Forecasting Monsoon Precipitation Using Artificial Neural Networks

Xiaodan Wu[1], Cao Hongxing (曹鸿兴)[2],
Andrew Flitman[1], Wei Fengying (魏凤英)[2] and Feng Guolin (封国林)[2]
[1]*School of Business Systems, Faculty of Information Technology, Monash University, Australia*
[2]*Chinese Academy of Meteorological Sciences, Beijing* 100081

(Received September 1, 2000)

ABSTRACT

This paper explores the application of Artificial Intelligent (AI) techniques for climate forecast. It presents a study on modelling the monsoon precipitation forecast by means of Artificial Neural Networks (ANNs). Using the historical data of the total amount of summer rainfall over the Delta Area of Yangtze River in China, three ANNs models have been developed to forecast the monsoon precipitation in the corresponding area one year, five–year, and ten–year forward respectively. Performances of the models have been validated using a 'new' data set that has not been exposed to the models during the processes of model development and test. The experiment results are promising, indicating that the proposed ANNs models have good quality in terms of the accuracy, stability and generalisation ability.

Key words: Forecasting, Monsoon precipitation, Artificial intelligent technique, Artificial neural networks

1. Introduction

The Delta area of the Yangtze River is one of the vital important industrial and agricultural areas in the Peoples' Republic of China. During the past, however, there often occur some meteorological disasters, such as the extraordinary rainfall in summer and the flood, unfortunately.

Disaster climate can claim for thousands of people's life and billions of dollars' loss in economy, as what happened during the floods in the summers of 1998 and 1999 in China. It is highly desirable that operationally accurate tools be available for predicting the occurrence of such a climate event so that measures can be taken to reduce or avoid its impact.

Conventionally, statistical analysis and human judgment are the main vehicles in operation to produce the monsoon precipitation forecast of which the accuracy is by no means very satisfactory. Aiming at the improvement, this paper explores the applications of the emerging information processing methodology, namely, Artificial Intelligent (AI) techniques. Applying Artificial Neural Networks (ANNs), one of the main branches of AI, operational models have been developed to forecast the monsoon precipitation over the Delta Yangtze for one–year, five–year, and ten–year forecast periods respectively. These models have been validated using "hold–out" observation data samples and shown promising results.

2. Experiment data

The original experimental data is the monthly precipitation during the monsoon season

(June, July, August) collected from ten weather observation stations distributed in the area of the Dealt Yangtze in China. The time span of the data series is 49 years dated from 1951 to 1999. Because the target output of a forecasting model to be developed in this paper will be the total amount of rainfall during the monsoon season, the original data points have been averaged over the entire monsoon season for each year to obtain a new data series of an annual monsoon precipitation.

The experimental data has been divided into three categories for the Neural Network modelling. The first category consists of the first 39 years of precipitation observations and is used to develop the forecast neural networks. To ensure the resultant forecast models have good quality in generalisation, this data set has further been subdivided, with 79% being used as training data and 21% being used as test data. Thus the networks will be trained using the training subset, with the network model being saved each time an improvement is registered on the test subset. The subdivision into training and test data categories has been performed using random sampling. The third data category consists of the last 10 years of data that is used as a validation data set. Thus none of the modelling techniques will be exposed to this data set during model development. The validation data is simply used to compare the models with the amount of the annual monsoon precipitation in a practical actual–use scenario. It is the forecast results against the validation data set which are of particular interest, as any statistical or neural network modelling approach can be applied to fit historical data with a good estimation criterion such as the R–square.

3. Modelling with artificial neural networks

3.1 *Artificial neural networks*

Artificial Neural Networks (ANNs) technique is one of the main branches of the so–called Artificial Intelligence family. It emulates or mimics the functions of the human brain. The components of an ANN are based on a simplistic mathematical representation of what people think biological neural networks look like. Much research has been conducted looking at the application of neural networks technology for classification and forecasting for various time series such as stock market pricing and customer demand forecasts and prediction of business failure (Selvaraj and Verma, 1998; Mohammadian, 1999). This paper investigates the development of a neural network system to forecast the monsoon precipitation over the Delta Yangtze area in China.

ANNs are nonlinear dynamic computational systems where, rather than relying on a number of pre–determined assumptions as in the case of statistical modelling, data is used to form the model. When using the ANN technique, model developers do not need to deal with the assumptions which are imposed by statistics and which limit their modelling ability. Furthermore, ANNs are capable of handling the noisy and approximate data, hence, are promising to be applied into the fields such as the climate forecast of which the data is typically non–linear, noisy and complicated in nature.

There exist many different architectures and learning algorithms for neural network models. Most applications utilise a three–layer back–propagation (BP) design, as illustrated in Fig 1.

In such a model, when the input neurons receive data (in our case the total amount of summer rainfall of previous years), a calculation is performed at each neuron, with a subsequent signal sent to each connected hidden neuron, which in turn passes a signal to each output neuron. The output layer then performs the evaluation (in our case the forward

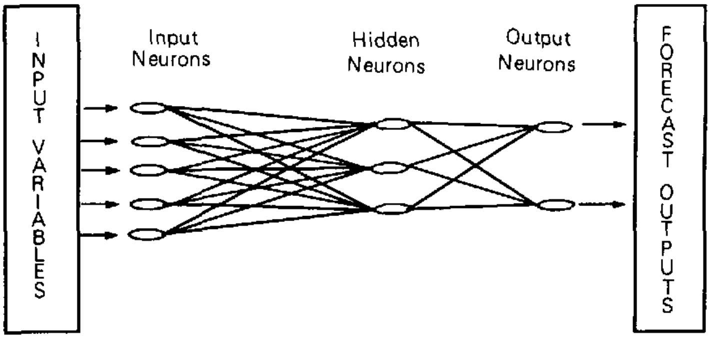

Fig. 1. Basic neural network architecture.

12–month, 5–year, or 10–year total amount of summer rainfall).

The neural network learns by using training data. Input attributes are supplied and resultant output is compared to the desired output. The network then adjusts the interconnection weights between layers. This process is repeated until the network performs well on the training set. The network can then be assessed on data not included in the training set, the test data, to estimate its performance.

The essential difference between neural networks and other forecasting techniques is that the neural networks use the training data to develop their representation for the modeled entity. This eliminates the situation associated with most models which must pre–determine assumptions about the modeled environment. This suggests that, in those cases where we are forced to make the most assumptions in order to model a problem using traditional models, neural networks may provide better results.

3.2 *Development of the neural network model*

3.2.1 *Back–propagation neural network modelling*

We first have experimented with the most popular and yet successfully used network design, the three–layer back–propagation (BP) architecture as sketched in Fig 1. Three models have been constructed for one–year, five–year, and ten–year forecasting time horizon respectively. The basic information fed into a model, i.e. the principle input variable of our neural network models, is the lagged precipitation observation. That is, for a one–year ahead forecast model, we use the precipitation of previous year as the input; for a five–year ahead forecast model we use the value of annual precipitation five years ago (Lag5y), and so forth.

Neural networks are like people. The more useful information provided the easier it is for them to understand a problem under consideration. When deciding on variables during the neural network development, it is usually better to use more variables than it is not enough. All of the ones that seem reasonable can be included in the initial model since neural networks can find subtle differences in data patterns that we human may not be able to discern. If a variable has no influence on the outcome, the network will learn to ignore it.

Additional input variables have therefore been created as listed below. They were derived from the raw observations existed to help the neural networks learn the underlying patterns of the data and make correct forecast.

- **C1y–Lag1y:** represents the change amount of the annual precipitation between two successive years with 1 year's lag; used for 1 year ahead forecast model.
- **C5y–Lag5y:** represents the change amount of the annual precipitation at five years' interval with 5 year's lag; used for 5–year ahead forecast model.
- **C10y–Lag10y:** represents the change amount of the annual precipitation at ten years' interval with 10 years' lag; used for 10–year ahead forecast model.
- **5YAve–Lag5y:** the mean of annual precipitation averaged over the last five years with 5 years' lag; used for 5–year ahead forecast model.
- **9YAve–Lag10y:** the mean of annual precipitation averaged over the last nine 9 years with 10–years' lag; used for 10–year ahead forecast model.
- **10YAve–Lag10y:** the mean of annual precipitation averaged over the last ten years; used for 10–year ahead forecast model.
- **5Yanomaly–Lag5y:** the average anomaly for last 5 successive years with 5 years' lag, used for 5–year ahead forecast model.
- **10Yanomaly–Lag10y:** the average anomaly for last 10 successive years with 10 years' lag, used for 10–year ahead forecast model.

Apart from the input variables, another important issue of concern in the neural network development is the mathematical function assigned to each calculation element (the neuron) in the network. While the connection weight will be modified during training of the network as observation patterns are passed along, activation functions should be decided before the network training. The selection of an activation function for the hidden layer is most important, since this is the layer that actually performs the feature extraction from the observation patterns processed. Accordingly, we have experimented with different forms of activation functions. It is not intended here to discuss *activation functions* in any detail; instead we refer the interested reader to Hertz et al. (1991).

Selection of the number of hidden neurons is yet another basic decision to be made in building a neural network. Heuristics have been suggested (Baum and Haussler, 1988). One must take care, however, when using them in practice, as the issue is case dependent and an inappropriate decision may degrade the accuracy of the network in generation (i.e. forecasting accuracy on new data). We have therefore experimented with the inclusion of different number of hidden neurons into the network to obtain an optimal model.

3.2.2 *Recurrent neural network modelling*

Considering what we are dealing with is a typical time series–forecasting problem, the appropriateness of using Recurrent Networks for modelling the precipitation forecast has also been investigated.

Recurrent networks are of the modified 3–layer backpropagation architecture. They are known for their ability to learn sequences, so they are excellent for time series data. In such network architecture, the user has a choice of feeding the input, hidden, or output layer back into the network for inclusion with the next pattern, which means that features detected in all previous patterns are fed into the network with each new pattern. Feeding the output layer into the input layer shows what the outputs of the previous patterns have been.

The main difference of a standard BP network and a recurrent network in structure is that there is one extra slab (a group of functionally similar neurons) in the input layer that is connected to the hidden layer just like the other input slab. This extra slab holds the connects of one of the layers as it existed when the previous pattern was trained. In this way the

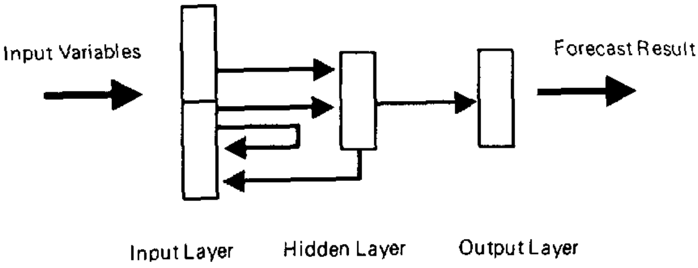

Fig. 2. A recurrent network.

network sees previous knowledge it had about previous inputs. This extra slab is sometimes called the network's long–term memory (Ward Systems Group, 1996).

There are different ways that can be used to develop a recurrent network. We have found that the most appropriate one for our problem is as follow:

A BP network with stand connections, as illustrated in Fig. 1, responds to a given input pattern with exactly the same output pattern every time the input pattern is presented. A recurrent network as shown in Fig. 2 may, however, respond to the same input pattern differently at different times, depending upon the patterns that have presented as inputs just previously. Thus, the sequence of the patterns is as important as the input pattern itself.

3.3 *Training the neural networks*

The networks are trained using basic supervised learning error–back–propagation. The input data was normalised in the range (−1,1) by applying a scaling function. Such scaling is essential as it minimises the effect of input magnitudes, and also aids the back–propagation learning algorithm.

For training of the stand back–propagation networks, the observations have been presented to the network in random order to minimise bias due to the network memorising the position of the training data. After every 200 complete pass of the training data through the network, the partially trained network has been tested using the test sub data set, and the average error in forecasting has been calculated and recorded. If there is an improvement in the forecasting accuracy, the network parameters were saved. Thus, we use the network performance in its processing of the test data rather than of the training data to determine the quality of the network. This approach helps to reduce the risk of saving a network that has memorised the features of training data, without the ability to generalise on new data.

The above training–testing process has continued until a number of successive tests using the test data subset yielded no further improvement. We have allowed for around 1000 tests on the test data without improvement, before concluding our training.

Recurrent networks are trained the same as standard back–propagation network except that patterns must always be presented in the same sequence without gaps in the data; random selection is not allowed. The network must also be tested with the patterns in sequence. This is because that the positions of the observation patterns play an important role in network learning as mentioned previously. We have used the last portion of the training data as the test subset.

3.4 *Validation of the network*

Once the training session of a network model has been completed, validation of the model is carried out using the validation data set. This set, represented the observations of the monsoon precipitation in last ten consecutive years, has not been exposed to the network at all during its training and test. The outputs of the neural network in processing this data set, compared with the actual observed values, represent a type of objective assessment on the performance of the forecasting model developed. A number of key statistics have been calculated in determining the quality of each resultant neural network model: *R-squared, Mean Squared Error, Mean Absolute Error, Min. Absolute Error, Max. Absolute Error, Correlation Coefficient.*

4. Results and discussion

The aim of our modelling task is to forecast the monsoon precipitation. Two kinds (in terms of architecture) three categories (in terms of time horizon) of neural network models have been developed which can respectively be used to carry out the forecast 1-year, 5-year, and 10-year forward. Obviously the accuracy is significant to characterize the success of a resultant model. Other aspects such as the model's generalization ability and error stability are also important to be considered in assessing a model's quality. In order to make comparisons and to evaluate the overall performance of each model as a tool for predicting the monsoon precipitation, we use the following statistics as standard criteria for model assessment:

- **MAE,** Mean absolute error, it measures the average error rate with the best being nearest to 0;
- **STDEV** (standard deviation), which measures the dispersion of the errors that a estimation model generates around the mean value of these errors; the best is that nearest to 0;
- **R-Squared,** which measures the actual variation explained by the model (i.e. explanation capacity), ranging from 0 (none explained) to 1 (all explained).
- **RSQ** (correlation coefficient), which measures the correlation between forecast and actual time series, ranging from 0 (poor fitting to the model) to 1 (good fitting);

Based on the above criteria, the performance of each model in processing the training and the validation data has been evaluated and summaries are listed in Table 1.

Table 1 (a). Neural network forecasting result on training data

	Standard backpropagation networks			Recurrent networks		
	Model I (1 yr forward)	Model II (5 yr forward)	Model III (10 yr forward)	Model IV (1 yr forward)	Model V (5 yr forward)	Model VI (10 yr forward)
MAE	**0.025**	0.094	0.182	0.030	**0.088**	**0.098**
STDEV	**25.46**	58.97	112.86	26.18	**57.01**	**100.34**
RSQ	**0.956**	0.794	0.243	0.954	**0.802**	**0.556**
R-squared	**0.958**	0.713	0.135	0.953	**0.851**	**0.385**

Model I is a standard backpropagation neural network used for 1-year ahead forecast. It consists of 4 layers and has 2 input variables Lag1y, C1y-Lag1y.

Model II is a standard backpropagation neural network used for 5-year forward forecast. It consists of 4 layers and has 2 input variables Lag5y, C5y–Lag5y.

Model III is a standard backpropagation neural network used for 10–year forward forecast. It consists of 4 layers and has 4 input variables Lag10y, C10y–Lag10y, 10YAve–Lag10y, 9YAve–Lag10y.

Model IV is a recurrent neural network used for 1–year forward forecast. It has 3 layers and 2 input variables Lag1y, C1y–Lag1y.

Table 1 (b). Neural network forecasting result on validation data

	Standard backpropagation networks			Recurrent networks		
	Model I (1 yr forward)	Model II (5 yr forward)	Model III (10 yr forward)	Model IV (1 yr forward)	Model V (5 yr forward)	Model VI (10 yr forward)
MAE	0.066	0.106	0.250	**0.043**	**0.098**	**0.150**
STDEV	74.21	97.41	195.61	**28.57**	**69.51**	**118.03**
RSQ	0.956	**0.951**	**0.454**	**0.992**	0.896	**0.836**
R–squared	0.885	0.738	0.050	**0.985**	**0.825**	**0.648**

(Best results in bold font)

Model V is a recurrent neural network used for 5–year forward forecast. It consists of 2 layers and has 3 input variables Lag5y, C5y–Lag5y.

Model VI is a recurrent neural network used for 10–year forward forecast. It consists of 3 layers and has 4 input variables Lag10y, C10y–Lag10y, 10YAve–Lag10y, 9YAve–Lag10y.

It is important to understand the relative significance of the results on the Training / Test data sets and the Validation data set. Since the former sets are used to develop the models, it is then usual for the results on these sets to be on the optimistic side, as can be seen from the above tables. Therefore the results from the validation set will be by far more significant. However, looking at both result sets can provide useful insight.

The training set results in Table 1(a) show that the quality of both standard BP and recurrent networks are comparable with the recurrent models performed marginally better than the standard BP models. However, larger differences can be detected, particularly for the long–term forecasting, from the validation results in Table 1(b). Bearing in mind that validation results are more significant and that long–term forecasting is usually much harder than the short–term one, we can clearly see that the recurrent networks outperformed the standard BP networks in terms of forecast accuracy (indicated by lower MAE), error structure stability (indicated by lower STDEV), higher correlation between the forecasted and actual precipitation (measured by RSQ), and better model explanation capacity (measured by higher R–squared).

Further more, by comparing the validation and training results horizontally within a single model, we see that the recurrent networks show less degradation in quality against all assessment measures from the training to the validation data processing, compared with what achieved by their standard BP–network counterparts. This indicates that the recurrent network forecast models have better generalization ability.

We have therefore decided using the recurrent neural networks as our forecast models. The following tables and figures provide the actual monsoon precipitation and the corresponding forecast values produced by these forecast models.

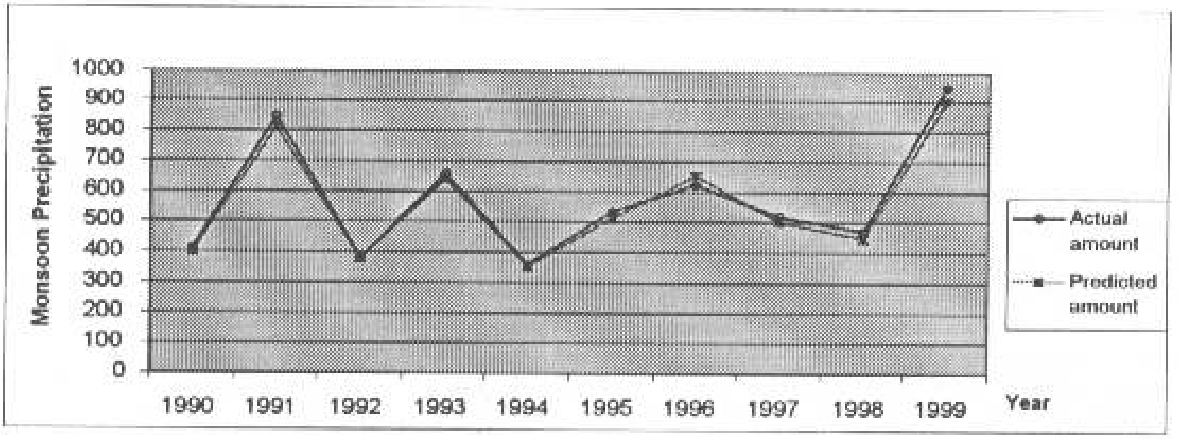

Fig. 3. 1-year forward forecast of neural network: validation set.

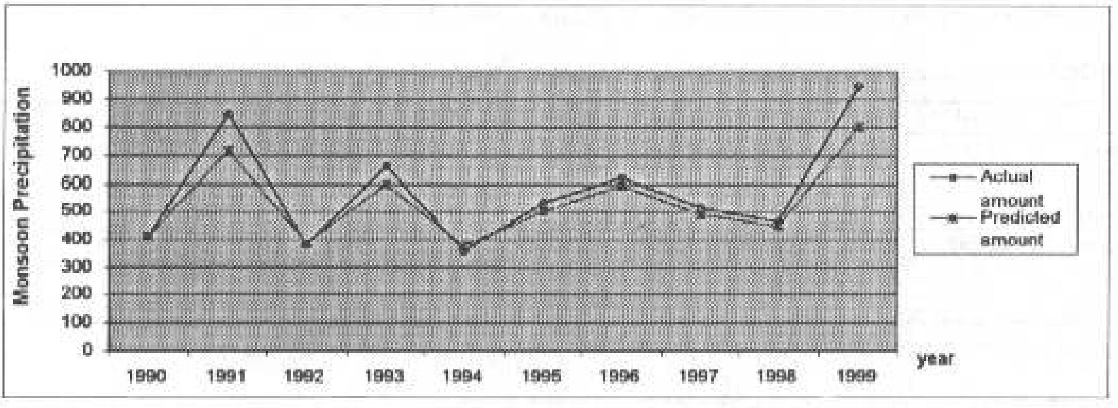

Fig. 4. 5-year forward forecast of neural network: validation set.

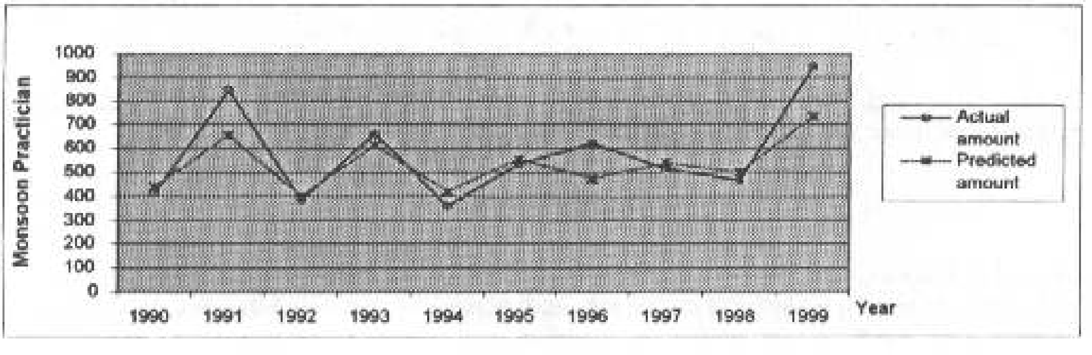

Fig. 5. 10-year forward forecast of neural network: validation set.

5. Conclusions

This paper explores the application of Artificial Intelligent Techniques to Climate forecasting. Artificial Neural Network technique has successfully been applied to develop models that are used to forecast the annual monsoon precipitation over the Delta Yangtze in China for respectively one year, five years, and ten years time horizon. These models have demonstrated of good quality in terms of forecast accuracy, error–structure stability, and generalization ability; especially when they were used to make forecast upon unseen validation data. Various neural network design issues have been discussed; different paradigms of artificial intelligent techniques, the fuzzy approach and neurofuzzy approach, have been investigated. The research clearly indicates that AI techniques are valuable tools for improving the climate forecast in operation.

Table 2(a). Delta Yangtze summer rainfall prediction 1–year forward

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Actual	413	847	382	660	361	534	622	516	469	949
Actual Anomaly	–	+	–	+	–	+	+	+	–	+
Forecast	398.65	813.06	376.68	642.01	352.95	512.7	653.28	500.92	447.41	902.74
Forecast Anomaly	–	+	–	+	–	–	+	–	–	+

Alignment between the Actual and Predicted (validation data) Anomaly: 80%

Table 2(b). Delta Yangtze summer rainfall prediction 5–year forward

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Actual	413	847	382	660	361	534	622	516	469	949
Actual Anomaly	–	+	–	+	–	+	+	+	–	+
Forecast	416.63	718.72	389.28	596.88	378.89	505.11	590.23	489.63	451.84	800.95
Forecast Anomaly	–	+	–	+	–	–	+	–	–	+

Alignment between the Actual and Predicted (validation data) Anomaly: 80%

Table 2(c). Delta Yangtze summer rainfall prediction 10–year forward

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Actual	413	847	382	660	361	534	622	516	469	949
Actual Anomaly	–	+	–	+	–	+	+	+	–	+
Forecast	442.14	657.53	397.13	615.38	421.95	552.74	472.62	541.87	498.6	739.02
Forecast Anomaly	–	+	–	+	–	–	–	–	–	+

Alignment between the Actual and Predicted (validation data) Anomaly: 70%

REFERENCES

Baum, E., and D. Haussler, 1998: What size net gives valid generalisation. *Neural Computation*, **1**, 151–160.

Flitman A., 1997: Numerical modelling of student failures. *Computers and Operations Research*, **24(4)**, 367–377.

Hertz, J., A. Korgh, and R. Palmer, 1991: *Introduction to the Theory of Neural Computation*, Addison–Wesley, USA.

Mohammadian M., (Ed), 1999: *Computational Intelligence for Modelling, Control & Automation*, IOS Press, Netherlands.

Selvaraj H., and B. Verma (Eds), 1998: *International Conference on Computational Intelligence and Multimedia Applications*, Australia.

Ward Systems Group, Inc., 1996: *Neural Network Products for the Next Century*, Ward Systems Group, Inc., USA.