An Application of Artificial Neural Networks for Rainfall Forecasting

Kin C. Luk, James E. Ball and Ashish Sharma
Water Research Laboratory, School of Civil Engineering, The University of New South Wales

Summary: Rainfall forecasting is important for many catchment management applications. Due to the complex nature of rainfall processes, however, it is not feasible to forecast rainfall based on a physical approach. With the advent of the digital computer and subsequent development of artificial intelligence approaches, data driven techniques such as the artificial neural networks (ANN), have emerged as an alternative to forecast rainfall time series. Presented herein are the results of a study investigating the application of ANN to forecast the spatial distribution of rainfall for an urban catchment. Three alternative types of ANN, namely multi-layer feedforward neural networks, partial recurrent neural networks and time delay neural networks were identified, developed and, as presented in this paper, found to provide reasonable predictions of the rainfall depth one time-step in advance. The data requirements for and the accuracy obtainable from these three alternative types of ANN are discussed.

1. INTRODUCTION

A challenging task for catchment management and flood management in particular is the provision of a quantitative rainfall forecast. Accurate forecasts of the spatial and temporal distribution of rainfall are useful for both water quantity and quality management. For example, a flash flood warning system may require a quantitative rainfall forecast to enhance flood prediction with an aim to increasing the lead time. Similarly, a rainfall forecast provides advance information for many water quality problems.

There are two possible approaches to rainfall forecasting. The first approach involves the study of the rainfall processes in order to model the underlying physical laws. However, this physical approach may not be feasible because

- rainfall is a complex dynamic system which varies both in space and time;
- even if the rainfall processes can be described concisely and completely, the volume of calculations involved may be prohibitive; and
- the data that is available to assist in definition of control variables for the models, such as rainfall intensity, wind speed, and evaporation, etc. are limited in both the spatial and temporal dimensions.

A second approach to forecasting rainfall is to apply a transformation to the input data for production of the desired output data. This transformation can be considered as a mapping between the inputs and outputs without a detailed consideration of the internal structure of the physical processes. This approach is essentially data driven. Since all important information is embedded in the data, an appropriate model therefore is developed to extract these essential features. This approach is considered appropriate for the present study because the main concern is to forecast short-duration rainfall at specific locations within a catchment. A thorough understanding of the physical laws usually is not required, and the data requirements are not as extensive as for a process model.

Artificial neural networks (ANN), which emulate the parallel distributed processing of the human nerve system, have proven to be very powerful in dealing with complicated problems, such as, function approximation, pattern recognition and time series prediction. ANN, therefore, were adopted in the present study to simulate the complex mapping of the rainfall time series.

There have been a number of reported studies that have used ANN to solve problems in hydrology. For example, French et. al. [1992] used an ANN to forecast rainfall for a catchment with artificial rainfall inputs, while Hsu et. al. [1995] applied an ANN to model the rainfall-runoff process. However, in many of the previous studies, only the most common type of ANN was used, namely, the multi-layer feedforward network (MLFN). Although the MLFN has proven to be a very powerful tool and amounts to 80% of practical applications in all fields of engineering and science, several alternative types of ANN have been developed recently for modelling temporal sequences. The specific alternatives appropriate for rainfall forecasting include the partial recurrent neural networks (PRNN) and time-delay neural networks (TDNN).

Presented in this paper are the results of an investigation of the alternative ANN (i.e. MLFN, PRNN and TDNN) for forecasting rainfall over an urban catchment in western Sydney. The ANN were used to forecast rainfall at multi-locations simultaneously for the next 15 minutes based on past spatial and temporal rainfall patterns. The important issues of identifying the order of lag and complexity of the networks are discussed also.

2. THE ARTIFICIAL NEURAL NETWORKS APPROACH

2.1 The Basics

An ANN is a computational approach inspired by studies of the brain and nerve systems in biological organisms. The powerful functions of a biological neural system is attributed to its parallel distributed processing by individual cells, known as neurons. In view of this, an ANN is constructed to emulate the structure of the biological neural system by distributing computations to small and simple processing units, called artificial neurons, or nodes. An ANN consists of layers of nodes connected together. A simple 3-layer ANN, as shown in Figure 1, is used to illustrate the basic structure and terminology of an ANN.

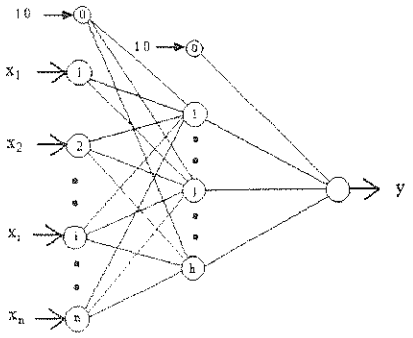


Figure 1 - A simple 3-layer feedforward neural network

There are basically three types of layers. The first layer connecting to the input variables is called input layer. The last layer connecting to the output variables is called output layer. Layers in between the input and output layers are called hidden layers; there can be more than one hidden layers. Information is transmitted through the connections between nodes. In a simple situation, information is passed forward only, as shown in Figure 1. This type of network is called a feedforward network, or multi-layer feedforward network (MLFN).

It was shown by Hornik et. al. [1989] that given enough hidden nodes, a MLFN can approximate any continuous function to any desired degree of accuracy. Mathematically, a three-layer MLFN with n input nodes, h hidden nodes, and one output node, can be expressed in the following formula:

$$y = s_1\left(\sum_{j=1}^{h} O_j \cdot w_j + w_0\right) \qquad (1)$$

Where y is the output,

O_j , is the output value of the j^{th} hidden node:

$$O_j = s_2\left(\sum_{i=1}^{n} x_i \cdot w_{ij} + w_{0j}\right) \qquad (2)$$

x_i are the inputs,

w_j are the connection weights between nodes of the hidden and output layer,

w_{ij} are the connection weights between nodes of the hidden and the input layer

x_0=1.0 is a bias and w_0 and w_{0j} are the weights for the biases, (the biases are used to prevent the error surface from passing through the origin at all times.)

s_1 and s_2 are activation functions. The most commonly used activation function is a logistic sigmoid function:

$$s(x) = \frac{1}{1+e^{-x}} \qquad (3)$$

From equation (1), it can be seen that a MLFN is a general-purpose, nonlinear model. The main parameters of this model are the connections weights.

The estimation of parameters is known as training where optimal connection weights are determined by minimising an objective function. The sum of squared error (SSE) between the network outputs and the desired outputs is a popular objective function, and was adopted in this study.

2.2 ANN for Rainfall Forecasting

The aim of the present study is to forecast rainfall for the next time-step. The rainfall output vector **X**(t+1) is assumed to be related to previous rainfall vectors, **X**(t), ..., **X**(t-k+1) using a general nonlinear model structure:

$$\mathbf{X}(t+1) = g(\mathbf{X}(t), \mathbf{X}(t-1), \mathbf{X}(t-2), .. \mathbf{X}(t-k+1)) + e(t) \qquad (4)$$

Where the dimension of the vector **X** represents the rainfall in space, g( ) is an unknown nonlinear mapping function, e(t) is an unknown mapping error (to be minimised), and k is the (unknown) number of past inputs contributing to rainfall at the next time-step. Sometimes, k is refers to the lag of the model. If k=1, the future rainfall is related only to the present rainfall, thus giving a lag-1 model.

According to equation (4), the development of an ANN for rainfall forecasting involves the following important considerations:

- Select an appropriate ANN to represent the recursive dynamic rainfall system.
- Estimate the order of lag for the ANN, i.e. to determine the number of past rainfall values to be included in the inputs
- Determine the optimal complexity of the ANN appropriate to the problem, i.e. to determine the number of hidden layers, and number of nodes in a hidden layer.

2.3 Alternative Networks

Three alternative types of ANN are identified appropriate for rainfall forecasting and adopted in this study for comparison. They are:

- Multi-layer Feedforward Network (MLFN)
- Partial Recurrent Neural Network (PRNN)

- Time Delay Neural Network (TDNN)

A MLFN is used because this type of network is good at pattern recognition and function approximation. The theory of the MLFN is well understood. There are many successful applications in various fields of engineering and science.

When a MLFN is used to model rainfall time series, a correct order of lag for the MLFN needs to be determined. The determination of the order of lag may involve a lengthy process of trial and comparison. To overcome this problem, a partial recurrent network (PRNN) is identified. The main feature of a PRNN is the inclusion of a set of feedback connections which allow information passing backward. The recurrence created by the feedback connections lets the PRNN retrain information of the previous time steps. The temporal structure of the rainfall series is internally represented by the feedback connections. As such, the order of lag is not required to be specified explicitly.

A rainfall time series usually contains local features, such as isolated peaks between prolonged low values of rainfall. These local features do not have a fixed position in time, rendering the prediction of their occurrence extremely difficult. In view of this, a Time Delay Neural Network (TDNN) is adopted. The TDNN was originally developed by Waibel [1989] for phoneme recognition. The main feature of a TDNN is the recognition of local features within a larger pattern, independent of the positions of the local features. The TDNN is essentially a feedforward network, but the connections between layers are modified. Essentially, the outputs of a layer are buffered several time steps and then connected to the next layer. This method enables local short duration features be formed at the lower layer and more complex longer duration features at the higher layer.

Following is a more detailed description of each type of network. They are illustrated in the context of rainfall forecasting.

A. MLFN

Presented in Figure 2 is a generic structure of a MLFN designed for rainfall forecasting.

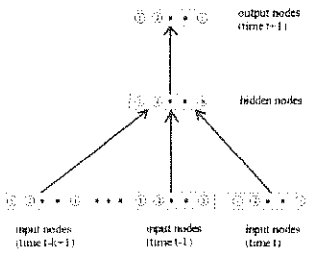


Figure 2 - A MLFN for Rainfall Forecasting

The output nodes of the network are rainfall of the next time step, which contain j elements, representing the spatial locations of rainfall. For example, if one wants to forecast rainfall at 16 points in space, j is equal to 16. The number of hidden nodes (h), which defines the complexity of the network, is a key variable to be estimated. Note that the number of hidden layer can be more than one. The input layer contains k batch of input nodes. The k is referred to as the lag of the network and is another key variable to be determined.

B. PRNN

The most popular PRNN called Elman network [1990] is adopted in this study. This network was originally developed for learning temporal language structure. It was subsequently found very useful in modelling time series. Figure 3 shows the structure of an Elman network for rainfall forecasting.

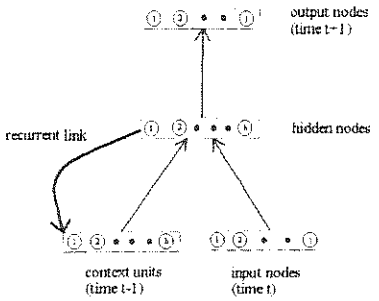


Figure 3 - An Elman Network for Rainfall Forecasting

The basic structure of the Elman network is feedforward. However, one special set of context units are included to receive feedback signals. The function of the context units is to store information of previous time steps. To achieve this aim, the context units make a copy of the activation of hidden nodes in the previous time step. Thus, at time t the context units have some signals of the state of network at time t-1. As a result, the whole network at a particular time depends on an aggregate of previous states as well as on the current input.

As shown in Figure 3, the input nodes contain j elements, representing the spatial dimensions of the rainfall. The number of hidden nodes is h, which is the same as the number of context units. The only variable to be specified is h.

One major advantage of the Elman network is that the temporal structure of rainfall is implicitly represented by the context units. The user does not need to pre-define the order of lag for the network.

C. TDNN

Presented in Figure 4 is a basic structure of a TDNN for rainfall forecasting. The main function of the TDNN is to detect features and their temporal relationship independent of their positions in time. For this purpose, each layers are structured into time frames. Two or more time frames in a lower level layer (e.g. input layer) are connected to a single time frame of the higher level layer (e.g. hidden layer). Thus, the higher level layer is able to abstract features at lower levels.

As an illustration, the TDNN shown in Figure 4 contains four time frames at input and three time frames at hidden layer. The time frames are combined to form a "window" to represent a duration in time.

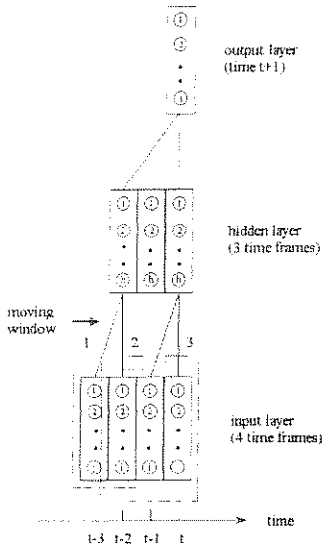


Figure 4 - A TDNN with 3 moving windows at inputs

There are totally three moving windows at the input, corresponding time delay of 1, 2 and 3 (as shown in dotted lines). Each node in the hidden layer is connected to a window of 2 time-frame of input nodes. The output is obtained by integrating (summing) the information over 3 time-frame windows in the hidden layer.

In this TDNN, the hidden nodes are able to detect local features within the range of the three delays. A shift in position of the features at the input can be detected by the hidden nodes and the output nodes.

The main variables of a TDNN need to be defined are: (1) numbers of time frames in input and hidden layers; (2) window size (the time delay); and (3) the number of hidden nodes.

3 APPLICATION TO A CATCHMENT

3.1 The Study Catchment

The Upper Parramatta River Catchment, shown in Figure 5, is located in the western suburbs of Sydney, Australia. Total area of the catchment is approximately 112 km^2. Within the upper catchment area, the dominant land use is typical of urban environments with a mix of residential, commercial, industrial and open space (parkland) areas.

Considerable development has occurred with the catchment over the past two decades which has resulted in an increase in the frequency of recorded flood levels. To mitigate the social and economic losses associated with flood events in this catchment, the Upper Parramatta River Catchment Trust (UPRCT) was instituted in 1989 with the role of managing flood mitigation measures within the catchment area.

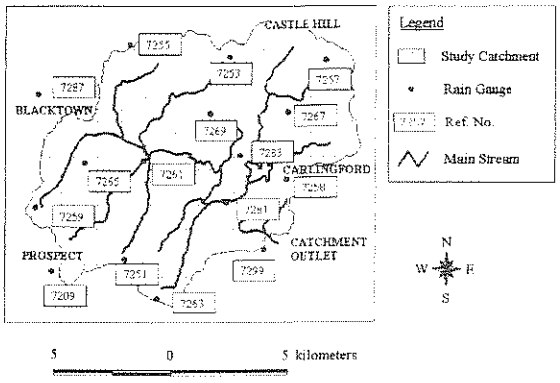


Figure 5 - The Upper Parramatta River Catchment

There are sixteen (16) continuous rain gauges within the Upper Parramatta River Catchment Areas; locations of these gauges are shown in Figure 5. The majority of these gauges have been installed by the UPRCT since its formation. Consequently, long-term records are not available form these gauges.

Records of the 16 rain gauges were obtained from January 91 to September 96. During this period, 34 storm events were extracted. The criterion for extraction was that the daily rainfall total was greater than 20 mm. The data series were in 15-minute interval giving a total number of rainfall values of 1749.

Three types of ANN were used to forecast the future rainfall for the 16 gauges simultaneously.

3.2 Methodology

A critical issue in developing an ANN is generalisation. If an ANN properly learns the essential features of the data, the ANN is then said to achieve good generalisation. An ANN may, however, suffer from either under-fitting or over-fitting the training data. An ANN that is not sufficiently complex can fail to detect fully the features in a complicated data set, leading to under-fitting. An ANN that is too complex may fit the noise, not just the features, leading to over-fitting.

A popular technique to achieve generalisation is the early stopping method presented by Sarle [1995]. According to the early stopping method, the data was split into three sets, namely a training set, a validation set and a test set. The training set was used to train the network whereas the validation set was used to monitor, or test the performance of the network at regular stages during training. Training stopped when the error on the validate set reached a minimum. Finally, the performance of the network was evaluated on the test data set which had not involved in the training process.

Where appropriate, the networks in this study used a signmoid activation function in the hidden nodes, but a linear function in the output nodes. The use of a signmoid function was to enable non-linearity of the network. However, the signmoid function was not adopted in the output nodes because it would force the output to be bounded between 0.0 and 1.0. This required scaling of the output variable by a known maximum value. This was undesirable for rainfall forecasting because it might not be possible to set a priori reasonable maximum rainfall value. To overcome this situation, an identity (linear) function was used.

To achieve better performance and faster convergence in training, the data were transformed with the log function.

$$z = 0.5 * \log_{10}(x+1) \tag{5}$$

where x is the input.

The normalised mean squared error (NMSE), defined as follows was chosen as the performance indicator.

$$NMSE = \frac{\sum_o \sum_p (d_{op} - y_{op})^2}{\sum_o \sum_p (d_{op} - \bar{d}_{op})^2} \approx \frac{1}{OP\sigma^2} \sum_o \sum_p (d_{op} - y_{op})^2 \tag{6}$$

where O is the number of output nodes
P is the number of patterns
d_{op} is the target output
y_{op} is the network's output
σ^2 is the variance of the target outputs

In essence, the NMSE is the sum of squared errors (SSE) normalised by the number of testing patterns over all output nodes and the estimated variance of the data. The use of NMSE enables a comparison of results with different length of testing patterns. Noted also that a value of NMSE = 1 corresponds to simply predicting the average.

4. TEST RESULTS AND DISCUSSIONS

The early stopping technique was adopted. Accordingly, the 34 rainfall events were divided into three parts: 16 events (748 numbers of 15-minute rainfall values) were used for training, 8 events (376 rainfall values) for validation and 10 events (625 rainfall values) were hold out for testing.

The maximum epoch (cycle) for training was set at 1000. During training, the networks were validated at every 100 epoch. Training was stopped when the validation error reached its lowest value, or the training reached the maximum epoch. Finally, the networks were evaluated against the testing data, which had not involved in the training process.

Various network configurations were attempted in order to determine the effect of two key variables: (1) order of lag, and (2) number of hidden nodes. For the MLFN, the orders of lag tried were 1, 2, 3 and 4. The numbers of hidden nodes tried were 2, 4, 8, 16, 24, 32, 64, and 128. Networks with two-layer of hidden nodes were also attempted. For each order of lag, the network with minimum NMSE was selected and shown in Table 1.

For the Elman network, the order of lag was fixed at 1 because the network would learn the temporal structure implicitly. The numbers of context units tried were 2, 4, 8, 16, 24, 32 and 64. The network with minimum NMSE was selected and shown in Table 1.

For the TDNN, networks with 2, 3 and 4 input windows were tested. The results of the three TDNN were shown in Table 1.

Table 1 - Comparison of the Three Type of Networks

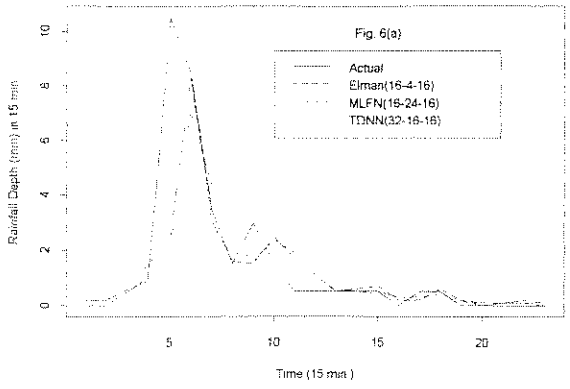
Network	Train (NMSE)	Validate (NMSE)	Test (NMSE)	Stopping epoch	Training Error at 1000 epoch (NMSE)
MLFN Lag 1 (16-24-16)	0.50	0.68	0.64	200	0.49
MLFN Lag 2 (32-8-16)	0.51	0.69	0.66	100	0.47
MLFN Lag 3 (48-4-16)	0.48	0.69	0.67	700	0.47
MLFN Lag 4 (64-2-16)	0.52	0.71	0.65	200	0.49
Elman (16-4-16)	0.49	0.67	0.64	300	0.48
TDNN Lag 2 (32-16-16)	0.50	0.67	0.63	100	0.41
TDNN Lag 3 (48-32-16)	0.50	0.69	0.64	100	0.41
TDNN Lag 4 (64-32-16)	0.51	0.69	0.65	100	0.40

Note: (1) The network configuration is denoted by three figures (x-y-z), where x= no. of input nodes, y = no. of hidden nodes and z = no. of output nodes
(2) The sigmoid activation function was used at both hidden and output nodes of the TDNNs. It was found that if a linear output activation function was used in the output nodes of a TDNN, large NMSE would result.

The following points were observed from the test results.

- All three types of networks had comparable performance.
- Networks with higher order of lag tended to over-learn the training data, resulting in smaller training errors, but larger test errors.
- Networks with lower order of lag required more hidden nodes, and vice visa.
- The MLFN with lower order of lag had slightly better performance than that of higher order of lag.
- The Elman network had comparable performance with the lag-1 MLFN and outperformed the MLFNs with higher order of lag.
- A lag-2 TDNN had the best performance. However, it was noted that the TDNNs required sigmoid activation function at the output nodes. Otherwise, very large NMSE would result.

Shown in Figure 6 are some of the better test results provided by the three types of networks.



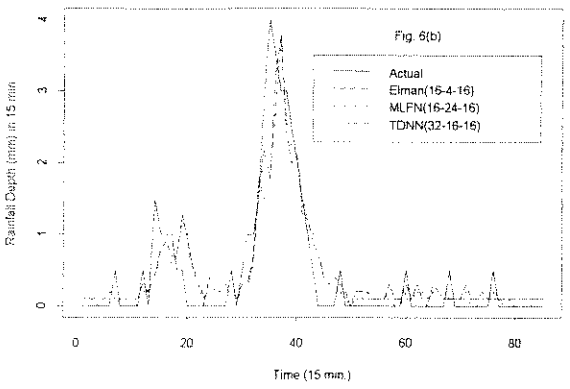


Figure 6 - (a) Forecasting Rainfall at Gauge No. 3 for the Storm Event on 2 Jan 96; (b) Forecasting Rainfall at Gauge No. 3 for the Storm Event on 6 Jan 96

In general, the three alternative types of networks provided reasonable predictions of the rainfall for the next time step. However, all the networks had difficulty in forecasting the peak values of rainfall. Improvement would be expected if more data were available for training the networks.

The networks with lower order of lag showed a better performance indicated that the rainfall series might not have a long time-dependence structure. Also, given the same number of hidden nodes, a network with higher lag contained more parameters, thus having a higher chance to over-learn the training data. Consequently, the networks with higher order of lag had smaller training error but larger testing error.

It seemed that there existed an optimal network complexity to cope with the complexity of the data. It was observed that the number of hidden nodes decreased as the order of lag increased, and vice versa. This fact might suggest that an optimal complexity of network needed to be maintained in order to achieve the best performance.

5. CONCLUSIONS AND RECOMMENDATIONS

The following conclusions were derived:

- Three alternative types of ANNs were set up for the Upper Parramatta River Catchment to forecast the spatial distribution of rainfall for the next 15 minutes. The procedure can be easily applied to other catchments.
- For each type of network, there existed an optimal complexity, which was determined by both the number of hidden node and the order of lag.
- The MLFN had comparable performance with more advanced networks.
- The TDNN and Elman network had good potential to model the dynamic structure of the rainfall process.

In summary, networks with simple structure, such as the Elman network, the lag-1 MLFN, and the TDNN with 2 input windows yielded better performance. The networks with higher order of lag and more hidden nodes tended to over-learn the training data. This was shown by the fact that they had small training errors but high testing errors. It was considered that more training data would be required to improve the performance of networks.

6. ACKNOWLEDGMENTS

The authors wish to acknowledge the Australian Research Council for financial support for the study. The authors also wish to acknowledge the Upper Parramatta River Catchment Trust for providing data for the study.

7. REFERENCES

Elman, J. L. (1990), Finding Structure in Time, *Cognitive Science*, 14, 179-211.

French, M., Krajewski, W. and Cuykendall, R. R. (1992), Rainfall Forecasting in Space and Time Using a Neural Network, *Journal of Hydrology*, 137, p.1-31.

Hsu, K. L., Gupta, V. and Soroshian, S. (1995), Artificial Neural Network Modeling of the Rainfall-Runoff Process, *Water Resources Research*, Vol. 31, No. 10, p. 2517-2530.

Hornik K., Stinchcombe M. and White. H. (1989), Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 2:359-366.

Sarle, W. S. (1995), Stopped Training and Other Remedies for Overfitting, *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics*, 352-360.

Waibel, A. (1989), Modular Construction of Time-Delay Neural Networks for Speech Recognition, *Neural Computation*, 1, 39-46.