monthly/seasonal averages of the surface and upper-air parameters corresponding to the winter and premonsoon periods. These are identified on the basis of their significant linear correlations with the ISMR. Based on their spatial domain, the predictors can be classified into four major groups as: (i) Indian region; (ii) Indian ocean region; (iii) El Nino; and (iv) global/hemispheric conditions.

To serve as a significant parameter for the model development, its correlation with the predictand (in our case ISMR) needs to be stable. Parthasarathy (1984) and Hastenrath (1987) have shown that to obtain a stable correlation coefficient between the Indian rainfall and regional/global circulation parameters, the data span needs to be of at least 20 to 30 years. The data used to construct the training and test sets for neural network modelling were obtained from various sources and are highly reliable. In this study, a total of 12 predictors have been used.

It is to be noted that we are constrained by the unavailability of all the twelve predictor data for equal length of time. The data of two parameters are available for 56 years (1939–1994), three parameters for 38 years (1957–1994), and ten parameters for 25 years (1963–1987). Due to this variability, the training and test sets had to be separately defined for each network model. It is essential from the generalization viewpoint that the training and test sets are properly chosen. Specifically, both the data sets should be sampled in such a way so as to be the true representatives of the entire input-output state space. In the following, the importance of significant predictors is discussed.

3.1 All India Summer Monsoon Rainfall

All India (India taken as a single unit) summer monsoon (June-September) rainfall data for the period 1939–1994, prepared by area-weighting 306 well-distributed (over the entire country) rain-guages has been taken from Parthasarathy et al. (1992, 1994). A detailed discussion of preparing these data and their magnitudes can be found in Mooley and Parthasarathy (1984) and Parthasarathy et al. (1987, 1992).

3.2 Southern Oscillation Indicator

The seesaw oscillation in the sea level pressure between the Indian ocean and the east Pacific ocean over the near-equatorial latitudes is termed as southern oscillation. Shukla and Paolino (1983), Parthasarathy and Pant (1985), and Shukla and Mooley (1987) showed that the seasonal tendency parameter MAM (March to May) minus DJF (December to February) of the

(Tahiti-Darwin) southern oscillation index (SOI) is a useful parameter for predicting the monsoon rainfall. During the deficient summer monsoon rainfall years over India, the SOI has been found to be negative (higher sea level pressure at Darwin and lower at Tahiti). The April minus January sea level pressure at Darwin (DSLP) is also considered as an indicator of the southern oscillation.

3.3 Surface Air Temperature

It has been identified that the pre-monsoon (March to May) thermal structure over India is a good indicator for the ensuing monsoon rainfall. Parthasarathy et al. (1990) showed that the mean surface temperatures at the west central India (WCl) stations during March, April and May have a high significant correlation (0.6) with the monsoon rainfall for the period 1951–1980. Mooley and Paolino (1988) have shown that the minimum temperature in May over the western Indian region may be considered as one of the predictors in the seasonal monsoon rainfall forecast. Krishnakumar et al. (1995) examined the spatial patterns of correlation coefficients of ISMR with the maximum and minimum temperatures at 121 stations in India and found that the minimum temperatures are more useful as predictors than the maximum ones. The correlation coefficient of WCl May minimum temperature (averaged over 10 stations) with ISMR for the period 1963–1987 is 0.649.

3.4 Location of 500 hPa Ridge over India

The north-south seasonal migration of the subtropical ridge provides a good indicator for the seasonal monsoon prediction. The ridge is located in the southernmost part during winter, and as the season advances, it gradually shifts northwards. Banerjee et al. (1978) first suggested a relationship between the monsoon activity and the ridge location. Mooley et al. (1986) have examined in detail the relationship and its stability between the 500 hPa ridge location along 75 degree east during April and the ISMR as well as the subdivisional monsoon rainfall for the period 1939–1984. Krishnakumar et al. (1992) found that the difference between the ridge locations during March and April shows a correlation coefficient of 0.73 with ISMR.

3.5 Bombay pressure Tendency

Bombay pressure, which is associated with the tropical circulation as well as the southern oscillation (Wright, 1975), is one of the leading predictors of the ISMR. Parthasarathy et al. (1991) showed that the seasonal mean sea level pressure tendency (MAM-DJF) for Bombay shows a significant correlation coefficient of −0.7 with the monsoon rainfall from the year 1951 onwards. This relationship is dominant only when the ENSO variance in Bombay pressure is high.

The set of parameters used in this study are: (1) Darwin sea level pressure (MAM-DJF) tendency (DWPM-D); (2) Tahiti-Darwin sea level pressure tendency (T-DPM-D); (3) Agalega sea level pressure tendency (AGPM-D); (4) Bombay sea level pressure tendency (BBPM-D); (5) Adelaide sea level pressure tendency (ADPM-D); (6) West Central India (WCI 10 station average) May minimum temperature (WCTN MAY); (7) Jodhpur mean temperature (May) (JDP.TMEA); (8) Jodhpur pressure (May) (JDP.PMAY); (9) Meridional wind index 200 hPa (May) (MWI200M); (10) April 500 hPa ridge location (Ridge 500); (11) De Bilt surface temperature (Jan.) (DeBT); and (12) Darwin sea level pressure (April-Jan.) (DSLP).

4. Results and Discussion

Based on the past studies and the analysis of the data from 1939-1984, Shukla and Mooley (1987) concluded that the two predictors, namely, the difference between the April and January sea level pressure at Darwin (DSLP) and the location of the 500 mb ridge during April along 75 E, show the most significant relationship with ISMR. Thus, these two predictors for the period 1939-1994 were chosen as the two inputs in network model-I. A total of 56 patterns (vectors) were divided into the training and test sets consisting of 49 and 7 patterns, respectively. Table 1 lists the values of the two predictors along with the respective ISMR. The patterns in the test set are marked with an asterisk. All the three models were trained using the error-back-propagation (EBP) algorithm. The stepwise training procedure that uses the momentum term in the learning rule, and the logistic sigmoid

Table 1. *Data for Model-I*

Year	Ridge	DSLP	All India Rainfall (mm)	Year	Ridge	DSLP	All India Rainfall (mm)
1939	14.0	4.4	789.7	1967	17.5	5.6	860.1
1940	15.3	5.0	853.5	*1968	12.5	5.7	754.6
1941	11.2	2.8	728.4	1969	17.3	5.2	831.0
1942	17.5	1.8	957.9	1970	15.8	1.8	939.8
*1943	16.0	2.5	868.4	1971	16.8	2.2	886.8
1944	14.5	3.0	920.6	1972	11.0	4.8	652.9
1945	16.7	3.8	911.1	1973	16.8	1.7	913.4
1946	17.3	4.1	903.8	1974	13.5	4.8	748.1
1947	18.0	2.1	945.6	1975	17.5	1.3	962.9
1948	14.5	1.7	874.0	* 1976	17.0	4.8	856.8
1949	17.0	2.8	904.1	1977	14.0	3.4	883.2
1950	17.0	3.0	877.0	1978	14.0	1.9	909.3
1951	12.0	4.1	739.1	1979	12.5	3.7	707.8
*1952	13.5	3.6	793.1	1980	15.0	3.8	882.8
1953	17.0	2.0	923.2	1981	17.0	4.8	852.2
1954	16.5	2.8	885.6	1982	11.3	3.6	735.4
1955	15.5	0.6	930.4	1983	14.5	0.3	955.7
1956	17.5	3.5	983.3	* 1984	14.8	3.0	836.7
1957	16.0	3.0	788.7	1985	14.7	−0.2	759.8
1958	17.0	0.4	889.4	1986	15.0	4.6	743.0
1959	16.0	2.8	944.1	1987	14.0	4.4	697.3
*1960	16.7	2.0	839.8	1988	14.5	2.3	961.5
1961	15.0	1.9	1020.3	1989	16.5	3.1	866.7
1962	14.8	4.7	809.8	* 1990	15.0	0.1	908.7
1963	13.5	2.4	857.9	1991	14.5	3.6	785.2
1964	18.3	1.9	922.6	1992	13.5	0.1	785.5
1965	14.0	5.0	709.4	1993	14.5	5.6	894.0
1966	13.5	2.5	739.9	1994	14.0	4.1	938.3
				1995	13.5	2.8	

* The data of the test set.

transfer function at the hidden and output layer nodes is described in Appendix-I.

The network inputs as well as the outputs listed in Table 1 were normalized so as to lie between 0 and 1. The normalization was carried out by simple linear transformation. The network architecture in model-I consists of two neurons in the input layer and one neuron (representing the ISMR) in the output layer. The network training consisted of 15,000 iterations (sweeps) through the training set. A single iteration involves as many forward and reverse passes as the number of input-output patterns in the training set. In the forward pass, network outputs are established, and based on the prediction error, in the reverse pass the network weights are adjusted. After each sweep, the network outputs for all the test set patterns also were computed and the RMS errors with respect to the training (E_{trn}) and the test (E_{tst}) sets were evaluated. At the end of 15,000 iterations, the E_{tst} profile as a function of training iteration was examined to locate its minima. This procedure is repeated several times by assuming a different set of random numbers for initializing the network weights. Note that in these runs (which are termed as *error-surface-exploration* runs) the number of hidden layer neurons is kept constant. The weights corresponding to that error-surface-exploration run which yields the lowest E_{tst} magnitude are considered to be optimal for the chosen number of hidden nodes. The objective of the error-surface-exploration runs is to reach the global minimum or the deepest possible local minimum on the error surface. To avoid overfitting and arrive at the optimal network architecture such error-surface-exploration runs were conducted by varying the number of hidden layer units between one and ten. We conducted 25 error-surface-exploration runs for a fixed number of hidden units. Figure 2a shows the minimum of the E_{trn} and E_{tst} obtained during the error-surface-exploration runs as a function of the number of hidden layer neurons. It can be seen that for six hidden units, the magnitude of the test set error (E_{tst}) is the smallest. This indicates that the optimal network architecture should have six neurons in the hidden layer. The optimal weights are the weights corresponding to that error-surface-exploration run which for six hidden units, yielded lowest E_{tst} value. The optimal weights so obtained are listed in Tables 2a and 2b. The detailed steps for computing the network-predicted ISMR value using the weight values in Tables 2a and 2b and for Ridge and DSLP values of 14.0 and 4.4, respectively, are given in Appendix-II.

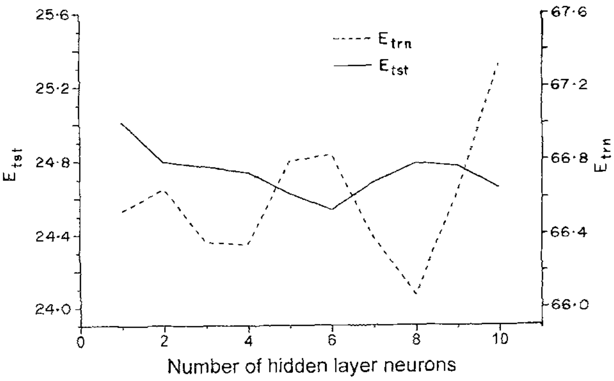

Fig. 2a. RMS errors with respect to Training and Test sets as a function of number of hidden layer neurons

The EBP network with the optimal architecture [2 (no. of input nodes): 6 (no. of hidden nodes): 1(no. of output nodes)] and weights yielded an RMS error value of 66.83 mm for the training set and 24.53 mm for the test set. The corresponding correlation coefficients (CC) were 0.7 and 0.91, respectively. The error in the prediction regime is smaller than half of the standard deviation of the ISMR series (84.05). The rainfall value predicted by the model-I for the year 1995 (DSLP = 2.8 and Ridge = 13.5) was 835.96 mm which is very close to the actual rainfall value (827.1 mm). Note that the predictor data for the year 1995 was not included in either the training set or the test set. Figures 2b and 2c, respectively, show the 3-dimensional plot of the actual and model-I fitted data. It can be observed that the model has captured the intrinsic relationship between the two predictors and ISMR. The points that show high deviations on either side of the average trend are treated by the network as

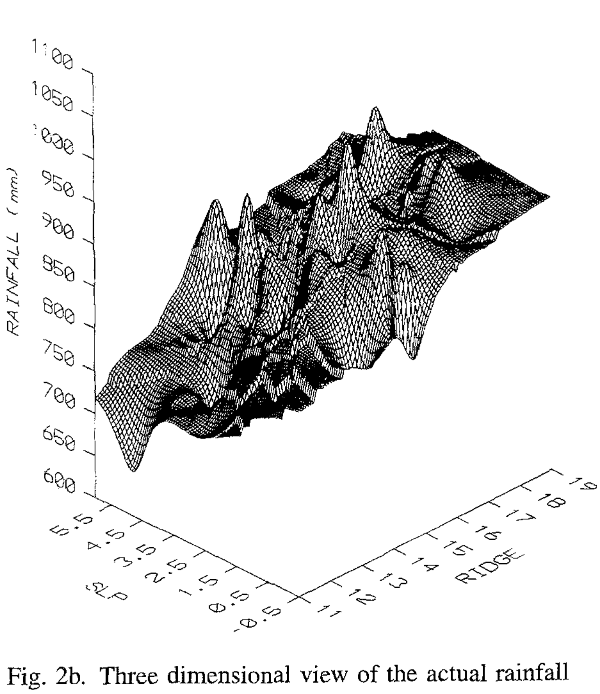

Fig. 2b. Three dimensional view of the actual rainfall

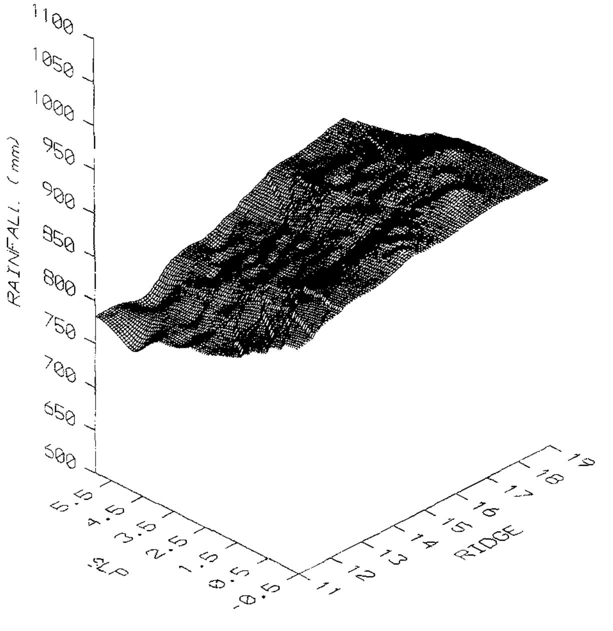

Fig. 2c. Three dimensional view of the model-I predicted rainfall

Table 2a. *Optimal Weights for Model-I*

From	To *j*th hidden layer neuron					
	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
Input neuron 1	−0.294163	−0.435573	0.168131	−1.43422	−1.69509	−0.109301
Input neuron 2	0.319022	−0.548771	−0.569852	1.16194	1.02023	−0.0324836
Bias neuron, Θ^H	−0.406677	−0.435841	−0.740890	−0.434811	−0.401620	0.379037

Table 2b. *Optimal Weights for Connections Between the Hidden Neurons and the Output Neuron*

From	To the output neuron
Hidden neuron 1	−0.484905
Hidden neuron 2	0.389519
Hidden neuron 3	0.978241
Hidden neuron 4	−1.21877
Hidden neuron 5	−1.89753
Hidden neuron 6	0.342731
Bias neuron, Θ^L	0.742981

noise and thus are filtered. Figure 3 shows the comparison between the model-I predicted and the observed rainfall values corresponding to the training and test sets.

Dugam et al. (1993) identified the De Bilt surface temperature (Jan.) anomaly as a potential predictor for ISMR. Hence, this predictor has been taken as an input to model-II, in addition to the two inputs of model-I. The optimal network configuration for model-II was found to be 3:3:1. The inputs chosen for this model correspond to the period 1957–1994. The RMS error values for the training and test sets have been 65.62 mm and 33.7 mm, respectively, and the corresponding correlation coefficient values are 0.68 and 0.96, respectively. The optimized network weights for model-II are listed in Tables 4a and 4b. The model-II predicted rainfall value for the year 1995 (DSLP = 2.8, Ridge = 13.5 and DeBilt = 1.6) was 842.04 mm. As can be noted even this value is very close to the actual rainfall during the year 1995. The comparison between the observed and the model-II predicted rainfall values for the training and test sets is shown in Fig. 4.

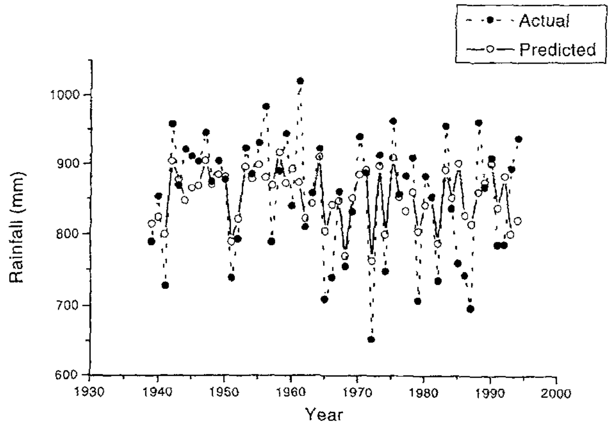

Fig. 3. Comparison between predicted and observed rainfall for 2-parameter model

In model-III, ten well-established and stable parameters for the period 1963–1987 have been used (Table 5). The correlation coefficient of these parameters with the ISMR are listed in Table 6. After training, the optimal network configuration comprising 6 hidden neurons was established. The optimum weights for this model are listed in Tables 7a and 7b. The RMSE values for the training and test sets were 26.97 mm and 46.55 mm, respectively, and the corresponding correlation coefficients were 0.96 and 0.68, respectively. Figure 5 shows the comparison between the model-III predicted and observed rainfall values. The ISMR prediction performance of models I to III is summarized in Table 8.

We have compared our results with those in the well-cited paper by Shukla and Mooley (1987) (hereafter referred as SM), in which the linear regression technique has been used for ISMR prediction. Additionally, we have compared our results with those of Navone and Ceccatto (1994) (hereafter referred as NC) who have used two types of neural network models for ISMR prediction. The first model by NC has 2:2:1 structure. The second one is a hybrid model in which two (deterministic and stochastic) neural networks were linked by connecting the output units to a new neuron. NC have recomputed the values of the regression coefficients using the data by SM and used them as the

Table 3. *Data for Model-II*

Year	Ridge 500 (mb)	DSLP	De Bilt (Temp)	All India Rain Fall (mm)	Year	Ridge 500 (mb)	DSLP	De Bilt (Temp)	All India Rain Fall (mm)
1957	16.0	3.0	+1.4	788.7	1977	14.0	3.4	+1.3	883.2
1958	17.0	0.4	−0.2	889.4	1978	14.0	1.9	+1.3	909.3
1959	16.0	2.8	−0.6	944.1	1979	12.5	3.7	−4.9	707.8
1960	16.7	2.0	+0.5	839.8	1980	15.0	3.8	−1.5	882.8
1961	15.0	1.9	−0.3	1020.3	1981	17.0	4.8	+1.0	852.2
*1962	14.8	4.7	+1.8	809.8	*1982	11.3	3.6	−0.6	735.4
1963	13.5	2.4	−6.9	857.9	1983	14.5	0.3	+4.5	955.7
1964	18.3	1.9	−1.1	922.6	1984	14.8	3.0	+1.7	836.7
1965	14.0	5.0	+1.0	709.4	1985	14.7	−0.2	−4.8	759.8
*1966	13.5	2.5	−1.3	739.9	*1986	15.0	4.6	−3.6	743.0
1967	17.5	5.6	+1.4	860.1	1987	14.0	4.4	−4.7	697.3
1968	12.5	5.7	+0.3	754.6	1988	14.5	2.3	+3.9	961.5
1969	17.3	5.2	+1.2	831.0	1989	16.5	3.1	+2.5	866.7
1970	15.8	1.8	−1.1	939.8	*1990	15.0	0.1	+3.7	908.7
*1971	16.8	2.2	+0.6	886.8	1991	14.5	3.6	+1.2	785.2
1972	11.0	4.8	−1.2	652.9	1992	13.5	0.1	+0.7	785.5
1973	16.8	1.7	+1.2	913.4	1993	14.5	5.6	+3.1	894.0
1974	13.5	4.8	+3.5	748.1	1994	14.0	4.1	+3.2	938.3
1975	17.5	1.3	+4.5	962.9	1995	13.5	2.8	+1.6	
*1976	17.0	4.8	+2.5	856.8					

* The data of the test set.

Table 4a. *Optimal Weights for Model-II*

From	To *j*th hidden layer neuron		
	$j = 1$	$j = 2$	$j = 3$
Input neuron 1	0.298638	−0.0135114	2.01872
Input neuron 2	0.521381	−0.560152	2.28456
Input neuron 3	0.202018	1.18764	−1.05404
Bias neuron, Θ^H	0.885472	−0.272585	−1.12101

Table 4b. *Optimal Weights for Connections Between the Hidden Neurons and the Output Neuron*

From	To the output neuron
Hidden neuron 1	0.0286994
Hidden neuron 2	−1.46296
Hidden neuron 3	2.93218
Bias neuron Θ^L	−1.21651

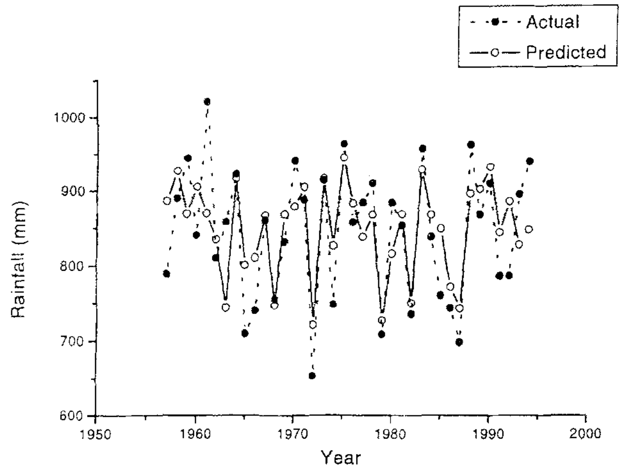

Fig. 4. Comparison between predicted and observed rainfall for 3-parameter model

standard for comparing the results from the network models. The results obtained by SM and NC are summarized in Table 9. Note that our work differs from that of the SM and NC in the size of the data used for constructing the models. Specifically, we have used most recent (1939–1994) two parameter data for constructing model-I (SM and NC have used the data between 1939–1984). A notable feature of the post-1984 predictor data is that they show greater variability. As can be noticed from Table 9, the CC values for the test sets of model-I and model-II compare favorably with both the SM and NC

Table 5. *Data for Model-III*

Year	WCTN MAY	APR500	ADPM-D	JDP-PMAY	JDP .TMEA	DWPM-D	T-DPM-D	AGPM-D	BBPM-D	MW1200M	AISMR (mm)
1963	26.1	13.50	−0.41	1003.0	34.1	−1.59	0.12	−0.89	+0.55	3.6	857.9
1964	26.1	18.30	−0.37	1001.0	34.2	−1.73	2.43	−0.78	−0.64	−2.0	922.6
1965	25.3	14.00	−0.24	1003.0	33.1	0.84	0.17	0.02	1.02	4.6	709.4
*1966	25.1	13.50	0.46	1003.0	33.8	1.22	−0.91	−0.11	1.27	6.1	739.9
1967	25.4	17.50	1.44	1002.0	33.3	1.19	−0.97	−0.40	1.02	2.0	860.1
1968	24.7	12.50	−1.15	1003.0	33.3	0.99	0.00	0.39	0.49	5.8	754.6
1969	26.4	17.30	1.16	1002.0	34.5	1.46	0.87	0.37	0.84	3.2	831.0
1970	26.8	15.80	−0.12	1000.0	36.1	−0.77	1.13	−0.90	−0.97	0.3	939.8
*1971	25.8	16.80	0.9	1001.2	33.3	−0.16	1.32	1.03	1.03	0.5	886.8
1972	25.6	11.00	3.01	1002.7	33.5	2.77	−2.08	1.76	1.75	8.1	652.9
1973	26.6	16.80	−1.47	1000.5	34.5	−3.17	2.48	−0.92	−1.46	0.9	913.4
1974	25.3	13.50	0.71	1002.7	33.0	0.53	−0.35	0.61	0.09	2.8	748.1
1975	26.6	17.50	−0.21	1000.3	34.2	−1.31	1.97	−1.41	−1.66	0.1	962.9
1976	25.8	17.00	0.79	1001.7	33.9	1.24	−1.28	0.62	0.78	0.6	856.8
*1977	25.8	14.00	−0.06	1001.7	33.6	1.05	−1.87	0.11	0.91	0.1	883.2
1978	26.4	14.00	−1.74	999.7	34.9	−1.28	2.59	−0.80	−2.31	−0.3	909.3
1979	24.6	12.50	0.42	1003.7	32.4	0.4	−0.37	−0.21	0.51	1.1	707.8
1980	26.5	15.00	−0.12	1001.7	34.7	−0.04	−1.66	−0.13	0.27	3.0	882.8
*1981	25.8	17.00	1.05	1001.3	33.7	1.75	−0.76	−0.92	0.14	0.6	852.2
1982	25.0	11.30	0.31	1004.7	31.0	0.50	−1.45	0.00	0.69	3.7	735.4
1983	26.0	14.50	−1.49	1002.7	34.4	−1.27	2.53	0.40	0.19	1.8	955.7
1984	26.4	14.80	0.37	1000.7	35.2	0.44	−0.81	1.70	1.30	−0.7	836.7
*1985	26.2	14.70	1.11	1000.3	34.9	−1.64	0.53	1.08	2.14	−0.5	759.8
1986	26.2	15.00	1.26	1001.3	34.0	1.04	0.11	0.58	0.14	−0.5	743.0
1987	25.9	14.00	1.34	1005.2	33.0	2.01	−1.28	1.89	0.42	5.1	697.3

* The data of the test set.

Table 6. *Correlation Coefficients Between All India Summer Monsoon Rainfall and Different Parameters Used in this Study for the Period 1963–1987*

Parameters	Correlation Coefficients
Darwin sea level pressure (MAM-DJF)	−0.629
Tahiti-Darwin sea level pressure (MAM-DJF)	0.598
Agalega sea level pressure (MAM-DJF)	−0.575
Bombay sea level pressure (DJF)	−0.589
Adelaide sea level pressure (MAM-DJF)	−0.561
West Central India (10 station average) May minimum Temperature	0.649
Jodhpur mean Temperature (May)	0.576
Jodhpur Pressure (May)	−0.655
Meridional wind Index 200 hPa (May)	−0.629
April 500 hPa ridge location	0.679

models. Also, the RMSE value with respect to the test set (which represents network's generalization capability for new inputs) of model-I is much less than the respective RMSE values for the test sets used by SM and NC.

Amongst the available number of predictors, some are known to have lost their prediction potential owing to the weaker correlation or even reversal of the correlation sign during certain periods (Parthasarathy et al., 1991, 1993). Fu and Fletcher (1988) and Elliot and Angell (1987, 1988) have showed that the variations in the correlation of these parameters are related to the changes in tropical circulation features. In recent years, especially after 1980, the correlation coefficients of almost all the predictors with the ISMR have decreased. Hence, the variance explained by some of the predictors also have

Table 7a. *Optimal Weights for Model-III*

From	To j^{th} hidden layer neuron					
	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
Input neuron 1	−0.118233	0.758983	−0.259375	−0.758112	−1.38346	−0.540929
Input neuron 2	−0.383483	3.35506	0.479517	−0.885589	−0.576843	−0.719829
Input neuron 3	1.02044	−2.79202	0.272786	−0.0900608	0.230365	0.0734147
Input neuron 4	0.817870	−0.436693	0.763278	−0.211119	−0.777732	0.440674
Input neuron 5	−0.955722	1.27883	0.301569	−0.579689	0.418510	−0.539417
Input neuron 6	−0.135996	−0.998858	−0.214136	−0.207407	1.02881	0.300735
Input neuron 7	0.0018947	−0.349058	−0.176352	−0.549264	0.334104	−0.757369
Input neuron 8	0.183087	−0.616139	0.0969557	−0.808030	0.883836	−0.992596
Input neuron 9	−0.290802	−0.488512	0.615908	−0.760707	−1.31930	0.242115
Input neuron 10	−0.497865	−0.742598	−0.050585	−0.625341	0.414690	−0.924803
Bias neuron, Θ^H	0.772729	0.224651	−0.121918	−0.143438	0.582924	−0.608201

Table 7b. *Optimal Weights for Connections Between the Hidden Neurons and the Output Neuron*

From	To the output neuron
Hidden neuron 1	−1.18713
Hidden neuron 2	3.29397
Hidden neuron 3	−0.0304826
Hidden neuron 4	0.382623
Hidden neuron 5	−1.41766
Hidden neuron 6	0.0354864
Bias neuron, Θ^L	−0.119173

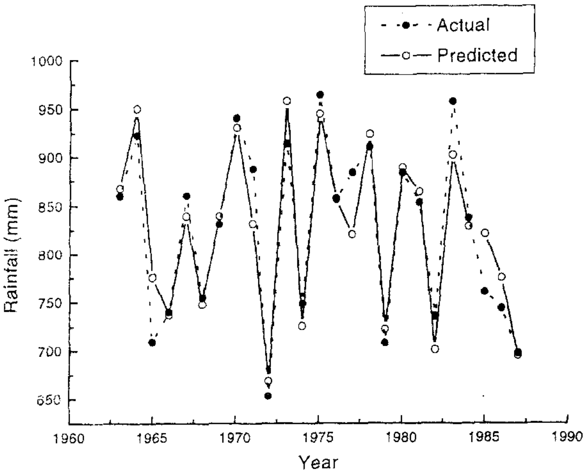

Fig. 5. Comparison between predicted and observed rainfall for 10-parameter model

Table 8. *The RMSE and CC Values Corresponding to the Training and Test sets for the Three Neural Network Models*

	model-I		model-II		model-III	
	TRN	TST	TRN	TST	TRN	TST
RMSE	66.83	24.53	65.62	33.7	26.97	46.55
CC	0.7	0.91	0.68	0.96	0.96	0.68

Table 9. *The RMSE and CC Values for the SM and NC Models*

	SM		NC			
			2:2:1 Neural Network		Hybrid Neural Network	
	TRN	TST	TRN	TST	TRN	TST
RMSE	50.9	41.5	50.6	33.6	49.0	26.7
CC	0.76	0.89	0.77	0.94	0.78	0.95

Table 10. *The RMSE and CC Values Corresponding to the linear Regression Models Using the Training and Test sets of the Three ANN Models*

	Regression model-I		Regression model-II		Regression model-III	
	TRN	TST	TRN	TST	TRN	TST
RMSE	61.39	32.85	65.83	35.28	35.29	61.33
CC	0.715	0.885	0.683	0.967	0.928	0.3

decreased considerably. For example, the standard deviation of DSLP (1.508) and ISMR (84.05) for the period 1939–1994 are higher than the corresponding standard deviation values for the period 1939–1984 (DSLP = 1.372 and rainfall = 82.91). This may be one of the reasons why we obtained slightly higher RMSE values for the training sets of neural network model-I and model-II as compared to those of SM and NC.

As a thumb rule, the number of patterns in the training set should be greater than or equal to the total number of weights in the EBP network. In model-III the number of data points used are 25, which is much smaller as compared to the total number of network connections (73). Thus, there is a possibility that the network model-III is an overfitted one. This factor might have contributed to the higher RMS error observed for the test set of model-III. Despite the grossly inadequate training data, the model-III ISMR predictions are not discouraging. As can be observed, the RMSE value for the model-III training set is significantly lower (26.97) than the RMSE values (50.9, 50.6, and 49.0) corresponding to the three training sets used by SM and NC. The ISMR prediction for 1995 using neural network model-III could not be attempted since the ten parameter data for the year 1995 is not available so far. We have also compared the neural network results with those obtained by performing linear regression analysis for the training sets of all the three ANN models. After estimating the regression coefficients, the same were used to predict the ISMR values corresponding to the respective test sets. The RMSE and CC values obtained for the three linear regression models are listed in Table 10. As can be easily verified from the values in Tables 8 and 10, all the FFNN models perform better as compared to the linear regression models.

5. Conclusion

This study has demonstrated that the multi-layered feedforward neural networks, trained using the EBP algorithm, can be effectively used to predict the Indian summer monsoon rainfall. Networks with single hidden layer have been studied for this purpose. Three network models with two, three and ten input parameters, respectively, have been analyzed. ISMR predictions based on the neural network models I and II for the year 1995 have been found to be near-accurate. Using neural network models I and II, we have made the ISMR predictions for the current year (i.e. 1996) for which Ridge = 13.5, DSLP = 3.2 and DeBilt = −2.09, and the values obtained thereby are 828.7 and 784.9 mm, respectively.

Acknowledgement

We are grateful to Dr. B. Parthasarathy for providing us the all India seasonal rainfall data. Our sincere thanks are due to Dr. K. Rupakumar for providing the data set for ten parameter model and Mr. S. S. Dugam for the De Bilt temperature data. The authors S.D.R., S.S.T., and B.D.K. from NCL gratefully acknowledge the Department of Science and Technology (DST), Govt. of India, for the support to develop some of the neural network tools used in this study. We thank the anonymous reviewers for their valuable suggestions towards the improvement of the paper.

Appendix-I

Error-back-propagation Algorithm

The detailed numerical steps to train a three-layer feedforward neural network by error-back-propagation algorithm are as follows.

Step 1: Initialize the connection weights to small random values so as to lie between −1 and 1.

Step 2: Apply an input pattern (vector) $X_q = (x_{q1}, x_{q2}, ..., x_{qN})^T$ to the input layer neurons where q refers to the pattern index.

Step 3: Compute the weighted sum of inputs for individual neurons in the hidden layer according to

$$Net^H_{qj} = \sum_{i=1}^{N} W^H_{ji} x_{qi} + \Theta^H_j ; j = 1, M \qquad \text{(A1)}$$

where Net^H_{qj} denotes the weighted sum for jth hidden layer neuron when qth input pattern has been applied; W^H_{ji} denotes the weight between ith input layer neuron and jth hidden layer neuron; x_{qi} represents the ith element of input pattern, X_q; N refers to the number of input layer neurons, M refers to the number of hidden units and Θ^H_j represents the bias weight for jth hidden layer neuron.

Step 4: Transform the weighted sum using an activation function (e.g. Sigmoid activation function) to get the outputs of the hidden layer neurons according to:

$$I^H_{qj} = \frac{1}{1 + \exp(-Net^H_{qj})} ; j = 1, M \qquad \text{(A2)}$$

where I^H_{qj} refers to the output of jth hidden layer neuron when qth pattern is presented.

Step 5: Compute the weighted sum of inputs for the individual neurons in the output layer as

$$Net^L_{qk} = \sum_{j=1}^{M} W^L_{kj} I^H_{qj} + \Theta^L_k ; k = 1, P \qquad \text{(A3)}$$

where superscript L refers to the output layer and M denotes the number of hidden layer neurons. W^L_{kj} is the connection weight between neuron k in the output layer and neuron j in the hidden layer, P denotes the number of neurons in the output layer and Θ^L_k represents the bias weight for kth output layer neuron

Step 6: Transform the weighted sum using the same activation function as in Step 4 to get the outputs of the output layer neurons as follows.

$$I^L_{qk} = \frac{1}{1 + \exp(-Net^L_{qk})} ; k = 1, P \qquad \text{(A4)}$$

where I^L_{qk} refers to the output of the kth output layer neuron.

Step 7: Obtain the error δ^L_{qk} associated with each neuron in the output layer as

$$\delta^L_{qk} = (Y_{qk} - I^L_{qk})[I^L_{qk}(1 - I^L_{qk})]; k = 1, P \qquad \text{(A5)}$$

where Y_{qk} = desired output of neuron k for pattern q and I^L_{qk} = actual output of the kth output layer neuron.

Step 8: Compute the error δ^H_{qj} associated with each neuron in the hidden layer by

$$\delta^H_{qj} = \sum_{k=1}^{P} \delta^L_{qk} W^L_{kj} [I^H_{qj}(1 - I^H_{qj})], j = 1, M \qquad \text{(A6)}$$

Step 9: Update the weights between output and hidden layer as

$$W^L_{kj}(t+1) = W^L_{kj}(t) + \eta \delta^L_{qk} I^L_{qj} + \alpha[W^L_{kj}(t) - W^L_{kj}(t-1)]; j = 1, M \qquad \text{(A7)}$$

where training iteration number is represented by t and η denotes the learning coefficient ($0 < \eta < 1$). The third term on the right hand side is referred to as momentum term where α denotes the momentum coefficient ($0 < \alpha < 1$). The addition of momentum term speeds up the training process and also helps to avoid local minima on the error surface.

Step 10: Update the weights between the hidden and input layer as given below and using another input vector implement steps 2 to 10.

$$W^H_{ji}(t+1) = W^H_{ji}(t) + \eta \delta^H_{qj} x_{qi} + \alpha[W^H_{ji}(t) - W^H_{ji}(t-1)]; \quad i = 1, N \qquad \text{(A8)}$$

In this procedure, Steps (2–6) and Steps (7–10) correspond to the forward and reverse passes respectively. The procedure barring Step 1 is repeated for all the input patterns a large number of times until the network weights are optimized such that the RMS error with respect to the test set i.e.,

$$E_{tst} = \sqrt{\sum_{q=1}^{N_{tst}}\sum_{k=1}^{P}\frac{(y_{qk} - I^L_{gk})^2}{N_{tst}P}} \tag{A9}$$

reaches a minimum. Here, N_{tst} signifies the number of test set patterns. Note that the bias neuron always possesses an output of "1" and weights Θ^H_j and Θ^L_k are adjusted in a similar fashion to the hidden and output layer weights.

Appendix-II

Procedure for computing the Magnitude of Yearly Monsoon Rainfall in (mm)

Here, we will illustrate the computational procedure for computing rainfall from the known parameter values. Consider that the values of *Ridge* (= 14.) and *DSLP* (= 4.4) are available and the task is to predict the rainfall using the weights listed in Tables 2a, b. To that end we may proceed as described below:

1: Normalize the *RIDGE* and *DSLP* values to get the two network inputs x_1 and x_2. Note that in the following computations, the subscript representing the pattern index q has been dropped.

$$\begin{aligned} RIDGE = x_1 &= \frac{(14 - 10.45)}{(19.215 - 10.45)} \\ &= 0.405020 \\ DSLP = x_2 &= \frac{(4.4 + 0.21)}{(5.985 + 0.21)} \\ &= 0.744148 \end{aligned}$$

where 19.215 and 10.45 are the scaling parameters for the Ridge and the corresponding parameters for DSLP are 5.985 and 0.21, respectively.

2: Compute the weighted-sums of inputs for the six hidden nodes according to Eq. A1 as

$$\begin{aligned} Net^H_1 &= -0.294163 \times 0.405020 + 0.319022 \times 0.744148 \\ &\quad - 0.406677 = -0.2884193 \\ Net^H_2 &= -0.435573 \times 0.405020 - 0.548771 \times 0.744148 \\ &\quad - 0.435841 = -1.0206236 \\ Net^H_3 &= 0.168131 \times 0.405020 - 0.569852 \times 0.744148 \\ &\quad - 0.740890 = -1.0968478 \\ Net^H_4 &= -1.43422 \times 0.405020 + 1.16194 \times 0.744148 \\ &\quad - 0.434811 = -0.1510434 \\ Net^H_5 &= -1.69509 \times 0.405020 + 1.02023 \times 0.744148 \\ &\quad - 0.401620 = -0.3289632 \\ Net^H_6 &= -0.109301 \times 0.405020 - 0.0324836 \times 0.744148 \\ &\quad + 0.3790371 = 0.3105953 \end{aligned}$$

Here, Net^H_j denotes the weighted-sum for jth hidden layer neuron for the input vector defined in Step 3.

3: Compute the outputs of the hidden nodes according to Eq. A2 as:

$$I^H_1 = \frac{1}{1 + e^{-(-0.2884194)}} = 0.4283908 \qquad I^H_2 = \frac{1}{1 + e^{-(-1.0206236)}} = 0.2649059$$

$$I^H_3 = \frac{1}{1 + e^{-(-1.0968478)}} = 0.2503309 \qquad I^H_4 = \frac{1}{1 + e^{-(-0.1510434)}} = 0.4623107$$

$$I^H_5 = \frac{1}{1 + e^{-(-0.3289632)}} = 0.4184929 \qquad I^H_6 = \frac{1}{1 + e^{-(0.3105953)}} = 0.5770305$$

4: Compute the weighted-sum of the inputs of the output layer node according to Eq. A3 as

$$\begin{aligned} Net^L_1 &= (-0.484905 \times 0.4283908) \\ &\quad + (0.389519 \times 0.2649059) \\ &\quad + (0.978241 \times 0.2503309) \\ &\quad - (1.21877 \times 0.4623107) \\ &\quad - (1.89753 \times 0.4184929) \\ &\quad + (0.342731 \times 0.5770305) + 0.742981 \\ &= -0.2764651 \end{aligned}$$

5: The output of the single output node is evaluated as

$$I^L_1 = \frac{1}{1 + e^{(0.2764651)}} = 0.4313206$$

6: Rescale the network output to finally get the predicted rainfall

$$\begin{aligned} \text{predicted Rainfall} &= 0.4313206 \times 451.055 + 620.255 \\ &= 814.804(\text{mm}) \end{aligned}$$

where 451.055 and 620.255 are the scaling parameters for ISMR. Note that while network training the actual magnitudes of ISMR must be scaled down using these scaling parameters so as to lie between zero and 1.

References

Banerjee, A. K., Sen, P. N., Raman, C. R. V., 1978: On foreshadowing southwest monsoon rainfall over India with midtropospheric circulation anomaly of April. *Indian J. Meteorol. Hydrol. Geophys.*, **29**, 425–431.

Basu, S., Andharia, H. I., 1992: The chaotic time series of Indian monsoon rainfall and its prediction. *Proc. Indian. Acad. Sci. (Earth Planet. Sci.)*, **101**, 27–34.

Beale, R., Jackson, T., 1990: *Neural Computing: An Introduction.* Bristol, UK: Adam Hilger.

Bhalme, H. N., Jadhav., S. K., Mooley, D. A., Ramana Murty, Bh V, 1986: Forecasting of monsoon performance over India. *J. Climatol.*, **6**, 347–354.

Bishop, C. M., 1994: Neural Network and their applications *Rev. Sci. Instr.*, **65**, 1803–1832.

Blanford, H. F., 1884: On the connection of the Himalaya snowfall with dry winds and seasons of droughts in India. *Proc. Roy. Soc. London,* **37**, 3.

Chen, S., Billings, S. A., 1992: Neural networks for nonlinear dynamic system modelling and identification. *Int. J. Control,* **56**, 319–346.

Derr, V. E., Slutz, R. J., 1994: Prediction of El-Nino events in the Pacific by means of neural networks. *AI Applications,* **8**, 51–63.

Dugam, S. S., Kakade, S. B., Verma, R. K., 1993: A new predictor for long-range forecasting of Indian summer monsoon. *Vatavaran,* **15**, 6–9.

Elliot, W. P., Angell, J. K., 1987: The relation between Indian monsoon rainfall, the southern oscillation, and hemispheric air and sea temperature 1884–1984. *J. Climate Appl. Meteor.,* **26**, 943–948.

Elliot, W. P., Angell, J. K., 1988: Evidence for changes in Southern Oscillation relationships during the last 100 years. *J. Climate,* **1**, 729–737.

Elsner, J. B., Tsonis, A. A., 1992: Nonlinear prediction, chaos, and noise, *Bull. Amer. Meteor. Soc.,* **73**, 49–60.

Freeman, J. A., Skapura, D. M., 1991: *Neural Networks: Algorithms, Applications and Programming Techniques.* Reading, MA: Addison-Wesley.

Fu, C., Fletcher, J., 1988: Large signals of climatic variations over the ocean in the Asian monsoon region. *Adv. Atmos. Sci.,* **5**, 389–404.

Goswami, P., Srividya, 1996: A novel neural network design for long-range prediction of rainfall pattern. *Curr. Sci.,* **70**, 447–457.

Gowariker, V., Thapliyal, V., Sarkar, R. P., Mandal, G. S., Sikka, D. R., 1989: Parametric and power regression models: New approach to long range forecasting of monsoon rainfall in India. *Mausam,* **40**, 115–122.

Gowariker, V., Thapliyal, V., Kulashrestha, S. M., Mondal, G. S., Sen Roy, N., Sikka, D. R., 1991: A power regression model for long range forecast of southwest monsoon rainfall over India. *Mausam,* **42**, 125–130.

Grieger, B., Latif, M., 1994: Reconstruction of the El-Nino attractor with Neural Networks. *Clim. Dyn.,* **10**, 267–276.

Hastenrath, S., 1986: On climate prediction in the Tropics. *Bull. Amer. Meteor. Soc.,* **67**, 696–702.

Hastenrath, S., 1987: On the prediction of Indian Monsoon Rainfall anomalies. *J. Climate Appl. Meteor.,* **26**, 847–857.

Hastenrath, S., 1995: Recent advances in tropical climate prediction. *J. Climate,* **8**, 1519–1532.

Haykin, S., 1994: *Neural Networks: A Comprehensive Foundation.* New York: Macmillan.

Hecht-Nielsen, R., 1990: *Neurocomputing.* Reading, MA: Addison-Wesley.

Huang, S. C., Huang, Y. F., 1991: Bounds on the number of hidden neurons in multilayer perceptrons. *IEEE Transactions on Neural Networks,* **2**, 47–55.

Krishna Kumar, K., Rupakumar, K., Pant, G. B., 1992: Premonsoon ridge location over India and its relation to monsoon rainfall. *J. Climate,* **5**, 979–986.

Krishna Kumar, K., Rupakumar K., Pant, G. B., 1995: Premonsoon maximum/minimum temperatures over India in relation to the summer monsoon rainfall (Unpublished, copy available from the authors.)

Lippmann, R. P., 1987: An introduction to computing with neural nets. *IEEE ASSP Mag.,* **4**, 4–22.

McCann, D. W., 1992: A neural network short-term forecast of significant thunderstorms. *Weather and Forecasting,* **7**, 525–534.

Muller, B., Reinhardt, J., 1990: *Neural Networks.* Berlin: Springer.

Minns, A. W., Hall, M. J., 1996: Artificial Neural Networks as rainfall-runoff models. *J. Hydrol.,* **41**, 399–417.

Mooley, D. A., Parthasarathy, B., 1984: Fluctuations in all-india summer monsoon rainfall during 1871–1978. *Climatic Change,* **6**, 287–301.

Mooley, D. A., Shukla, J., 1987: Variability and forecasting of the summer monsoon rainfall over India. In: Chang, C. P., Krishnamurti, T. N. (eds.) *Monsoon Meteorology. (Oxford monogr. on Geol. and Geophys., No. 7)* New York: Oxford University Press, pp. 26–58.

Mooley, D. A., Paolino, D. A., 1988: A predictive monsoon signal in the surface level thermal field over India. *Mon. Wea. Rev.,* **116**, 339–352.

Mooley, D. A., Parthasarathy, B., Pant, G. B., 1986: Relationship between all-india summer monsoon rainfall and location of ridge at 500 mb level along 75° E. *J. Climate Appl. Meteor.,* **25**, 633–640.

Navone, H. D., Ceccatto, H. A., 1994: Predicting Indian monsoon rainfall: A Neural Network approach. *Climate Dynamics,* **10**, 305–312.

Parker, D. B., 1982: Learning Logic. Invention Rep. 581-564. Stanford University, CA.

Parthasarathy, B., 1984: Interannual and long-term variability of Indian summer monsoon rainfall. *Proc. Indian Acad. Sci. (Earth Planet. Sci.),* **93**, 371–385.

Parthasarathy, B., Pant, G. B., 1985: Seasonal relationship between Indian summer monsoon rainfall and Southern Oscillation. *J. Climatol.,* **5**, 369–378.

Parthasarathy, B., Diaz, H. F., Eischeid, J. K., 1988: Prediction of all-India summer monsoon rainfall with regional and large-scale parameters. *J. Geophys. Res.,* **93/D5**, 5341–5350.

Parthasarathy, B., Rupakumar, K., Sontakke, N. A., 1990: Surface and upper air temperatures over India in relation to monsoon rainfall. *Theor. Appl. Climatol.,* **42**, 93–110.

Parthasarathy, B., Rupakumar, K., Munot, A. A., 1991: Evidence of secular variations in Indian monsoon rainfall-circulation relationships. *J. Climate* **4**, 927–938.

Parthasarathy, B., Rupakumar, K., Kothawale, D. R., 1992: Indian summer monsoon rainfall indices: 1871–1990. *Meteorol. Mag.,* **12**, 174–186.

Parthasarathy, B., Rupakumar, K., Munot, A. A., 1993: Homogeneous Indian Monsoon rainfall: Variability and prediction. *Proc. Indian Acad. Sci. (Earth Planet. Sci.),* **102**, 121–155.

Parthasarathy, B., Munot, A. A., Kothawale, D. R., 1994: All-India monthly and seasonal rainfall series: 1871–1993. *Theor. Appl. Climatol.,* **49**, 217–224.

Parthasarathy, B., Sontakke, N. A., Munot, A. A., Kothawale, D. R., 1987: Droughts/Floods in the summer monsoon season over different Meteorological subdivisions of India for the period 1871–1984. *J. Climatol.,* **7**, 57–70.

Peak, J. E., Tag, P. M., 1989: An Expert system approach for prediction of maritime visibility obscuration. *Mon. Wea. Rev.,* **117**, 2641–2653.

Peak, J. E., Tag, P. M., 1992: Towards automated interpretation of satellite imagery for navy shipboard applications. *Bull. Amer. Meteor. Soc.,* **73**, 995–1008.

Peak, J. E., Tag, P. M., 1994: Segmentation of satellite imagery using Hierarchical Thresholding and Neural Networks. *J. Appl. Meteor.,* **33**, 605–616.

Prasad, K. D., 1992: Large-scale features and long-range predictions of the Indian summer monsoon. *Ph.D Thesis,* Poona University, Pune, pp. 287.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986a: *Parallel Distributed Processing,* Vol. **1**, Cambridge, MA: MIT Press, pp. 318–362.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986b: Learning representations by back propagating errors. *Nature,* **323**, 533–536.

Schalkoff, R. J., 1992: *Pattern Recognition: Statistical, Structural and Neural approaches.* New York: John Wiley.

Shukla, J., 1987: Long-range forecasting of monsoons. In: Fein, J. S., Stephens, P. L. (eds.) *Monsoons.* New York: Wiley and Sons, pp. 523–547.

Shukla, J., Paolino, D. A., 1983: The Southern Oscillation and long-range forecasting of the summer monsoon rainfall over India. *Mon. Wea. Rev.,* **111**, 1830–1837.

Shukla, J., Mooley, D. A., 1987: Empirical prediction of the summer monsoon rainfall over India. *Mon. Wea. Rev.* **115**, 695–703.

Tambe, S. S., Kulkarni, B. D., Deshpande, P. B., 1996: *Elements of Artificial Neural Networks with Selected Applications in Chemical Engineering and Chemical & Biological Sciences.* Louisville: Simulation and Advanced Controls.

Tang, B., Flato, G. M., Holloway, G., 1994: A study of Arctic sea ice and sea-level pressure using POP and Neural Network methods. *Atmos. Ocean,* **32**, 507–529.

Thapliyal, V., 1981: ARIMA model for long-range prediction of monsoon rainfall in Peninsular India. *India Meteorological Department Monograph Climatology* No. 12/81.

Thapliyal, V., 1982: Stochastic dynamic model for long-range prediction of monsoon rainfall in peninsular India. *Mausam,* **33**, 399–404.

Thapliyal, V., 1990: Large-scale prediction of summer monsoon rainfall over India: Evolution and development of new models. *Mausam,* **41**, 339–346.

Thomson, A. W. P., 1996: Non-linear predictions of Ap by Activity class and Numerical value. *PAGEOPH,* **146**, 163–193.

Tong, H., 1990: *Nonlinear Time Series: A Dynamic System Approach.* Oxford: Clarendon Press.

Tsonis, A. A., Elsner, J. B., 1988: The weather attractor over very short time scales. *Nature,* **333**, 545–547.

Tsonis, A. A., Elsner, J. B., 1989: Chaos, Strange attractors, and weather. *Bull. Amer. Meteor. Soc.,* **70**, 14–23.

Venkatasubramanian, V., Vaidyanathan, R., Yamamoto, Y., 1990: Process faults detection and diagnosis using neural networks-I. Steady state processes. *Compu. Chem. Engng.,* **14**, 699–712.

Verma, R. K., 1982: Long-range prediction of monsoon activity: A synoptic diagnostic study. *Mausam,* **33**, 35–44.

Walker, G. T., 1908: Correlation in seasonal variation of climate (Introduction). *Memoirs of India Meteorological Department,* **20**, 117–124.

Walker, G. T., 1910: On the meteorological evidence for supposed changes of climate in India. Mem. *India Meteorological Department,* **21**, 1–21.

Walker, G. T., 1933: Seasonal Weather and its prediction. *British Association for Advancement of Science,* Rep. **103**, 25–44. (Reprinted Annual Smithsonian Institute for 1935, 117–138).

Wasserman, P. D., 1989: *Neural Computing: Theory and Practice.* New York: Van Nastrand Reinhold.

Werbos, P. J., 1974: Beyond recognition, New tools for prediction and analysis in the behavioral sciences. Ph.D Thesis, Harvard Univ., Cambridge, MA.

Wright, P. B., 1975: An index of the southern oscillation. Climate Research Unit. Rep. No. CRUPR4. University of East Anglia, Norwich, UK, 20 pp.

Xu, L., Klasa, S., Yuille, A., 1992: Recent advances on techniques of static feedforward networks with supervised learning. *Int. J. Neural System,* **3**, 253.

Yoh-Han, Pao, 1989: *Adaptive Pattern Recognition and Neural Networks.* USA: Addison Wesley.

Zhang, M., Scofield, R. A., 1994: Artificial Neural Network techniques for estimating heavy convective rainfall and recognizing cloud mergers from satellite data. *Int. J. Remote Sensing, London,* **15**, 3241–3261.

Zupan, J., Gasteiger, J., 1993: *Neural Networks for Chemists.* New York: VCH.

Authors' addresses: C. Venkatesan and R. N. Keshavamurty, Indian Institute of Tropical Meteorology (IITM), Dr. Homi Bhabha Marg, Pashan, Pune 411008, India; S. D. Raskar, S. S. Tambe and B. D. Kulkarni, Chemical Engineering Division, National Chemical Laboratory (NCL), Dr. Homi Bhaba Marg, Pashan, Pune 411008, India.