SCIENTIFIC BRIEFING

HYDROLOGICAL PROCESSES
Hydrol. Process. **16**, 1325–1330 (2002)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/hyp.554

A data-driven algorithm for constructing artificial neural network rainfall-runoff models

K. P. Sudheer,[1]*
A. K. Gosain[2] and
K. S. Ramasastri[3]

[1] *National Institute of Hydrology, DRC, Kakinada 533 003, India*
[2] *Civil Engineering, Indian Institute of Technology, Delhi 110 016, India*
[3] *National Institute of Hydrology, Roorkee 247 667, India*

Correspondence to:
Dr K. P. Sudheer, National Institute of Hydrology, Siddartha Nagar, Kakinada 533 003, India.
E-mail:šudheer_nih@rediffmail.com

Abstract

A new approach for designing the network structure in an artificial neural network (ANN)-based rainfall-runoff model is presented. The method utilizes the statistical properties such as cross-, auto- and partial-auto-correlation of the data series in identifying a unique input vector that best represents the process for the basin, and a standard algorithm for training. The methodology has been validated using the data for a river basin in India. The results of the study are highly promising and indicate that it could significantly reduce the effort and computational time required in developing an ANN model. Copyright © 2002 John Wiley & Sons, Ltd.

Key Words neural network model; input vector; rainfall-runoff modelling

Introduction

In recent years, artificial neural networks (ANNs) have been used for forecasting in many areas of science and engineering. An ANN is an information processing system composed of many nonlinear and densely interconnected processing elements or neurons, which are arranged in groups called layers. The basic structure of an ANN usually consists of three layers: the input layer, where the data are introduced to the network; the hidden layer or layers, where data are processed; and the output layer, where the results of given input are produced. The interconnection between neurons is accomplished by using known inputs and outputs, and presenting these to the ANN in some ordered manner; this process is called training. The strength of these interconnections is adjusted using an error convergence technique so that a desired output will be produced for a known pattern. The main advantage of the ANN approach over traditional methods is that it does not require information about the complex nature of the underlying process under consideration to be explicitly described in mathematical form.

The merits and shortcomings of this methodology have been discussed in a recent review by the ASCE task committee on application of ANNs in hydrology (ASCE, 2000a,b). They have indicated that rainfall-runoff modelling has received maximum attention by ANN modellers. In a preliminary study, Halff *et al.* (1993) designed a three-layer feed-forward ANN using the rainfall hyetographs as input and hydrograph as output. This study opened up several possibilities for rainfall-runoff application using neural networks. The studies by Smith and Eli (1995) and

Received 16 October 2001
Accepted 19 December 2001

Shamseldin (1997) may be viewed as a ‘proof of concept’ analysis for ANNs in rainfall runoff modelling. Subsequently, a number of studies have been reported that employed neural networks for rainfall-runoff modelling (e.g. Hsu *et al.*, 1995; Tokar and Johnson, 1999; Abrahart and See, 2000). The rainfall-runoff process lends itself well to ANN applications. The nonlinear nature of the relationship, availability of long historical records, and the complexity of the physically based models in this regard are some of the factors that have forced researchers to consider alternative models; ANNs have been a logical choice.

However, one of the most unresolved questions in modelling of the rainfall-runoff process when applying ANNs is what architecture should be used to map the process effectively. The selection requires choosing an appropriate input vector, besides the hidden units and weights. Unlike the physically based models, the sets of variables that influence the system are not known *a priori*. Therefore, the selection of an appropriate input vector that will allow an ANN to map to the desired output vector successfully is not a trivial task. In most of the applications that are reported this has been done by a trial- and error-procedure (Fernando and Jayawardena, 1998). When developing models such as of the auto-regressive moving average (ARMA) or multiple linear regression (MLR) type, the order of the inputs can be determined using empirical and/or analytical approaches (Haltiner and Salas, 1988). Taking a statistical perspective is especially important for ‘atheoretical models’ like ANNs, because the reason for applying them is that they do not require knowledge about an adequate functional form. However, the analytical approaches are not used to determine the inputs for multivariate ANN models. The main reason for this is that ANNs belong to the class of data-driven approaches, whereas the conventional statistical methods are model driven (Chakraborty *et al.*, 1992). In the model-driven approaches, the structure of the model has to be determined first, which is done with the aid of the empirical or analytical approaches before the unknown model parameters can be determined. The data-driven approaches, on the other hand, have the ability to determine which model inputs are critical. However, presenting a large number of inputs to ANN models and relying on the network to determine the critical model inputs usually increases network size. This has a number of disadvantages, such as increasing training time, increasing the amount of data required for efficiently estimating the connection weights, and increasing the number of local minima in the error surface, which makes it more difficult to obtain a near-optimal combination of the weights for the problem under consideration.

Despite the huge amount of network theory and the importance of neural networks in applied work, there is still little experience with a statistical and/or analytical approach to model construction. In hydrological problems, since the number of input parameters involved in most real situations is quite large, implying unnecessarily large networks and large computation time, it is preferable to have a logical choice of the input vector prior to training of the network. Consequently, there are distinct advantages in using a systematic technique to help determine the inputs for multivariate ANN models. The model selection becomes more comprehensible, and reconstructible, when based on a clearly defined decision rule. Hydrologists need not train an arbitrarily large number of networks with different input vectors, then use some criterion based on information theory and predicted output to compare and select the best network.

The aim of this article is to outline a procedure for selecting an appropriate input vector in ANN rainfall-runoff models, based on statistical pre-processing of the data set. The proposed methodology has been illustrated by presenting an application of the procedure to an Indian river basin. The results reported by some researchers have also been analysed to check the effectiveness of the proposed algorithm. Focus will be on the input vector selection in this article.

Methodology

There are no fixed rules for developing an ANN, even though a general framework can be followed based on previous successful applications in engineering. The goal of an ANN is to generalize a relationship of the form of

$$\boldsymbol{Y}^m = f(\boldsymbol{X}^n) \qquad (1)$$

where $\boldsymbol{X}^n$ is an n-dimensional input vector consisting of variables $x_1, \ldots, x_i, \ldots, x_n$; $\boldsymbol{Y}_m$ is an m-dimensional output vector consisting of the resulting variables of interest $y_1, \ldots, y_i, \ldots, y_m$. In rainfall-runoff modelling, values of x_i may be rainfall/runoff values with different lags and the value of y_i is generally the next day's flow. However, how many antecedent rainfall/runoff values should be included in the vector X^n is not known *a priori*. A firm understanding of the hydrologic system under consideration would play an important role in successful implementation of ANNs. This would help in avoiding loss of information that may result if key input variables are omitted, and also prevent inclusion of spurious input variables that tend to confuse the training process.

The parameters that need to be selected in the input vector are the number of rainfall/runoff values for different intervals of time that can best represent the process by an ANN model. Determining the number of rainfall/runoff parameters involves finding the lags of rainfall/runoff that have a significant influence on the predicted flow. These influencing values corresponding to different lags can be very well established through statistical analysis of the data series. A qualitative examination of the cross-correlation curves between the rainfall and runoff series would reveal which antecedent rainfall heavily influences the runoff at a certain time. Similarly, an autocorrelation function (ACF) and partial autocorrelation function (PACF) would suggest the influencing antecedent discharge patterns in the flow at a given time. The ACF and PACF are generally used in diagnosing the order of the autoregressive process and can be employed in ANN modelling, too. The ACF and PACF with 95% confidence levels and the cross-correlation of the data series in question can be examined, and the modeller can decide on the number of antecedent rainfall/runoff values that should be included in the input vector. The variables that may not have a significant effect on the performance of the model can be trimmed off from the input vector, resulting in a more compact network.

Once the appropriate input vector to the network has been identified, a standard training algorithm can be employed for estimating the connection weights. In the present study, a standard radial basis function network (Moody and Darken, 1989) with a minimum description length (MDL) algorithm (Leonardis and Bischof, 1998) has been employed. The MDL algorithm is efficient in determining the significant basis functions and optimizing the connection weights of the network (Sudheer, 2000). The suggested procedure has also been checked by designing the ANN through a traditional method. In this analysis, many networks have been trained with various combinations of rainfall corresponding to different lags (varying from 1 to 10 days) and runoff lags (varying from 1 to 8 days). The root-mean-square error (RMSE) statistic of the predicted flow of the network has been analysed to quantify the influence of variation in the input vector.

Case Study

The methodology for selection of the input vector described earlier is illustrated with its application to the Baitarani River basin in the eastern part of India. The data series consists of 23 years (1972–94) of daily values of rainfall and runoff for the monsoon season (June to October). The basin has a drainage area of 8570 km^2.

Results and Discussion

The cross-correlation statistics of the rainfall-runoff series are presented in Figure 1. In the present application, the Pearson cross-correlation between the rainfall and runoff series (Figure 1) showed a significant correlation for up to a 7 days lag in rainfall data on the flow at any time. Further analysis (MLR) of the data suggested that rainfall intervals up to a 5 days lag are able to explain 89% of the total variance and no significant improvement is observed when the lag is increased to six or more.

The ACF and the corresponding 95% confidence bands from lag 0 to lag 16 (0 to 16 days) were estimated for the standardized flow series (Figure 2). The main reason for standardizing the data matrix

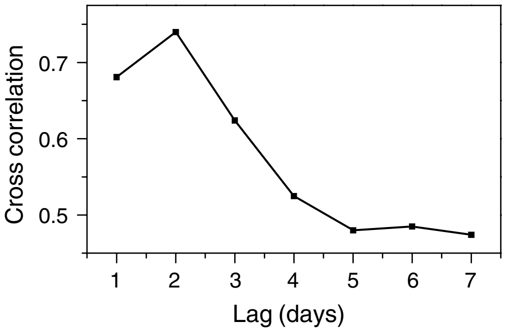

Figure 1. The cross-correlation plot of the rainfall-runoff series of Baitarani river basin

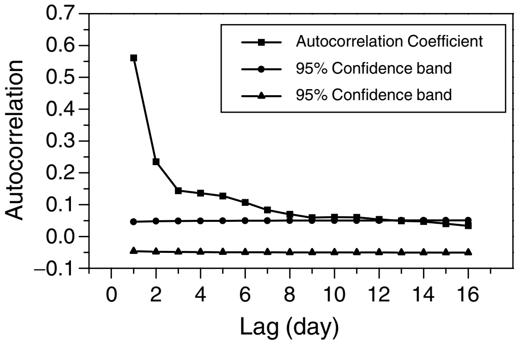

Figure 2. Autocorrelation plot of the flow series

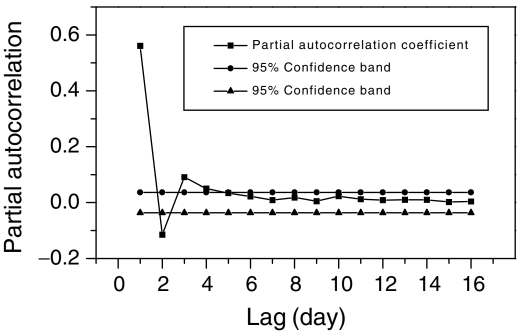

Figure 3. Partial autocorrelation plot of the flow series

is that the variables are usually measured in different units. By standardizing the variables and recasting them in dimensionless units, the arbitrary effects of similarity between objects are removed. The current study used the procedure described by Raman and Sunilkumar (1996). The ACF showed a significant correlation, at the 95% confidence level, up to lag 7 (7 days), and thereafter, fell within the confidence band. The gradual decaying pattern of the autocorrelation exhibits the presence of a dominant autoregressive process. Similarly, the PACF and corresponding 95% confidence limits were estimated for lag 0 to lag 16 (Figure 3). The PACF showed significant correlation up to lag 4 (4 days) and, thereafter, fell within the confidence band. The rapid decaying pattern of the PACF confirms the dominance of the autoregressive process, relative to the moving-average process. The above analysis of auto- and partial-correlation coefficients suggested incorporating flow values of up to 4 days lag in the input vector to the network.

The foregoing analysis shows that the input vector to the network can be defined with four antecedent runoff intervals and five antecedent rainfall intervals, thus making the input a nine-element vector. This qualitative analysis of the data series relieves the modeller of a long trial- and error-procedure in identifying the appropriate input vector that best represents the process in the basin.

The results pertaining to the validation of the proposed methodology are presented in Table I, which depicts the RMSE for predicted standardized flow series during training of the network with different rainfall–runoff lag combinations in the input vector. The RMSE statistics measure the residual variance; the optimal value is 0·0. The value of RMSE is found to vary considerably between different input vectors (from 0·005 83 to 0·046 20). However, it is worth noting that the errors are reasonably small in all cases. A significant observation is that the RMSE is minimum for the network with an input vector containing five antecedent rainfall and four antecedent runoff values, which corresponds to a flow of 25 cumec. This result is in direct agreement with the input vector selected based on the proposed methodology. It is easy to see that the aforementioned method involves a simple procedure that could assign just one single input vector

Table I. The RMSE statistics of validation of the algorithm

Antecedent runoff values in input vector Antecedent rainfall	1	2	3	4	5	6	7	8
1	0.008 16	0.009 23	0.024 10	0.018 70	0.026 50	0.008 12	0.009 23	0.008 54
3	0.009 46	0.007 82	0.006 87	0.006 76	0.046 20	0.025 60	0.007 12	0.007 60
5	0.009 25	0.006 32	0.016 98	0.005 83	0.012 40	0.009 32	0.009 25	0.008 12
7	0.008 16	0.009 23	0.024 10	0.018 70	0.026 50	0.008 12	0.009 23	0.008 54
10	0.009 46	0.007 82	0.006 87	0.005 76	0.046 20	0.025 60	0.007 12	0.007 60

to the ANN model, which would map the process satisfactorily.

The performance of the optimal neural network model developed using this procedure has been compared with that of corresponding ARMA and MLR models. The details on the development of these ARMA and MLR models for the Baitarani basin can be seen in Sudheer (2000). The values of some of the commonly used performance indices are presented in Table II, for comparison. It may be noted that the values presented in Table II correspond to the forecasted flow during validation, for each of the models.

It is evident from the Table II that the optimum ANN was able to forecast the river flow to a smaller RMSE compared with other models. The peak flow was overpredicted by ANN (+5·14%), whereas other models underpredicted the peak flow (−4·92% for ARMA and −41·23% for MLR). The results indicate that the performance of the ANN was superior to the other models.

As mentioned before, an examination has been made on the reported results of Fernando and Jayawardena (1998), Raman and Sunilkumar (1996), and Tokar and Johnson (1999) for decision on the input vector. All of them used a trail- and-error procedure for identifying the optimal input vector. Fernando and Jayawardena (1998) developed an ANN rainfall-runoff model using hourly data of rainfall and runoff for an experimental catchment in Kamhonsa in Japan. A qualitative examination of the cross-correlation curves between the rainfall and runoff presented by them reveals that the correlation is highest for a lag of 3 h. The serial correlation of the runoff indicates a decay with increasing lag, and was significant up to lag of 3 h. Consequently, according to the proposed algorithm, the optimal input vector to the network model would consist of rainfall values of up to three antecedent intervals and runoff values of up to three antecedent intervals. Obviously, the best network they reported also considers the same input vector.

Table II. Statistics of model predictions during validation

Performance index	Model		
	ANN	ARMA	MLR
Explained variance (%)	97·99	90·35	89·02
Coefficient of efficiency (%)	92·06	80·70	73·62
RMSE (cumec)	80·63	258·69	219·38
Correlation	0·96	0·90	0·64

Raman and Sunilkumar (1996), in their study on multivariate time series modelling using ANN for two basins, *viz.* Mangalam, Pothundi in India, presented the ACF and PACF for the data series. An examination of the ACF and PACF reveals that the flow at any time is highly related to two antecedent flows. They developed 12 ANN models using 12 different input vectors, and, based on an error analysis of the residuals, they reported that the ANN model with two antecedent flow values was the best among them. This is in direct agreement with the proposed algorithm presented in this work.

In a recent study on rainfall-runoff modelling using ANNs by Tokar and Johnson (1999), they developed nine ANN models for the Little Patuxent River Watershed in Howard County, Maryland, by using different input combinations. In the model development stage, they systematically increased the number of variables in the input vector (readers are referred to Tokar and Johnson (1999) for more details of the procedure) and uncertainty analysis was performed for all the models. They reported that the best ANN model out of nine consisted of two antecedent rainfall patterns. A qualitative examination of the cross-correlation matrix presented by them indicates that the flow at any time was significantly related to two antecedent rainfall values, hence validating the proposed methodology.

The above analysis reveals that the proposed algorithm would also result in the same network architecture that they have defined based on various criteria, thus eliminating a huge amount of computational time in training and developing an arbitrarily large number of ANN models and selecting the best one. The methodology presented, therefore is more comprehensive, and the decision is based on clearly defined rules and can be applied for any ANN rainfall-runoff model development. However, it is worth mentioning that the input vector selection procedure presented relies on the linear relationship between the variables and the effect of an additional variable to capture any non-linear residual dependencies is not assessed in this procedure. Still, the approximations that have been made based on this procedure can be justified, as it aids in defining an optimality criterion for choosing the input vector prior to training the network and eliminates relative comparisons of trained networks using a large number of possible input vectors. Conversely, there is clearly room for further research in this direction.

Summary and Conclusions

A new approach for designing an ANN rainfall-runoff model is presented. The method utilizes the statistical properties of the data series for identifying an appropriate input vector to the network, and trains the network with a standard algorithm. The overall results of the study are highly promising, as they lead to a significant reduction in computing time. From the aforementioned discussions, it can be concluded that the proposed algorithm would easily lead to a more compact network, thus avoiding a long trial- and-error procedure. The study suggests that the statistical selection of an input vector integrated with a standard optimization algorithm for network parameters during training may considerably reduce the computation time, and thus minimize the effort put into model development. A successful implementation of the methodology presented can lead to certain automation procedures in model development. Since the proposed methodology is based on the information contained in the data series itself, and is based on clear statistical properties as decision rules, the approach becomes more explicit and can be adopted for any basin. The specific advantages of the approach are: (1) it can determine which inputs are significant for the model in question; (2) it utilizes valuable information about the relationship between input and output time series; (3) it is simpler and quicker to use, since there is no need for large preprocessing of the data. It also relieves the modeller of a long trial- and-error procedure that requires some knowledge of the ANNs. The results of application of the proposed methodology validate the approach.

References

Abrahart RJ, See L. 2000. Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes* **14**: 2157–2172.

ASCE. 2000a. Artificial neural networks in hydrology–I: preliminary concepts. *Journal of Hydrologic Engineering, ASCE task committee on application of ANNS in hydrology* **5**(2): 115–123.

ASCE. 2000b. Artificial neural networks in hydrology–II: hydrologic applications. *Journal of Hydrologic Engineering, ASCE* **5**(2): 124–137.

Chakraborty K, Mehrotra K, Mohan CK, Ranka S. 1992. Forecasting the behavior of multivariate time series using neural networks. *Neural Networks* **5**: 961–970.

Fernando AK, Jayawardena AW. 1998. Runoff forecasting using RBF networks with OLS algorithm. *Journal of Hydrologic Engineering, ASCE* **3**(3): 203–209.

Halff AH, Halff HM, Azmoodeh M. 1993. Predicting Runoff from Rainfall using Neural Networks. In *Engineering Hydrology*, Kuo CY (ed.). Proceedings of the Symposium sponsored by the Hydraulics Division of ASCE, San Francisco, CA, July 25–30, 1993. ASCE: New York; 760–765.

Haltiner JP, Salas JD. 1988. Short-term forecasting of snowmelt runoff using ARMAX models. *Water Resources Bulletin* **24**(5): 1083–1089.

Hsu K, Gupta VH, Sorooshian S. 1995. Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research* **31**(10): 2517–2530.

Leonardis A, Bischof H. 1998. An efficient MDL-based construction of RBF networks. *Neural Networks* **11**: 963–973.

Moody J, Darken C. 1989. Fast learning in networks of locally tuned processing units. *Neural Computation* **1**: 281–294.

Raman H, Sunilkumar N. 1996. Multivariate modeling of water resources time series using artificial neural networks. *Hydrological Sciences Journal* **40**: 146–163.

Shamseldin AY. 1997. Application of neural network technique to rainfall runoff modeling. *Journal of Hydrology* **199**: 272–294.

Smith J, Eli RN. 1995. Neural network models of rainfall runoff process. *Journal of Water Resources Planning and Management, ASCE* **121**(6): 499–508.

Sudheer KP. 2000. *Modeling hydrological processes using neural computing technique*. PhD Thesis, Indian Institute of Technology, Delhi, India.

Tokar AZ, Johnson PA. 1999. Rainfall-runoff modeling using artificial neural network. *Journal of Hydrologic Engineering, ASCE* **4**(3): 232–239.