Artificial Neural Networks in Hydrology. II: Hydrologic Applications

By the ASCE Task Committee on Application of Artificial Neural Networks in Hydrology[1]

Abstract: This paper forms the second part of the series on application of artificial neural networks (ANNs) in hydrology. The role of ANNs in various branches of hydrology has been examined here. It is found that ANNs are robust tools for modeling many of the nonlinear hydrologic processes such as rainfall-runoff, stream flow, ground-water management, water quality simulation, and precipitation. After appropriate training, they are able to generate satisfactory results for many prediction problems in hydrology. A good physical understanding of the hydrologic process being modeled can help in selecting the input vector and designing a more efficient network. However, artificial neural networks tend to be very data intensive, and there appears to be no established methodology for design and successful implementation. For this emerging technique to find application in engineering practice, there are still some questions about this technique that must be further studied, and important aspects such as physical interpretation of ANN architecture, optimal training data set, adaptive learning, and extrapolation must be explored further. The merits and limitations of ANN applications have been discussed, and potential research avenues have been explored briefly.

INTRODUCTION

In Part I of this two-part series, the important structural and functional aspects of neural networks were presented. ANNs mimic the functioning of a human brain by acquiring knowledge through a learning process that involves finding an optimal set of weights for the connections and threshold values for the nodes. The differences between different types of networks were described briefly, and some details of the supervised learning algorithms were provided.

Mathematically, an ANN may be treated as a universal approximator. The ability to learn and generalize ''knowledge'' from sufficient data pairs makes it possible for ANNs to solve large-scale complex problems such as pattern recognition, nonlinear modeling, classification, association, control, and others —all of which find application in hydrology today. A significant growth in the interest of this computational mechanism has occurred since Rumelhart et al. (1986) developed a mathematically rigorous theoretical framework for neural networks. Since then, ANNs have found increasing use in diverse disciplines ranging over perhaps all branches of engineering and science. Researchers in hydrology have shown serious interest in this computational tool only during the last decade.

Hydrology is the scientific study of water and its properties, distribution, and effects on the earth's surface, soil, and atmosphere (McCuen 1997). Hydrologists are often confronted with problems of prediction and estimation of runoff, precipitation, contaminant concentrations, water stages, and so on. Most hydrologic processes exhibit a high degree of temporal and spatial variability and are further plagued by issues of nonlinearity of physical processes, conflicting spatial and temporal scales, and uncertainty in parameter estimates. Our understanding in many areas is far from perfect, so that empiricism plays an important role in modeling studies. Hydrologists attempt to provide rational answers to problems that arise in design and management of water resources. An attractive feature of ANNs is their ability to extract the relation between the inputs and outputs of a process, without the physics being explicitly provided to them. They are able to provide a mapping from one multivariate space to another, given a set of data representing that mapping. Even if the data is noisy and contaminated with errors, ANNs have been known to identify the underlying rule. These properties suggest that ANNs may be well-suited to the problems of estimation and prediction in hydrology.

The goals of this paper are to examine how successful ANNs have been in hydrologic problems and to evaluate if indeed all the strengths of ANNs have been effectively utilized in these applications. The role of ANNs in specific areas is considered individually for organizational purposes. This is followed by a short critique. The separation of articles into various hydrologic applications is not always very clear. Many papers could be easily classified into multiple categories. Thus, our classification here is for organizational convenience only and is not meant to categorize a paper into a specific application. The focus of the task committee has been on targeting this work for hydrologists, particularly those who practice hydrology in the field. While we have tried to cover the popular journals that are usually read by hydrologists, there are other areas (particularly agricultural engineering, chemical engineering, soil science, atmospheric science, etc.) where journals report articles of common interest. Articles from journals that are typically outside the purview of hydrologists have not been included and were considered as being outside the scope of the task committee activities. Daniel (1991) reported some of earliest applications of ANN in hydrology and water resources engineering. General applications of ANNs have been discussed briefly by Taylor (1996). Flood and Kartam (1994, 1997) reviewed the application of artificial neural networks to various branches of civil engineering. However, this study did not present any details on hydrologic applications. Therefore, this paper complements these earlier studies.

APPLICATIONS IN RAINFALL-RUNOFF MODELING

Determining the relationship between rainfall and runoff for a watershed is one of the most important problems faced by hydrologists and engineers. Information about rainfall and runoff is needed for hydrologic engineering design and management purposes. This relationship is known to be highly nonlinear and complex. In addition to rainfall, runoff is dependent on numerous factors such as initial soil moisture, land use, watershed geomorphology, evaporation, infiltration, distribution, duration of the rainfall, and so on. Although many watersheds have been gauged to provide continuous records of stream flow, engineers are often faced with situations where

[1]Rao S. Govindaraju, Assoc. Prof., Purdue Univ., School of Civ. Engrg., 1284 Civ. Engrg. Build., West Lafayette, IN 47907-1284.

Note. Discussion open until September 1, 2000. Separate discussions should be submitted for the individual papers in this symposium. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on August 20, 1998. This paper is part of the ***Journal of Hydrologic Engineering***, Vol. 5, No. 2, April, 2000. ©ASCE, ISSN 1084-0699/00/0002-0124–0137/$8.00 + $.50 per page. Paper No. 19091.

little or no information is available. In such instances, simulation models are often used to generate synthetic flows.

A number of researchers have investigated the potential of neural networks in modeling watershed runoff based on rainfall inputs. In a preliminary study, Halff et al. (1993) designed a three-layer feedforward ANN using the observed rainfall hyetographs as inputs and hydrographs recorded by the U.S. Geological Survey (USGS) at Bellvue, Washington, as outputs. The authors decided to use five nodes in the hidden layer. A total of five storm events were considered. On a rotation basis, data from four storms were used for training, while data from the fifth storm were used for testing network performance. A sequence of 25 normalized 5 min rainfalls was applied as inputs to predict the runoff. This study opened up several possibilities for rainfall-runoff application using neural networks.

Hjelmfelt and Wang (1993a–c) developed a neural network based on the unit hydrograph theory. Using linear superposition, a composite runoff hydrograph for a watershed was developed by appropriate summation of unit hydrograph ordinates and runoff excesses. To implement this in a neural network framework, the number of units in the input and hidden layer was kept the same. Connections only existed between the corresponding pairs in the first two layers, i.e., the *i*th node in the first layer connects only to the *i*th node in the second layer, with the weights being set to unity. The nodes in the hidden layer were fully connected with the single output node representing runoff. The inputs to the ANN were sequences of rainfall. Instead of the threshold function, a ramp transfer function corresponding to the rainfall ϕ-index was used for the hidden layer. The hidden layer served to extract the infiltration from rainfall, and its outputs were rainfall excesses. The output layer calculated a weighted sum of the rainfall excesses. Rainfall and runoff data from 24 large storm events were chosen from the Goodwater Creek watershed (12.2 km^2) in central Missouri to train and test the ANN. The resulting network was shown to reproduce the unit hydrograph better than the one obtained through the standard gamma function representation. In a later study, Hjelmfelt and Wang (1996) compared this method with a regular three layered artificial network with back-propagation. The authors concluded that a regular network could not reproduce the unit hydrograph very well and was more susceptible to noise than a network whose architecture was more suited for unit hydrograph computations.

In an application using two neural networks, Zhu et al. (1994) predicted upper and lower bounds on the flood hydrograph in Butter Creek, New York. Off-line predictions were made when present flood data were not available and estimates had to be based on rainfall data alone. On-line predictions were based on both rainfall and previous flood data. Data for ANN testing and validation were generated from a nonlinear storage model. Model performance was strongly influenced by the training data set. The authors found that, while the ANN did well during interpolation, predictions made by ANNs outside the range of the training data set were not encouraging. The process of trying to make ANNs adaptive was computationally very demanding, because the entire training process needed to be repeated with each new data pair. As the lead time for forecasting increased, ANN performance deteriorated. By comparison, ANNs were found to be marginally better than fuzzy inference-based techniques.

Smith and Eli (1995) applied a back-propagation neural network model to predict peak discharge and time to peak over a hypothetical watershed. Data sets for training and validation were generated by either a linear or a nonlinear reservoir model. By representing the watershed as a grid of cells, it was possible for the authors to incorporate the spatial and temporal distribution information of rainfall into the ANN model. As an example, the authors chose a synthetic watershed that was composed of 5×5 cells. A tree-type drainage pattern was superimposed on the grid to concentrate runoff towards a single watershed outlet. Each cell was treated as a reservoir and water was routed in a cascading fashion. A rainfall depth of one unit was applied instantaneously at several cells on a random basis. Each rainfall pattern in the training set was presented to the network as an input image consisting of Boolean values with 1 representing a wet cell and 0 a dry cell. The peak discharge and the time to peak corresponding to each rainfall pattern were computed using a linear and nonlinear reservoir model and served as target outputs for the ANN model. Many such patterns formed the training set. These cases represented single-storm events for which the number of input units was the same as the number of cells. To simulate the occurrence of several storms in a sequence, three stochastically generated rainfall patterns were imposed consecutively over the synthetic watershed. In this case, the input layer had 75 units, corresponding to three rainfall patterns requiring 25 cells each. The output was either the watershed runoff alone or the runoff and the time to peak. The number of nodes in the hidden layer was determined by trial and error for each case. For single-storm events, the peak discharge and the time to peak were predicted well by the neural network, both during training and testing. The authors were less successful for multiple-storm events. One reason for this may have been insufficient number of nodes in the output layer. In a separate application dealing with multiple storms, Smith and Eli (1995) represented the entire hydrograph by a Fourier series with 21 coefficients, rather than just two attributes as in single-storm events. The ANN output layer now consisted of 21 nodes corresponding to the Fourier coefficients. Using this method, the authors found the prediction of the entire hydrograph to be very accurate for multiple storm events.

The issue of enhancing the training speed using a three-layer network was addressed by Hsu et al. (1995) and Gupta et al. (1997). These studies advocated the linear least squares simplex (LLSSIM) algorithm, which partitions the weight space to implement a synthesis of two training strategies. The input-hidden layer weights were estimated using a multistart downhill simplex nonlinear optimization algorithm, while the hidden-output layer weights were estimated using optimal linear least square estimation. The nonlinear portion of the search was thereby confined to a smaller dimension space, resulting in acceleration of the training process. The simplex search involves multiple starts that are initiated randomly in the search space, and the probability of finding local minima is virtually eliminated. The authors applied this technique to daily rainfall-runoff modeling of the Leaf River Beam near Collins, Mississippi. The performance of neural networks was compared with the linear ARMAX time series model and the conceptual SAC-SMA model. Figs. 1 and 2 show the performance of these models during calibration and validation, representatively, in Hsu et al. (1995). Even though all the models seemed to underestimate low flows in general, the ANN performance was found to be superior to the other models. Gupta et al. (1997) concluded that the LLSSIM is likely to be a better training algorithm than back-propagation or conjugate gradient techniques, especially in the absence of a good initial guess of weights. In another related study over the Leaf River Basin, Hsu et al. (1997) used a three-layer feedforward ANN and a recurrent ANN to model daily rainfall-runoff. They concluded that the feedforward ANN needed a trial-and-error procedure to find the appropriate number of time-delayed input variables to the model and also was not suitable to distributed watershed modeling. On the other hand, the recurrent ANN was able to provide a representation of the dynamic internal feedback loops in the system, eliminating the need for lagged inputs and

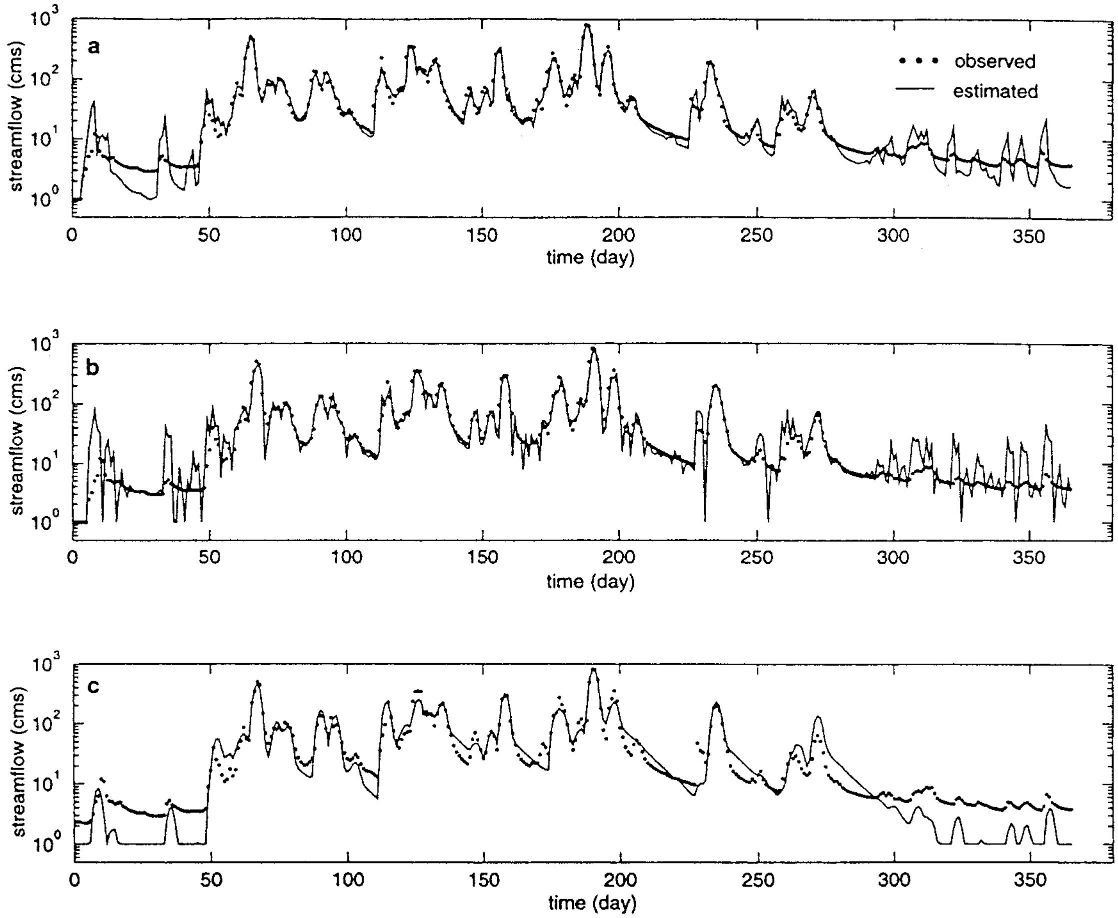

FIG. 1. Best Fit Hydrographs for Calibration Data Year 1982: (a) ANN (5, 4, 3, 2); (b) ARMAX (2, 4, 3); (c) SAC-SMA (Hsu et al. 1995)

resulting in a compact weight space. However, both ANNs performed equally well at runoff prediction.

Carriere (1996) developed a virtual runoff hydrograph system that employed a recurrent back-propagation artificial neural network to generate runoff hydrographs. A recurrent back-propagation network was utilized, in which input layer feeds back to itself during training to capture time dependence in the series. The network consisted of 7 input nodes, 35 nodes in hidden layer, and a single node in the output layer. Bipolar linear normalization was used in the input layer, and the logistic function was used for activation in the nodes of the hidden and output layer. Data from 45 laboratory experiments over a small watershed under different conditions of slope and cover were selected to develop the neural network. Out of these, 29 data sets were employed to train the neural network, and the rest were used for testing. The author concluded that the neural network could predict runoff hydrographs accurately, with good agreement between the observed and predicted values.

In a study by Minns and Hall (1996), data for network training consisted of model results from one storm sequence, and two such sequences were generated for testing. Each storm sequence was generated using a Monte Carlo procedure that preserved predetermined storm characteristics. For each such storm sequence, the corresponding runoff sequence was constructed using a simple nonlinear model for flood estimation (called RORB) that allowed for different levels of nonlinearly in the response. A three-layer network with back-propagation was used. Network inputs consisted of concurrent and 14 antecedent rainfall depths and 3 antecedent runoff values, and the network output was current runoff. It was found that ANN performance was hardly influenced by level of nonlinearity, with performance deteriorating only slightly for high levels of nonlinearity. This could be rectified by using 2 hidden layers and the associated extra cost on network training. Minns and Hall (1996) point out the importance of standardization based on maximum and minimum values of inputs and outputs. Whenever the network was required to predict ''out of range'' of the standardized values, the performance dropped significantly, suggesting that ANNs are not very good extrapolators.

Haykin (1994) showed that design of a supervised neural network might be pursued in a number of different ways. While the back-propagation algorithm for the design of a multilayer perceptron (under supervision) may be viewed as an application of stochastic approximation, radial-basis function (RBF) networks can be viewed as a curve-fitting problem in a high-dimensional space. Therefore, the learning for such networks is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data, with the criterion for ''best fit'' being expressed in a statistical sense. Mason et al. (1996) used RBF networks for accelerating the training procedure as compared with regular back-propagation techniques. Data were generated using the Simulation Program for Interactive Drainage Analysis (SPIDA) model. The network output was runoff based on inputs consisting of time, rainfall intensity, cumulative rainfall, and derivative of rainfall intensity. The authors briefly discuss network architectures and compositions and tried five different forms of basis functions in their study. Sixty data sets were utilized for network training, and 39 were used for validation of the model. The authors concluded that, while RBF networks did provide for faster training, such networks require the solution of a linear system of equations that may become ill conditioned, especially if a large number of cluster centers are chosen.

Jayawardena and Fernando (1995, 1996) and Fernando and Jayawardena (1998) also used RBF methods for flood fore-

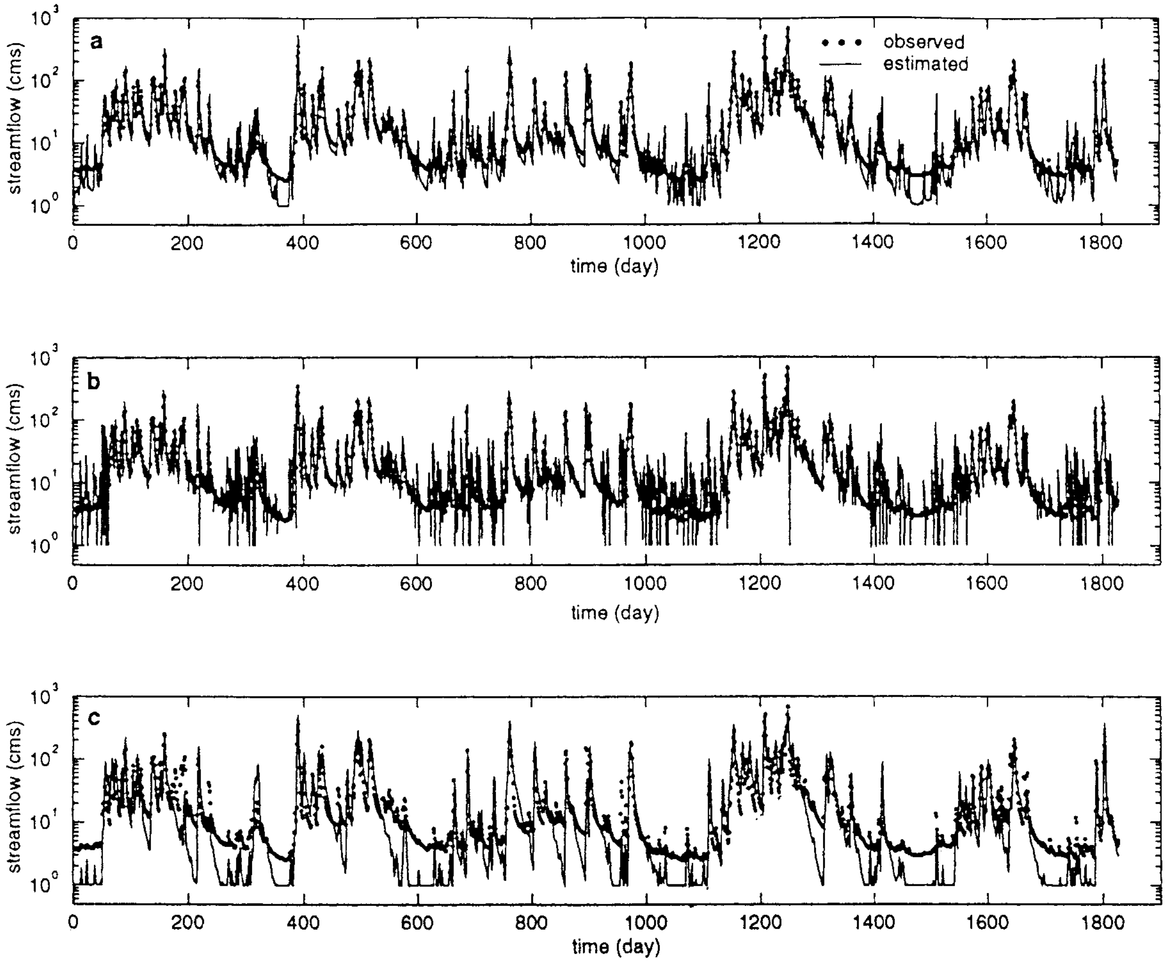

FIG. 2. One-Step-Ahead Prediction Hydrographs for Five-Year Validation Period: (a) ANN (5, 4, 3, 1); (b) ARMAX (2, 4, 3); (c) SAC-SMA (Hsu et al. 1995)

casting. They illustrated the application of (RBF) artificial neural networks using an orthogonal least squares algorithm (OLS) to model the rainfall-runoff process. Hourly rainfall and runoff data from a 3.12 km^2 watershed were collected and used in developing the ANN. The autocorrelation of runoff, and the cross correlation between rainfall and runoff indicated that the discharges at a certain time were influenced by antecedent rainfall from up to three previous hours. Therefore, the input nodes contained three antecedent discharges and two rainfall values—that is, $Q(t\text{-}1)$, $Q(t\text{-}2)$, $Q(t\text{-}3)$, $R(t\text{-}2)$, and $R(t\text{-}3)$. The output was the discharge at the current hour, $Q(t)$. Both a multiple layer perceptron (MLP) neural network and a RBF network were developed and compared with the statistical ARMAX model. Even though both the RBF and MLP networks performed well, it was found that RBF networks could be trained much faster than MLP networks using back propagation. Both networks performed better than the ARMAX model.

Shamseldin (1997) compared ANNs with a simple linear model, a season-based linear perturbation model, and a nearest neighbor linear perturbation model. Daily average values of rainfall and runoff from six different watersheds around the world were collected for this study. Three different types of input information were compiled from this data. These were weighted averages of recent rainfall measurements, seasonal information on ϕ-index and average discharges, and nearest neighbor information. Four different scenarios based on combinations of some or all of these types of input information were examined. A three-layer neural network was adopted by the author, and the conjugate gradient method was used for training. A two-parameter gamma function representation was chosen as the impulse response of the rainfall series. The parameters were also estimated as part of the training procedure. The network output consisted of the runoff time series. The results suggested that the neural networks generally performed better than the other models during training and testing.

In an effort to relate runoff to precipitation, snow and temperature, and previous streamflows, Tokar and Markus (1997) employed ANNs to predict monthly flows on the Fraser River near Granby, Colordo, and daily flows on the Raccoon Creek near Bayard, Iowa. The WATBAL and the SAC-SMA models were used as alternative tools for comparison purposes over the two watersheds, respectively. They created a three-layer ANN to predict monthly and daily runoff from two small basins. For the Fraser River, the ANN produced better results than the WATBAL model. In the case of Raccoon Creek, the best neural network was chosen among four alternatives and produced comparable results to the conceptual SAC-SMA model.

Tokar and Johnson (1999) reported that ANN models provided higher training and testing accuracy when compared with regression and simple conceptual models. Their goal was to forecast daily runoff for the Little Patuxent River, Maryland, with daily precipitation, temperature, and snowmelt equivalent serving as inputs. It was found that the selection of training data has a large impact on accuracy of prediction. The authors trained and tested the ANN with wet, dry, and average-year data, respectively, as well as combinations of these, in order to illustrate the impact of the training series on network performance. The ANN that was trained on wet and dry data had the highest prediction accuracy. The length of training record had a much smaller impact on network performance than the types of training data.

Dawson and Wilby (1998) used a three-layer back-propagation network to determine runoff over the catchments of the Rivers Amber and Mole. The two catchments are about 140 km^2 in size, and are prone to floods. ANN inputs were past flows and averages of past rainfall and flow values. The ANN output consisted of predicting future flows at 15 min intervals up to a lead time of six hours. Their results show that ANNs performed about as well as an existing forecasting system that required more information. When compared with actual flows, the ANNs appeared to overestimate low flows for the Mole River. Bonafe et al. (1994) assessed the performance of a neural network in forecasting daily mean flow from the upper Tiber River basin in central Italy. The previous discharge, daily precipitation, daily mean temperature, total rainfall of the previous five days, and mean temperature over the previous ten days were selected as ANN inputs. They concluded that the ANN was able to yield much better performances than ARMA models.

The problem of rainfall-runoff modeling has perhaps received the maximum attention by ANN modelers. This problem lends itself admirably to ANN applications (Hsu et al. 1995). The nonlinear nature of the relationship, availability of long historical records, and the complexity of physically-based models in this regard, are some of the factors that have caused researchers to look at alternative models—and ANNs have been a logical choice. Research activities in this aspect have been quite revealing, and they can be broadly classified into two categories.

The first category of studies are those where ANNs were trained and tested using existing models (e.g., Smith and Eli 1995; Shamseldin 1997). These studies may be viewed as providing a ''proof of concept'' analysis for ANNs. They have laid the foundations for future ANN use by demonstrating they are indeed capable of replicating model behavior, provided sufficient data is available for training. This requirement is easily met, as the data necessary for training can be generated on a computer relatively easily. In such cases, ANN performance can at best equal the original model that provided the data for training. Such experiments are a first step in evaluating the applicability of ANNs for use in real catchments.

One may argue that, after the training process is complete, ANNs would provide much faster responses than the original model—especially so if the model was a complex one. Given the increased computing capacity that is available nowadays, such an advantage is unlikely to be meaningful for deterministic predictions based on a single realization. The speed of ANN-based predictions could prove to be useful when dealing with Monte Carlo—based studies, where a very large number of realizations are required to obtain representative sample statistics. One could envision applications where Monte Carlo simulations are started using an existing model. Simultaneously, an ANN could be trained on these realizations. During the course of generating realizations for the Monte Carlo procedure, one would switch from the model to ANN predictions after it had acquired the necessary training. While this is a good idea in principal, some other issues have to be resolved before this can be implemented in an efficient manner. These issues are discussed in a later section of this paper.

Most ANN-based studies fall into the second category, those that have used observed rainfall-runoff data. Frequently, supplementary inputs such as temperature, snowmelt equivalent, and historical stream flows have been included. In such instances, comparisons with other empirical or conceptual models have also been provided. These studies provide a more comprehensive evaluation of ANN performance and are capable of establishing ANNs as viable tools for modeling rainfall-runoff. While most studies report that ANNs have resulted in superior performance, they have not been useful for providing any useful insight or furthering our understanding of watershed processes. Using ANNs as a mere black-box to reproduce an input-output sequence well does not help in advancing the scientific understanding of hydrological processes. More creative use of ANNs is modeling the rainfall-runoff process will be needed in the future. Some of these issues are discussed towards the end of this paper.

MODELING STREAMFLOWS

Streamflows are often treated as estimates of runoff from watershed and could be considered as part of the previous section. The focus here is on papers that have directly dealt with streamflow itself, usually without involving precipitation as input. In some studies, streamflow prediction was an intermediate goal. In one of the earlier applications involving streamflows, Kang et al. (193) used ANNs and autoregressive moving average models to predict daily and hourly streamflows in the Pyung Chang River basin in Korea. Different three-layered ANN architectures were investigated. This preliminary study concluded that ANNs are useful tools for forecasting streamflows.

In a more detailed study along similar lines, Karunanithi et al. (1994) were interested in estimating streamflows at an ungauged site on the Huron River in Michigan, based on data from USGS stream gauging stations located 30 km upstream and 20 km downstream of the sampling site. They compared ANN performance to an empirical two-station power law relationship that is based on log-transformation of the actual streamflow values. Fig. 3 shows a comparison of observed versus predicted daily flows for a testing period of two years. Daily data were found to exhibit rapid fluctuations, and the authors worked with five-day non-overlapping averages and five-day moving averages to obtain a smoother representation when using regression. However, the raw data were utilized as ANN inputs. They used the cascade-correlation algorithm (see Part I) so that the network architecture could be determined during training. When using empirical regression equations, the largest errors were associated with the highest stream flows. Neural networks were found to better predict these high events, while both methods predicted low streamflows fairly well. These authors stated that ANNs are capable of adapting their complexity to accommodate temporal changes in historical streamflow records. They also found that including another gauging station that supposedly had little or no effect on streamflows at the gauging site caused the performance of the

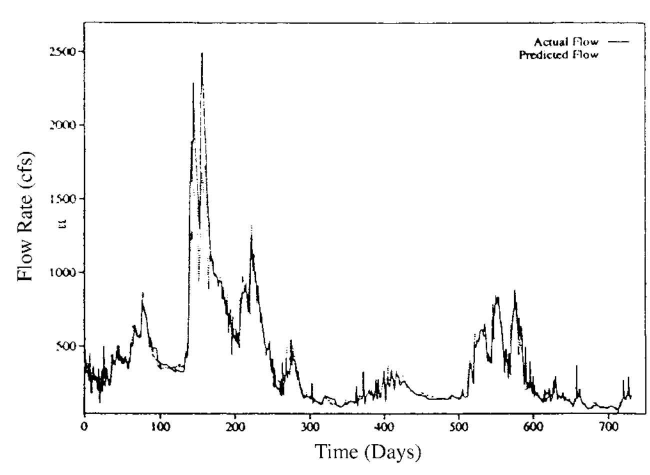

FIG. 3. Observed versus Predicted Flows at Dexter by Power Model for Testing Period 1976 and 1977 (Karunanithiu et al. 1992)

regression technique to deteriorate, while the ANN performance was not affected. The authors claimed that ANNs are likely to be more robust when noisy data is present in the inputs. Karunanithi et al. (1994) found lag time to be important in predicting streamflows. This reflects the longer memory associated with streamflows. The authors did not use any statistical techniques to evaluate the lag time and include it in the network architecture.

Markus et al. (1995) used ANNs with the back-propagation algorithm to predict monthly streamflows at the Del Norte gauging station in the Rio Grande Basin in Southern Colorado. The inputs used were snow water equivalent alone, or snow water equivalent and temperature. They used periodic transfer functions (PTFs) to predict streamflows based on similar inputs as an alternative form of prediction. For training, monthly data from 1948–1977 were used, and they tested model performance on monthly data from 1978–1987. They looked at forecast bias and root mean square error for assessing model performance. The results indicated that both ANNs and PTFs did a good job of predicting streamflows, and that including temperature as input improved model performance.

In an effort to evaluate ecological implications in terms of hydrologic variables. Poff et al. (1996) used ANNs to evaluate the changes in stream hydrograph from hypothetical climate change scenarios based on precipitation and temperature changes. The synthetic daily hydrograph was generated based on historic precipitation and temperature as inputs. They studied two streams in the northern United States under different hydro-climatological factors. The streamflow in the Little Paxutent River (near Baltimore, Maryland) is dominated by rainfall, while the Independence River in New York is influenced by both snow and rain. Three classes of hydrological variables of interest were derived from ANN-generated streamflow output. Mean flow conditions were composed of mean daily discharge, coefficient of variation of daily flow, and predictability of daily flow. High flow conditions included flood frequency, flood predictability, and flood-free period. The ANNs were particularly geared towards modeling these kinds of hydrologic variables. Finally, the low flow conditions were represented by the baseflow index. Four climate change scenarios were implemented by increasing the precipitation by 25%, decreasing the precipitation by 25%, doubling the coefficient of variation of daily precipitation, and increasing the average temperature by 3°C. The ecological implications of these changes for the two streams were discussed in terms of the hydrological variables.

Muttiah et al. (1997) also used the cascade-correlation algorithm in their efforts to predict two-year peak discharge from watersheds all over the continental United States. An interesting goal of this work was to investigate the possibility of a single model that could predict peak discharges from local to regional-sized watersheds. They wanted to use data that was easily available from GIS databases. Therefore, network inputs consisted of the log of the drainage basin area, elevation, average slope, and average annual precipitation. The authors claim that ANNs showed some improvement over the standard regression techniques employed by the USGS. Using input vector reduction techniques based on the cascade-correlation method, the authors concluded that drainage area and basin elevations could be used for predicting two-year peak discharges.

Stream rating curves often exhibit hysteresis, with the stage-discharge relationship being different for rising and receding stages. A single relationship is inadequate, while using two separate relationships leads to problems of separation. Tawfik et al. (1997) used ANNs, with a saturating linear transfer function, to predict flow discharges at two locations over the Nile River using the stage, H, and the rate of change of stage, dH/dt, as network inputs. ANNs were shown to predict discharge without exhibiting the separation problem associated with a method that uses different regression relationships for the rising and receding portions based on when dH/dt changes sign.

These studies indicate that ANNs have achieved some success in streamflow prediction, particularly when these are desired over a certain range of streamflow values. They have been used for obtaining quick and reliable forecasts. It has been shown to be superior to regression techniques and time-series models. Many of the comments presented about rainfall-runoff modeling are applicable to the problem of streamflow prediction as well. A major limitation appears to be in trying to design robust prediction techniques over a wide range of streamflows. The studies of Karunanithi et al. (1994) and Thirumalaiah and Deo (1998) directed network training to better replicate low streamflow events, while Poff et al. (1996) concentrated on high flow events to generate improved statistics for floods. It is generally believed that the hydrology of high and low flow events is different from events that are perceived as being normal. Future efforts should be directed towards designing ANNs to account for these different scenarios, in order to represent both normal and extreme conditions.

ANNs IN WATER QUALITY MODELING

In recent years, ANNs have found a number of applications in the area of water quality modeling. Water quality is influenced by many factors such as flow rate, contaminant load, medium of transport, water levels, initial conditions and other site-specific parameters. The estimation of such variables is often a complex and nonlinear problem, making it suitable for ANN application.

Maier and Dandy (1996) illustrated the utility of ANNs for estimating salinity at the Murray Bridge on the River Murray in South Australia. This river serves as a vital source for irrigation and water supply. The high level of salinity in the water had adverse effects on domestic, industrial, and agriculture uses. The authors felt that effective pumping strategies could be implemented if salinity values could be estimated 14 days in advance. In this study, the inputs to the ANN model were daily salinity values, and water levels and flows at upstream stations and at antecedent times. This resulted in 141 nodes in the input layer at the initial stage. Network output was the 14-day-advance forecast of river salinity. The authors chose two hidden layers and used back-propagation for training. The optimal ratio of number of nodes in the second and third layers was found to be 3:1. A sensitivity analysis was performed to screen the unnecessary inputs, and the number of input nodes was eventually reduced to 51. The ANN structure became 51-45-15-1 (51 inputs, 45 neurons in the first hidden layer, 15 neurons in second hidden layer, and a single output). Cross validation was used to overcome the problem of overtraining. Fig. 4 shows that the ANN model was able to

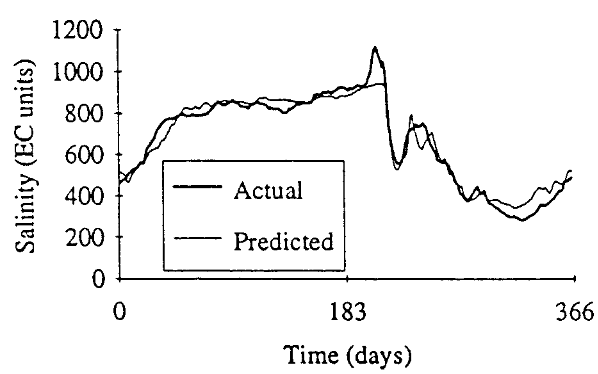

FIG. 4. Best 14-Day Forecast of Salinity at Murray Bridge for 1991 (Model 4-91) (Maier and Dandy 1996)

replicate salinity levels fairly accurately based on 14 day forecasts (Maier and Dandy 1986). It was observed that the average percentage errors of the independent 14 day forecasts for four different years of data varied from 5.3 to 7.0%. The authors also concluded that the impact of using different learning rates and different network geometries was relatively minor.

Rogers (1992) and Rogers and Dowla (1994) employed an ANN, which was trained by a solute transport model, to perform optimization studies in ground-water remediation. They investigated hypothetical scenarios of one or several contaminant plumes moving through a ground-water region with a number of pumping wells. The wells could be on or off. The goal of remediation was to keep contamination concentration in some specified monitoring wells lower than the regulatory limit. The optimization arises in trying to minimize the total volume of pumping. A multilayer feedforward ANN was trained using the back-propagation training algorithm. The input represented possible pumping cases, with wells being assigned a value of 1, indicating that the well was pumping at the maximum capacity, or 0, indicating that the well was off. The ratio of the number of nonpumping wells to the total number of wells was included as an additional input. The ANN output represented whether or not the realization of pumping met the regulatory consistent (successful), with the value being either 1 if successful, or 0 if not. The authors supplemented the conjugate gradient method with some weight elimination techniques to accelerate convergence and improve performance. ANN architectures with different numbers of hidden layers and nodes were investigated. Following the completion of training, the neural network was guided by a genetic algorithm for searching through various realizations of pumping patterns to determine whether they would be successful. Table 1 presents the accuracy and generalization performances for seven ANN training rounds with different architectures in Rogers and Dowla (1994). Results obtained by this method were consistent with those resulting from a conventional optimization technique using the solute transport model and nonlinear programming using a quasi-Newton search. This methodology was applied to a Superfund site by Rogers et al. (1993) and Johnson and Rogers (1995). They concluded that ANNs, combined with a genetic algorithm, result in robust and flexible tools that can be used for planning effective strategies in ground-water remediation. Morshed and Kaluarachchi (1998) used an ANN to estimate the saturated hydraulic conductivity and the grain size distribution parameter for application in the problem of free product recovery. They also concluded that the search process in the parameter space could be accelerated when the ANN was guided by a genetic algorithm.

Basheer and Najjar (1995) used a three-layer artificial neural network to predict the breakthrough time in a fixed-bed adsorption system. The data for training and validating the network were generated using the HSDM model. Using a systematic analysis, the authors identified three inputs as being the most influential in determining the breakthrough time. These were influent concentration, specific weight of the adsorbent, and the particle diameter of the porous bed material. A trial-and-error procedure led them to select 10 nodes in the hidden layer. The authors found ANN predictions to be reliable as long as the inputs were within the range of the data sets.

There are several other instances where ANNs have been used to address water quality related issues. For instance, Starrett et al. (1996) employed an ANN to predict pesticide leaching through turfgrass-covered soil. After an extensive selection procedure, the variables chosen as ANN inputs were pesticide solubility, rate of pesticide application, time since pesticide application, and type of irrigation practice being implemented. The ANN output was the percentage of pesticide that leached

TABLE 1. Accuracy and Generalization Performances for Seven ANN Training Rounds (Rogers and Dowla 1994)

Architecture (1)	Sets in input series (number of patterns) (2)	Accuracy (3)	Generalization (4)	Numbers of weights (5)	Number of function evaluations[a] (6)
	(*a*) Training round 1				
21-4-1	1,2 (100)	100%	82%	93	119
21-4-1	1,3 (100)	100%	82%	93	126
21-4-1	1,4 (100)	100%	85%	93	118
21-4-1	2,3 (100)	100%	91%	93	135
21-4-1	2,4 (100)	100%	91%	93	228
21-4-1	3,4 (100)	100%	69%	93	158
	(*b*) Training round 2				
21-4-1	1,2,3 (150)	100%	52%	93	—
21-4-1	2,3,4 (150)	92.7%	92%	93	4,560[b]
21-4-1	1,3,4 (150)	100%	72%	93	172
21-4-1	1,2,4 (150)	100%	86%	93	1,295
	(*c*) Training round 3				
21-5-1	1,2,4 (150)	100%	94%	116	301
21-6-1	1,2,4 (150)	100%	100%	139	3,571
21-2-2-1	1,2,4 (150)	98%	72%	53	5,000[b]
21-4-1	1,2,4 (150)	100%	86%	93	1,295
	(*d*) Training round 4				
21-6-1	1,2,3 (150)	100%	72%	139	68
21-6-1	2,3,4 (150)	100%	86%	139	477
21-6-1	1,3,4 (150)	100%	82%	139	205
21-6-1	1,2,4 (150)	100%	100%	139	3,571
	(*e*) Training round 5				
21-7-1	1,2,3 (150)	100%	62%	162	148
21-7-1	2,3,4 (150)	100%	92%	162	353
21-7-1	1,3,4 (150)	100%	80%	162	257
21-7-1	1,2,4 (150)	99.3%	98%	162	1,727[b]
	(*f*) Training round 6				
21-7-1	1,2,3,5 (195)	99.5%	80%	162	1,583[b]
21-7-1	2,3,4,5 (195)	100%	90%	162	1,300
21-7-1	1,3,4,5 (195)	100%	82%	162	238
21-7-1	1,2,4,5 (195)	99%	96%	162	2,883[b]
	(*g*) Training round 7 (weight elimination)				
21-7-1	1,2,3,5 (195)	87.7%	94%	162	1,265[b]
21-7-1	2,3,4,5 (195)	85.1%	98%	162	—
21-7-1	1,3,4,5 (195)	85.6%	96%	162	—
21-7-1	1,2,4,5 (195)	89.7%	92%	162	1,189[b]

[a]Number of evaluations required for error function to reach final value of $<10^{-5}$.
[b]Failed to reach successful optimization.

through 50 cm of turfgrass-covered soil. There were 3 nodes in the hidden layer. Out of a total of 200 data sets, 175 patterns were used to train the ANN, and 25 were used to test it. In another pesticide-related application, Ray and Klindworth (1996) lay a blueprint for addressing the problem of agriculture chemical assessment in the rural private wells in Illinois using neural networks. They envisioned that important inputs would be depth to the aquifer material, well depth, land topography in the vicinity of the well, distance of potential contaminant sources from the well, and timing of precipitation with respect to pesticide application. They also discussed how data would be collected for such an application and commented about the utility of ANNs in such applications. Sandhu and Finch (1996) used ANNs to relate flow conditions and gate positions in the Sacramento San Joaquin Delta to salinity levels in the interior and along the boundary of the delta. ANNs were further used to estimate flow in the Sacramento River to meet salinity standards. Sandhu and Finch (1996) found simulation models too slow and the commonly used statistical models to be inadequate, and they concluded that neural networks would be suitable for this application. Historical flows from various gauging stations and gate positions served as inputs to the network. Total dissolved solids concentrations data for 20 years were available as network output. In their preliminary work, the authors used the data from 1980–1990 for calibration, and the data from 1971–1980 for validation. Future plans include more rigorous testing of neural networks for salinity predictions. Hutton et al. (1996) also used ANNs for trihalomethane (THM) formation and transport in delta waters. They used neural networks to enhance the capability of an existing model for predicting THM formation and specification by including variable reaction conditions. The ANNs were trained to predict total THM and a bromine incorporation factor based on inputs such as chlorine dose, reaction time, temperature and pH. They utilized 5 nodes in the first hidden layer and 3 in the second hidden layer. Sensitivity analyses showed that ANNs were predicting the right trends of TPH chemistry. The authors concluded that ANN models predict THM formation species and total concentrations in delta waters in an adequate manner.

ANN Applications in Ground Water

It is difficult to separate ground water and water quality as different sections. Many articles have addressed both these topics to some extent (Rogers and Dowla 1994; Roger et al. 1995). In this section, the writers briefly review those applications that emphasized ground-water quality aspects. Aziz and Wong (1992) illustrated the use of ANNs for determining aquifer parameter values from normalized drawdown data obtained from pumping tests—commonly referred to as the inverse problem in ground-water hydrology. This study drew on the pattern recognition ability of an ANN based on aquifer test data. Using measured drawdowns as inputs, neural networks were trained to yield transmissivity *T*, storage coefficient *S*, and the ratio r/B, where r represents the distance to the observation well and B is the aquifer thickness. Both confined and leaky-confined aquifers were considered. The input layer contained 16 nodes representing the confined aquifer data and 12 nodes representing the leaky-confined aquifer inputs. If the ANN was to be used for a strictly confined aquifer, then inputs to the last 12 nodes were set to zero, and vice versa. A three-layer network was trained with data generated from the Theis and Hantush-Jacob solutions. After training, the ANNs were tested on two sets of field data. The values of aquifer parameters predicted by the ANN compared well with results using traditional methods.

When trying to control hydraulic head gradient for ground-water reclamation, the issue of spatial variability becomes very important. Most studies rely on Monte Carlo simulations of the log-transformed hydraulic conductivity field. This technique tends to be computationally expensive. One way to address this problem is to be able to identify if a particular realization is going to be critical based on predefined features of the conductivity field. However, the problem of selecting critical realizations in a Monte Carlo procedure becomes a complex task of pattern recognition. Ranjithan et al. (1993) used a three-layer feedforward network to screen such critical realizations by first identifying characteristics that cause a realization to be a critical one. In a simulation example, the input layer consisted of 102 nodes representing the hydraulic conductivity values in an aquifer discretized by a 10×10 grid system, and mean and standard deviation of these values. The hidden layer had 36 nodes that were decided by a trial-and-error procedure. The network predicted a single output that represented how critical realization was going to be on a normalized scale of 0 to 1—called the level of criticalness by the authors. Fig. 5 is a scatter-plot showing the effectiveness of ANNs in predicting the level of criticalness in Ranjithan et al. (1993). The authors conclude that the pattern recognition strengths of ANNs are particularly useful for identifying the more critical realizations. The problem of identifying optimal pumping strategies to control hydraulic gradient becomes simplified, as only these selected critical realizations have to be further investigated while still maintaining high reliability.

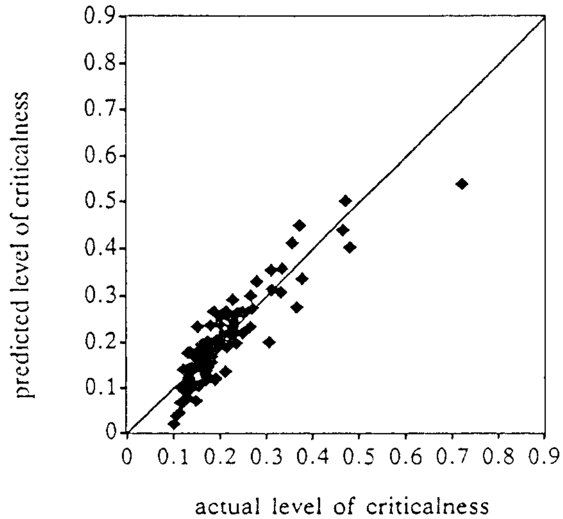

FIG. 5. Performance of Neural Network Model [Training: Number of Training Cases = 100; Number of Potential Reclamation Well Locations = 9. Testing: Number of Testing Cases = 100; Number of Potential Reclamation Well Locations = 3. (Ranjithan et al. 1993)]

Kriging is a spatial interpolation technique that is widely used in geohydrology. Rizzo and Dougherty (1994) introduced the idea of neural kriging for characterization of aquifer properties. A three-layer neural network utilizing the counter propagation algorithm was combined with kriging for estimating hydraulic conductivity. The input nodes represented the coordinates of observation points. The output nodes predicted the class of hydraulic conductivity at various locations. The hidden layer was called the Kohonen layer, as it used a Kohonen unsupervised learning algorithm. Similarly, the output layer was called the Grossberg layer. Without a nonlinear activation function, the competition (winner-take-all) occurred in the hidden layer. The output of the hidden node with the maximum weighted sum of the inputs was set to one, while the output of all the other hidden nodes was set zero. The ANN output was the weighted sum of the output from the winning node. In the training process, only those weights connected to the winning node were updated for each pattern. Based on their results, the authors pointed out that neural kriging produced unbiased estimates of the hydraulic conductivity values at unmeasured locations. They concluded that ANNs could be useful tools in geohydrology when applied to specific problems of aquifer characterization.

In ground-water remediation, a few optimal pumping strategies that meet management goals as well as being cross effective often need to be identified from a vast number (millions) of possible pumping patterns. The conventional method is to study many realizations using a flow and transport model. This brute force method imposes an extremely heavy computational burden. ANNs, when combined with a genetic search algorithm, were shown to accelerate the search process dramatically (Rogers 1992; Rogers and Dowla 1994; Johnson and Rogers 1995; Rogers et al. 1995). However, caution needs to be exercised when applying this method. It can provide meaningful solutions only over the problem dimensions defined by initial model runs used for training. Once the scope of the problem changes, such as an increase in management time frame or the addition of new prospective well locations, training must be repeated with this new information. This is one of the limitations faced by ANNs. The weight space cannot be dynamic but remains frozen after completion of training.

Yang et al. (1997) utilized an ANN to predict water table elevations in subsurface-drained farmlands. Daily rainfall, potential evapotranspiration, and previous water table locations were selected as inputs to the ANN. The output was the current location of the water table. They found that a three-layer feedforward ANN could predict water table elevations satisfactorily after training using observed values. Other ANN applications to problem of irrigation and drainage were investigated by Yang et al. (1996a, b).

ANNs FOR ESTIMATING PRECIPITATION

Precipitation serves as the driving force for most hydrologic processes. It is difficult to predict because it exhibits a large degree of spatial and temporal variability. French et al. (1992) used a three-layer feedforward ANN with back-propagation to forecast rainfall intensity fields at a lead time of 1 hour with the current field as input. Over a hypothetical two-dimensional rainfall domain consisting of a 25×25 regular grid of 4 km resolution, the training patterns were fabricated with a mathematical rainfall simulation model. The 625 output nodes represented the rainfall intensity values of the cells one hour in the future, corresponding to the 625 input nodes representing the current rainfall intensity values at the cells. They studied the impact of different number of hidden nodes, using 15, 30, 45, 60, and 100 hidden nodes. The authors compared ANN generated results with those from persistence and forecasting models. Their results suggested that ANNs performed slightly better than these models during the training stage after a suitable architecture had been identified. But their performance over the testing data set was not satisfactory. They concluded that the ability of an ANN to generalize the underlying rule was strongly dependent on selecting a large enough hidden buyer.

Tohma and Igata (1994) employed a three-layer ANN to estimate rainfall fields based on visible and infrared remote-sensing cloud images in the coastal region of south-western Hokkaido and in a heavy rainfall area of Hokkaido, Japan. The simulation domain was represented by 12×12 pixels of 5×5 km resolution each. Both visible and infrared images were classified into 16 gradations and then converted into values from 0 to 1 for use as network inputs. Both input and output layers had 123 nodes representing the number of domain pixels (12×12) of images except for the 21 pixels lying along the coastline. The network outputs were rainfall intensities associated with pixels. After trying several discrete numbers of hidden nodes (6, 12, 18, 24, 36, 60, and 120), the authors found the optimal number to lie in a range of 24—36. They reported that ANNs could map the relationship between remotely sensed images of clouds and rainfall intensities and provide short-term forecasts of rainfall.

Navone and Ceccatto (1994) have used an ANN model to predict summer monsoon rainfall (SMR) over India. Previous studies have suggested that SMR might be considered as a deterministic (possibly chaotic) process or as a stochastic process. Predictive models for SMR were constructed using a deterministic as well as a stochastic framework. The relative success of both methods prompted Navone and Ceccatto (1994) to combine these two approaches within an ANN framework. First, an ANN (with two input nodes, two hidden nodes, and one output node) is trained to correlate predictors (two indices related to El Nino–Southern Oscillations) with SMR. Second, another ANN (with seven input nodes, four hidden nodes, and one output node) is trained to learn the dynamics of the time series. Seven inputs are chosen from the reconstructed phase space using time-delayed values of SMR. Two output nodes from these two networks are then linked together by connecting those to a new node. The resulting hybrid network was shown to perform 40% more accurately than the best linear statistical method using the same data. It is also shown that the proposed ANN model is better than a more complex linear statistical model with 16 predictors.

Hsu et al. (1996, 1997) developed a modified counterpropagation ANN for transforming satellite infrared images to rainfall rates over a specified area. Their algorithm was similar to the one utilized by Rizzo and Dougherty (1994) in that both used a Kohonen hidden layer and a Grossberg output layer in the three-layer structure. An interesting feature was that the connection weight between the input and hidden layer and between the hidden and output layer could be trained separately. Hsu et al. (1996, 1997) treated as inputs normalized values of infrared brightness temperature of the prediction pixel, surface index representing overland type, mean brightness temperature of the 3×3 pixel window centered at prediction pixel, standard deviation of brightness temperature on those pixels, mean brightness temperature of the 5×5 pixel window centered at the prediction pixel, and standard deviation of brightness temperature of 5×5 pixel window centered at the prediction pixel. The hidden and output layers had 225 nodes each, arranged in a 15×15 matrix. The ANN output was the rainfall rate over the prediction pixel. Training for hidden and output layers was implemented separately. The connection weights between the input and hidden layer were trained using an unsupervised self-organizing clustering procedure based on the principle of competition. For each hidden node, the distance between the input vector and the corresponding weight vector was computed. Only those weights connected to the node with the smallest distance (winner node) were updated for each pattern. A recursive process of competitive node selection and parameter adjustment was continued by repetitive sequential processing of the input data. The size of the neighborhood of the node with the smallest distance as well as the training rate was reduced gradually. The hidden layer acted as a filter that associates each input vector with one of the hidden layer classes. The output of the hidden layer was related to the vector distance for the 3×3 matrix surrounding the winner node and 0 for the others. The output layer served as a linear summation unit that calculated the weighted sum of the outputs of the winner node and its neighbors. The training algorithm for these hidden-output weights was equivalent to a back-propagation algorithm. Hourly and monthly training data were collected from the Japanese Islands and the Florida Peninsula. Fig. 6 reproduces a sample of the results of Hsu et al. (1997), showing comparisons between observed and predicted rainfall rates. The results indicated that ANNs provided a good estimation of rainfall and yielded some insights into the functional relationships between the input variables and the rainfall rate.

Zhang et al. (1997) proposed that ANNs need to be employed in groups when the transformation from the input to the output space is complex. This group theory treats the input-output mapping as being piecewise continuous. The idea is that each network predicts only in the range where the transformation is continuous, while a ''reasoning'' network determines the appropriate summation of responses. The authors were successful in making half-hourly rainfall estimates.

Kuligowski and Barros (1998) present an ANN approach for short-term precipitation prediction. Their model uses a feedforward architecture with upper atmospheric wind direction and antecedent precipitation data from a raingauge network to generate a 0–6 hour precipitation forecast for a target location. Upper air wind direction was used to determine which input variables are relevant for a particular situation. For example, data from a particular predictor gauge may be relevant if it is downstream from the forecast point relative to the direction of precipitation movement (estimated from upper air wind direction). Compared with a persistence model, the

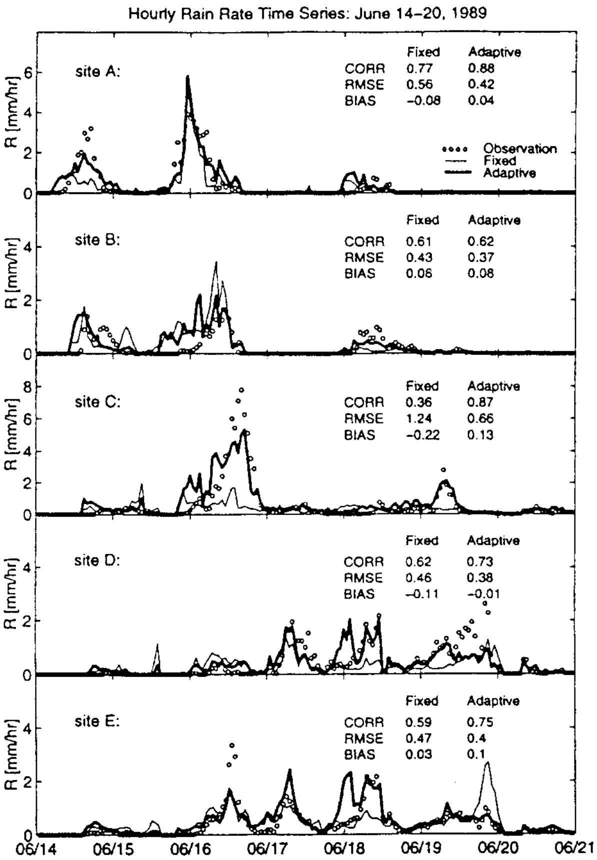

FIG. 6. Comparison of Ground-Based and ANN Model Estimates of Hourly Rainfall Time Series at Five Representative Japanese Island Locations for Period June 14–20, 1989 (Hsu et al. 1997)

proposed ANN model showed significant improvement for short-term precipitation prediction. A particularly powerful feature of the ANN model is its ability to generalize and utilize the latest available information, which makes it attractive for real-time operational prediction.

OTHER APPLICATIONS OF ANNs IN HYDROLOGY

In this section, the writers list miscellaneous applications in hydrology that do not specifically belong to any of the above categories. Raman and Sunilkumar (1995) employed an ANN to model a multivariate water resource time series and compared the results with those obtained by traditional autoregressive moving average (ARMA) models. The objective was to synthesize monthly inflow data for two reservoir sites in the Bharathapuzha basin in south India. A three-layer feedforward ANN with back-propagation was used in this study. The consecutive normalized inflow to the reservoir for two previous months were chosen as inputs. The output was normalized inflow for the current month. They concluded that the results obtained using the ANN compared well with those obtained using statistical models. Raman and Chandramouli (1996) adopted similar ANNs as an alternative tool for describing a general reservoir operating policy. The inputs were initial storage, inflow, and the demand during each fortnight period, the latter being obtained using dynamic programming. The ANN output was the optimal reservoir release. A trial-and-error process indicated that four hidden nodes were best suited for this application. The ANN performed better than three other regression models.

Forecasting river stages accurately in flood prone areas is an extremely crucial problem. For instance, the city of Jagdalpur in India lies on the Indravathi River and is prone to frequent flooding. It is cut off from surrounding areas during such events. Thirumalaiah and Deo (1998) selected a three-layer ANN for predicting flood stages for the city of Jagdalpur. The ANN was trained with back-propagation, conjugate gradient, and the cascade correlation algorithm, respectively. They found that the three training algorithms performed equally well in terms of predicting river stages. Back-propagation needed the most training epochs, and the cascade correlation algorithm needed the least. The ANNs predicted lower water levels accurately but generally underestimated the high water levels. The authors felt that not enough training data were available for proper training of high water level instances.

In operating long-term hydroelectric power systems, a disaggregation procedure is often needed to transform a large-scale problem into small-scale problems. In an attempt to solve this problem for nonlinear relations, Saad et al. (1994) developed a four-layer ANN with back-propagation. The first hidden layer had a linear activation function, while the other layers had a nonlinear activation function. The hydroelectric power systems were aggregated to form only one reservoir. A large number of deterministic optimizations for equally likely sequences of streamflows were used to train the ANN. The inputs were the aggregated storage levels, and the ANN generated the nonlinear functions that minimized the quadratic error between the deterministic optimization and ANN outputs (i.e., disaggregated storage levels). They reported satisfactory performances by the network.

Utilizing an artificial neural network, Zhang et al. (1994) developed an approach to predict daily water demands. Because a number of exogenous factors influence daily water demands, the authors argued that water demands will be related to various inputs in a nonlinear fashion, unlike the linear relationship implied by simplex methods. The water delivery of the previous day, maximum daily temperature, weather, precipitation, and type of day (weekdays or weekends) were selected as inputs to the ANN. Water delivery from the previous day and daily temperature were transformed into the values within (0,1) using a sigmoid function. The weather-related input was assigned values of 0, 0.5, or 1, to represent sunny, cloudy, and rainy days. Similarly, the precipitation-related input was assigned 0, 0.5, or 1 to represent precipitation ranges of (0, 1), (1, 5), or (5, ∞) mm/day, respectively. The type of day was quantified as 0 for Sundays and holidays, and 1 for weekdays. They selected an ANN with 5 input nodes, 17 hidden nodes, and one output node representing the predicted daily water demand. Back-propagation was employed to train the ANN. Their results suggested that the ANN was able to identify the nonlinear pattern of daily water demands. The authors investigated the sensitivity of water demand to the various inputs using the network.

Hydrologic time series is often nonstationary, autocorrelated, and cross correlated. Time series modeling is aimed to account for correlation in real-time data, representing chronological dependence among sequential samples of a given variable. ANNs may be used to generalize the behavior of a hydrologic time series by filtering correlated data. Arguing that ANNs are data driven. Lachtermacher and Fuller (1994) proposed a methodology to relieve the data requirement. Their goal was to forecast annual river flows as a stationary time series, using a three-layer feedforward ANN. They first examined the time series using the Box-Jenkins methods in order to identify the lag components of the series as inputs for an ANN. This procedure reduced the number of primary inputs and consequently decreased the size of the ANN. They generated a synthetic time series using the ANN and compared this with observed values for validation. The authors stressed the importance of an exploratory analysis of the original series before deciding on network structure. They observed that the calibrated ANN model performed marginally better than time series models.

Clair and Ehrman (1998) used ANNs to study the impact of climatic variations on flow discharge and dissolved organic carbon and nitrogen contents over several rivers of the Atlantic Provinces of Canada. They partitioned the monthly data from January 1983 to December 1992 so as to have 84% of the data for training, 12% for cross training, and 4% for testing. A three-layer network with 38 hidden nodes was utilized. The inputs to the network consisted of sample month; basin area and slope; minimum, maximum, and mean monthly temperature; and total rain and snow. To incorporate memory effects, these variables from the previous month were also included as inputs. A sensitivity analysis suggested that hydrological changes resulted in a significant impact on ecological and water resources, especially for the spring months.

Sanchez et al. (1998) used functional link networks to predict the nonfulfillment time (NFT) that is required for design coastal sewage systems. The NFT is a measure of the time during which concentration limits exceed government regulations. Functional link networks do not contain hidden layers. This function is taken over by an expanded input layer. The inputs are augmented by including combinations of input variables or other functions of input variables (derived inputs) to represent the interactions between the causal variables explicitly. They applied this method to two beaches in the city of Gijon in Spain. The primary network inputs consisted of volume and duration of discharge, tidal state at the beginning of discharge, and intensity and direction of wind. A genetic algorithm was utilized to optimize the functional links, and the augmented list of inputs converged to 34 and 20 for the two beaches. The contention of these authors is that neural networks can be used in lieu of mathematical models to avoid complexities and increase speed of simulations.

Other studies are worthy of mention in the context of ANN applications. Schmuller (1990) indicated several environmental applications for ANNs; however, these were not directly related to hydrology. Garrett et al. (1993) discussed several engineering applications, including the optimization of pumping costs. Sun et al. (1995) trained an ANN to estimate the snow-water equivalent from the Spatial Sensor Microwave/Imager. Zhang and Trimble (1995) applied an ANN to forecast water availability using global and solar indices. Kojiri et al. (194) combined fuzzy logic and neural networks for reservoir management. Kao (1996) used an ANN to determine a digital elevation model (DEM)–based drainage model. Dartus et al. (1993) used an ANN to study the flood wave propagation in an open channel. Wen and Lee (1998) were interested in multiobjective optimization of water pollution control and river pollution planning for the Tou-Chen river basin in Taiwan. They used a neural network to replicate a decision maker's preference based on the relative importance of input variables such as 5 day biochemical oxygen demand, cost of wastewater treatment, and river assimilative capacity. The ANN predicted a set of output weights that reflected the decision maker's preferences and could be used in the optimization procedure.

FUTURE OF ANNs IN HYDROLOGY

ANNs have been used by researchers for rainfall-runoff modeling, streamflow prediction, ground-water modeling, water quality, water management, precipitation forecasting, time series, reservoir operations, and other hydrologic applications. These studies have indicated that ANNs can perform as well as existing models. Most mathematical models are able to represent our limited understanding of the physics. The portion that is not well understood is either supplemented with empirical knowledge or is lumped under the various assumptions that may go into the development of the model. In contrast, ANNs can be trained on input-output data pairs with the hope that they are able to mimic the underlying hydrologic process. In essence, the physics is locked up in the set of optimal weights and threshold values and is not revealed back to the user after training. Therefore, artificial neural networks cannot be considered as a panacea for hydrologic problems, nor can they be viewed as replacements for other modeling techniques. Many studies have also warned about the pitfalls of ANNs and caution against their use indiscriminately (Chatfield 1993; Carpenter and Barthelemy 1994; Hill et al. 1994; Jain and Mao 1997). Before embarking on an ambitious use of ANNs, it would be prudent to take a close look at these references and form a more objective opinion. At this stage, they may be perceived as alternative modeling tools that are worthy of further exploration. In this context, the writers discuss some future avenues of ANN research/application that will further the role of ANNs in hydrology. The writers pose some important questions and follow these with a brief discussion.

1. *Can ANNs be made to reveal any physics?* This is perhaps the most commonly asked question by researchers and practitioners alike. For ANNs to gain wider acceptability, it is increasingly important that they have some explanation capability after training has been completed. Most ANN applications have been unable to explain in a comprehensibly meaningful way the basic process by which ANNs arrive at a decision. It is highly desirable that ANNs be capable of imparting an explanation, even if only a partial one, as an integral part of its function. Some effort along these lines has gone into the formation of knowledge-based ANNs and rule extraction techniques. However, such ideas have not been utilized in hydrologic applications. Hsu et al. (1997) provided some heuristic functional relationships between input and output variables of an ANN by using self-organizing feature maps of the input variables. Knowledge-based networks are capable of incorporating some theoretical knowledge from which a network can be constructed. The network can then be further refined using training examples (Towell and Shafik 1994).

   Many neural networks are fairly well adapted to providing statistical interpretations in terms of conditional probabilities, especially in the problem of classification. For instance, a feedforward network can learn the posterior probability of a classification. This means that the neural network takes into account the relative frequency of occurrence of classes by giving more weight to frequently occurring classes. Applications arise in problems of source identification of contaminants, predicting streamflows and river stages for high and low flow events, neural kriging, and many others. Providing some neuristic or probabilistic interpretation would certainly enhance the acceptability of neural networks. In a Bayesian sense, it would be possible to define prior and posterior probabilities based on concepts of unsupervised and supervised learning.

   Physical understanding can be useful in selecting the appropriate neural network and the learning algorithm. Some spatial interpolation problems may be particularly suited for RBF networks. Problems that involve decisions at several levels, or those dealing with hierarchical decision making may be better represented through modular neural networks, which consist of a group of regular neural networks.
2. *Can an optimal training set be identified?* The question is meaningful only if one is willing to look at alternatives other than a brute-force trial-and-error method. In many hydrologic applications, there is a prohibitive cost and

time associated with data collection. Unfortunately, ANNs are data intensive. Without training data, ANN can not learn at all. But repetitive data could slow down ANN training. Therefore, the question of generating an optimal training data set takes on importance in this context. Having too few data sets will lead to poor generalization by the network. An optimal data set for training would be one that fully represents the modeling domain and has the minimum number of data pairs in training. All the hydrologic conditions should be considered when other models are used to generate data for ANN training. Moreover, an optimal data set would be chosen in such a way that the ANN would perform well during prediction. The purpose would be not to invest in generating those training examples that will yield only marginal improvement in network performance. To our best knowledge, this question has not been addressed satisfactorily.

Other related issues arise from this question. Very often, we may have no alternative but to proceed with limited data. Under these circumstances, can we say when generalization will fail so that we understand the range of applicability of the ANN? Unfortunately, this question cannot be answered easily because cross validation does not always provide a complete answer. We need a technique that can anticipate or predict the set of circumstances in which the network will fail. Here again, physically-based methods have an advantage, because the physics can be used to fill the gaps where data is not available. In case of artificial neural networks, it would be useful if the user could identify regions in the input space that are not represented sufficiently in the training data set.

3. *Can ANNs improve on time series analysis?* There are several related questions in this category as well. Many ANN applications have shown how they can predict future streamflows or river stages based on past measurements and have compared their performance with time series models. Apart from some comparative statements about ANNs being better, there have been little, if any, insights provided by ANNs in this context. One of the important issues in time series analyses is memory structure, which is usually characterized by a covariance function. Typically, recurrent ANNs have been used to represent the dependence in time series. Some studies have used historical inputs, with the time window being dictated by the correlation length. These studies have provided useful information, but they have invariably resulted in more complicated networks. Indeed, this has been the primary means of incorporating time-dependent changes into network learning. This would imply that ANNs are restricted to time-homogeneous cases. Consequently, changes in land use, irrigation patterns, crop rotations, and others cannot be accommodated easily at this stage.

   It is obvious that ANNs are based on concepts different from those of time series models. An extremely powerful use of ANNs, and one that has received very little attention in hydrology, would be for data exploration with an added goal being the discovery of unknown dependencies and relationships among the variables. It would appear that ANNs should learn interesting and new nonlinear relationships and bring out features of the input data that are not revealed by other techniques. We need to find innovative ways of extracting this information. Scientific theories cannot be formulated as long as this information is buried in the weight and threshold vectors of a trained network in an incomprehensible fashion.
4. *Can training of ANNs be made adaptive?* Most studies agree that the process of training is an important aspect, and the performance of an ANN is crucially dependent on successful training. This is also the most time-consuming part, with some ANNs requiring several thousands of epochs before training can be accomplished. An unfortunate aspect of this process is that if new training examples become available, there seems to be little alternative but to retrain the network all over again. Thus is true especially when a hydrologic process is dynamic and evolves along with the time. A network already trained on previous examples may not be suitable to the current hydrologic condition. The ability to incorporate this new information into our forecast is desirable. If ANN weights and thresholds can be assigned some physical or statistical interpretation, it is likely that training can be handled in an adaptive fashion.
5. *Are ANNs good extrapolators?* Several studies have reported that ANNs perform well when it is faced with problems that fall within the domain of inputs that were used for training. This can be seen as a problem of interpolation. However, these studies reported that the performance of an ANN deteriorated rapidly when the input vectors were far from the space of inputs used for training. In this sense, ANNs are not very capable when it comes to extrapolation. One way around this problem would be ensure that an ANN does not have to extrapolate by using the widest limits of examples during training. This goes back to the idea of finding an optimal set of training patterns, so that during prediction the ANN does not face any input that is far removed from the examples it saw during training. There are no easy answers to this problem when extrapolation must be performed. All empirical models suffer in this aspect to some degree.

CLOSING REMARKS

The proper use of an ANN requires not only a physical understanding of the hydrological process under consideration but also a knowledge of ANNs and their operation. Trying to extract rules from a network or impart them with some explanation capability will entail extra computer effort. These fundamental aspects will lead to the construction of good training and validation data sets, selection and inclusion of relevant input variables, and development of proper ANN architectures and selection of training algorithms. While the articles reviewed in this paper on ANN applications in hydrology are not exhaustive, it is obvious that ANNs have made a significant impact in this area. Many other applications should be forthcoming, especially if this technique gains acceptance among hydrologists.

ACKNOWLEDGMENT

The Task Committee on Application of Artificial Neural Networks in Hydrology consisted of the following participants: *Chair*—R. S. Govindaraju, Purdue University. *Vice Chair*—A. R. Rao, Purdue University. *Members*—David Leib, Kansas Water Office; Y. M. Najjar, Kansas State University; H. V. Gupta, University of Arizona; A. Hjelmfelt, University of Missouri. *Mail-In Members*—M. Markus, National Weather Service; A. S. Tokar, National Weather Service; S. Islam, University of Cincinnati; J. D. Salas, Colorado State University; C. Ray, University of Hawaii. This paper was prepared with assistance provided by Bin Zhang, a doctoral student at Purdue University. The studies of R. S. Govindaraju and Bin Zhang were supported by NSF Grant EAR-9524578.

APPENDIX. REFERENCES

Aziz, A. R. A., and Wong, K. F. V. (1992). ''Neural network approach to the determination of aquifer parameters.'' *Ground Water*, 30(2), 164–166.

Basheer, I. A., and Najjar, Y. M. (1995). ''Designing and analyzing fixed-bed adsorption systems with artificial neural networks.'' *J. Envir. Syst.*, 23(3), 291–312.
Bonafe, A., Galeati, G., and Sforna, M. (1994). ''Neural networks for daily mean flow forecasting.'' *Hydr. Engrg. Software V*, W. R. Blain and K. L. Katsifarakis, eds., Computational Mechanics Publications, Southampton, U.K., 1, 131–138.
Carpenter, W. C., and Barthelemy, J.-F. (1994). ''Common misconceptions about neural networks as approximators.'' *J. Comp. in Civ. Engrg.*, ASCE, 8(3), 345–358.
Carriere, P., Mohaghegh, S., and Gaskari, R. (1996). ''Performance of a virtual runoff hydrograph system.'' *J. Water Resour. Plng. and Mgmt.*, ASCE, 122(6), 421–427.
Chatfield, C. (1993). ''Neural networks: forecasting breakthrough or passing fad.'' *Int. J. Forecasting*, 9, 1–3.
Clair, T. A., and Ehrman, J. M. (1998). ''Using neural networks to assess the influence of changing seasonal climates in modifying discharge, dissolved organic carbon, and nitrogen export in eastern Canadian rivers.'' *Water Resour. Res.*, 34(3), 447–455.
Daniel, T. M. (1991). ''Neural networks—applications in hydrology and water resources engineering.'' *Proc., Int. Hydrol. and Water Resour. Symp.*, Institution of Engineers, Perth, Australia.
Dartus, D., Courivaud, J. M., and Dedecker, L. (1993). ''Use of a neural net for the study of a flood wave propagation in an open channel.'' *J. Hydr. Res.*, Delft, The Netherlands, 31(2).
Dawson, C. W., and Wilby, R. (1998). ''An artificial neural network approach to rainfall-runoff modeling.'' *Hydrological Sci.*, 43(1), 47–66.
Fernando, D. A. K., and Jayawardena, A. W. (1998). ''Runoff forecasting using RBF networks with OLS algorithm.'' *J. Hydrologic Engrg.*, ASCE, 3(3), 203–209.
Flood, I., and Kartam, N. (1994). ''Neural networks in civil engineering. I: Principles and understandings.'' *J. Comp. in Civ. Engrg.*, ASCE, 8(2), 131–148.
Flood, I., and Kartam, N. (1997). ''Systems.'' *Artificial Neural Networks for Civ. Engrs.: Fundamentals and Applications*, Kartam, I. Flood, and J. H. Garrett Jr., eds., ASCE, New York, 19–43.
French, M. N., Krajewski, W. F., and Cuykendal, R. R. (1992). ''Rainfall forecasting in space and time using a neural network.'' *J. Hydrol.*, Amsterdam, 137, 1–37.
Garrett, J. H., et al. (1993). ''Engineering applications of artificial neural networks.'' *J. Intel. Manuf.*, 4, 1–21.
Gupta, H. V., Ksu, K., and Sorooshian, S. (1997). ''Superior training of artificial neural networks using weight-space partitioning.'' *Proc., IEEE Int. Conf. on Neural Networks*, Institute of Electrical and Electronics Engineers, New York.
Halff, A. H., Halff, H. M., and Azmoodeh, M. (1993). ''Predicting runoff from rainfall using neural networks.'' *Proc., Engrg. Hydrol.*, ASCE, New York, 760–765.
Haykin, S. (1994). *Neural networks: a comprehensive foundation*. MacMillan, New York.
Hill, T., Marquez, L., Connor, M. O., and Remus, W. (1994). ''Artificial neural networks for forecasting and decision making.'' *Int. J. Forecasting*, 10, 5–15.
Hjelmfelt, A. T., and Wang, M. (1993a). ''Artificial neural networks as unit hydrograph applications.'' *Proc., Engrg. Hydrol.*, ASCE, New York, 754–759.
Hjelmfelt, A. T., and Wang, M. (1993b). ''Runoff simulation using ANN.'' *Proc., 4th Int. Conf. in the Application of Artificial Intelligence to Civ. and Struct. Engrg.: NN and Combinatorial Optimization in Civ. and Struct. Engrg.*, B. H. V. Topping and A. I. Khan, eds., Civl-Comp Ltd., Edinburgh, U.K., 517–522.
Hjelmfelt, A. T., and Wang, M. (1993c). ''Runoff hydrograph estimation using artificial neural networks.'' *Proc., ASAE Conference*, American Society of Agricultural Engineers, St. Joseph, Mich.
Hjelmfelt, A. T., and Wang, M. (1996). ''Predicting runoff using artificial neural networks.'' *Surface-Water Hydrol.*, 233–244.
Hsu, K., Gao, X., Sorooshain, S., and Gupta, H. V. (1997). ''Precipitation estimation from remotely sensed information using artificial neural networks.'' *J. Appl. Meteorology*, 36(9), 1176–1190.
Hsu, K., Gupta, H. V., and Soroochian, S. (1997). ''Application of a recurrent neural network to rainfall-runoff modeling.'' *Proc., Aesthetics in the Constructed Envir.*, ASCE, New York, 68–73.
Hsu, K., Gupta, H. V., Sorooshian, S., and Gao, X. (1996). ''An artificial neural network for rainfall estimation from satellite infrared imaginary.'' *Applications of Remote Sensing in hydrol., Proc., 3rd Int. Workshop, NHRI Symp. No. 17*, NASA, Greenbelt, Md.
Hsu, K., Gupta, H. V., and Sorooshian, S. (1995). ''Artificial neural network modeling of the rainfall-runoff process.'' *Water Resour. Res.*, 31(10), 2517–2530.
Hutton, P. H., Sandhu, N., and Chung, F. I. (1996). ''Predicting THM formation with artificial neural networks.'' *Proc., North Am. Water and Envir. Conf.*, ASCE, New York, 3557–3556.
Jain, A. K., and Mao, J. (1997). ''Guest editorial: special issue on artificial neural networks and statistical pattern recognition.'' *IEEE Trans. on Neural Networks*, 8(1), 1–3.
Jayawardena, A. W., and Fernando, D. A. K. (1995). ''ANN in hydrometeorological modeling.'' *Proc., 4th Int. Conf. in the Application of Artificial Intelligence to Civ. and Struct. Engrg.: Devel. in NN and Evolutionary Computing for Civ. and Struct. Engrg.*, B. H. V. Topping and A. I. Khan, eds., Civil-Comp, Ltd., Edinburgh, U.K., 115–120.
Jayawardena, A. W., and Fernando, D. A. K. (1996). ''Comparison of multi-layer perceptron and radial basis function network as tools for flood forecasting.'' *Proc., North Am. Water and Envir. Conf.*, ASCE, new York, 457–458.
Johnson, V. M., and Rogers, L. L. (1995). ''Location analysis in groundwater remediation using NN.'' *Groundwater*, 33(5), 749–758.
Kang, K. W., Kim, J. H., Park, C. Y., and Ham, K. J. (1993). ''Evaluation of hydrological forecasting system based on neural network model.'' *Proc., 25th Congress of Int. Assoc. for Hydr. Res.*, International Association for Hydraulic Research, Delft, The Netherlands, 257–264.
Kao, J. J. (1996). ''Neural net for determining DEM-based model drainage pattern.'' *J. Irrig. and Drain. Engrg.*, ASCE, 122(2), 112–121.
Karunanithi, N., Grenney, W. J., Whitley, D., and Bovee, K. (1994). ''Neural networks for river flow prediction.'' *J. Comp. in Civ. Engrg.*, ASCE, 8(2), 201–220.
Kojiri, T., Ito, K., and Sakakimo, S. (1994). ''Rainfall estimation and real-time reservoir operation with neural network and fuzzy inference.'' *Application of artificial intelligence in engineering*, G. Rzevski, R. A. Adey, and D. W. Russel, eds., Computational Mechanics, Southampton, U.K.
Kuligowski, R. J., and Barros, A. P. (1998). ''Experiments in short-term precipitation forecasting using artificial neural networks.'' *Monthly Weather Rev.*, 126(2), 470–482.
Lachtermacher, G., and Fuller, J. D. (1994). ''backpropagation in hydrological time series forecasting.'' *Stochastic and statistical methods in hydrology and environmental engineering time series analysis in hydrology and environmental engineering*, Vol. 3, K. W. Hipel, et al., eds., Kluwer, Dordrecht, The Netherlands, 229–242.
Maier, H. R., and Dandy, G. C. (1996). ''The use of artificial neural networks for the prediction of water quality parameters.'' *Water Resour. Res.*, 32(4), 1013–1022.
Markus, M., Salas, J. D., and Shin, H.-K. (1995). ''Predicting streamflows based on neural networks.'' *Proc., 1st Int. Conf. on Water Resour. Engrg.*, ASCE, New York, 1641–1646.
Mason, J. C., Price, R. K., and Tem'me, A. (1996). ''A neural network model of rainfall-runoff using radial basis functions.'' *J. Hydr. Res.*, Delft, The Netherlands, 34(4), 537–548.
McCuen, R. H. (1997). *Hydrologic analysis and design*, 2nd Ed., Prentice Hall, Upper Saddle River, N.J.
Minns, A. W., and Hall, M. J. (1996). ''Artificial neural networks as rainfall-runoff models.'' *Hydrologic Sci.*, 41(3), 1996.
Morshed, J., and Kaluarachchi, J. J. (1998). ''Parameter estimation using artificial neural network and genetic algorithm for free-product and recovery.'' *Water Resour. Res.*, 34(5), 1101–1113.
Muttiah, R. S., Srinivasan, R., and Allen, P. M. (1997). ''Prediction of two-year peak stream discharges using neural networks.'' *J. Am. Water Resour. Assoc.*, 33(3), 625–630.
Navone, H. D., and Ceccatto, H. A. (1994). ''Predicting Indian monsoon rainfall: a neural network approach.'' *Climate Dyn.*, 10, 305–312.
Poff, N. L., Tokar, S., and Johnson, P. (1996). ''Steam hydrological and ecological responses to climate change assessed with an artificial neural network.'' *Limnol. and Oceanog.*, 41(5), 857–863.
Raman, H., and Chandramouli, V. (1996). ''Deriving a general operating policy for reservoirs using neural networks.'' *J. Water Resour. Plng. and Mgmt.*, ASCE, 122(5), 342–347.
Raman, H., and Sunilkumar, N. (1995). ''Multi-variate modeling of water resources time series using artificial neural networks.'' *Hydrological Sci.*, 40, 145–163.
Ranjithan, S., Eheart, J. W., and Rarret Jr., J. H. (1993). ''Neural network-screening for groundwater reclamation under uncertainty.'' *Water Resour. Res.*, 29(3), 563–574.
Ray, C., and Klindworth, K. K. (1996). ''Use of artificial neural networks for agricultural chemical assessment of rural private wells.'' *Proc., North Am. Water and Envir. Conf.*, ASCE, New York, 1687–1692.
Rizzo, D. M., and Dougherty, D. E. (1994). ''Characterization of aquifer properties using artificial neural networks: neural kriging.'' *Water Resour. Res.*, 30(2), 483–497.
Rogers, L. L. (1992). ''Optimal groundwater remediation using artificial neural network and the genetic algorithm,'' PhD disertation, Stanford University, Stanford, Calif.

Rogers, L. L., and Dowla, F. U. (1994). ''Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling.'' *Water Resour. Res.*, 30(2), 457–481.

Rogers, L. L., Dowla, F. U., and Johnson, V. M. (1995). ''Optimal field-scale groundwater remediation using neural networks and the genetic algorithm.'' *Envir. Sci. and Technol.*, 29(5), 1145–1155.

Rogers, L. L., Johnson, V. M., and Dowla, F. U. (1993). ''Network dissection of neural networks used in optimal groundwater remediation.'' *Proc., 2nd USA/CIS Joint Conf. on Envir. Hydrol. and Hydrogeology*, American Institute of Hydrology, Arlington, Va.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). ''Learning internal representations by error propagation.'' *Parallel distributed procesing*, Vol. 1, MIT Press, Cambridge, Mass., 318–362.

Saad, M., Turgeon, A., Bigrs, P., and Duquete, R. (1994). ''Learning disaggregation technique for the operation of long-term hydro-electric power systems.'' *Water Resour. Res.*, 30(1), 3195–3202.

Sanchez, L., Arroyo, V., Garcia, J., Koev, K., and Revilla, J. (1998). ''Use of neural networks in design of coastal sewage systems.'' *J. Hydr. Engrg.*, ASCE, 124(5), 457–464.

Sandhu, N., and Finch, R. (1996). ''Emulation of DWRDSM using artificial neural networks and estimation of Sacramento River flow from salinity.'' *Proc., North Am. Water and Envir. Conf.*, ASCE, New York, 4335–4340.

Schmuller, J. (1990). ''Neural networks and environmental applications.'' *Expert systems for environmental applications*, J. M. Hushon, ed., American Chemical Society, Washington, D.C., 52–68.

Shamseldin, A. Y. (1997). ''Application of a neural network technique to rainfall-runoff modeling.'' *J. Hydrol.*, Amsterdam, 199(1997), 272–294.

Smith, J., and Eli, R. N. (1995). ''Neural-network models of rainfall-runoff process.'' *J. Water Resour. Plng. and Mgmt.*, ASCE, 121(6), 499–508.

Starrett, S. K., Najjar, Y. M., and Hill, J. C. (1996). ''Neural networks predict pesticide leaching.'' *Proc., Am. Water and Envir. Conf.*, ASCE, New York, 1693–1698.

Sun, C., Neale, C. M. U., and McDonnell, J. J. (1993). ''The potential of using ANN in estimation of snow water equivalent from SSM/I data.'' *Proc., Engrg. Hydrol. Symp.*, ASCE, New York,

Tawfik, M., Ibrahim, A., and Fahmy, H. (1997). ''Hysteresis sensitive neural network for modeling rating curves.'' *J. Comp. in Civ. Engrg.*, ASCE, 11(3), 206–211.

Taylor, J. G. (ed.). (1996). *Neural networks and their applications*. Wiley, Chichester, U.K.

Thirumalaiah, K., and Deo, M. C. (1998). ''River stage forecasting using artificial neural networks.'' *J. Hydrologic Engrg.*, ASCE, 3(1), 26–32.

Tohma, S., and Igata, S. (1994). ''Rainfall estimation from GMS imagery data using neural networks.'' *Hydraulic engineering software V*, Vol. 1, W. R. Blain and K. L. Katsifarakis, eds., Computational Mechanics, Southampton, U.K., 121–130.

Tokar, A. S., and Johnson, P. A. (1999). ''Rainfall-runoff modeling using artificial neural networks.'' *J. Hydrologic Engrg.*, ASCE, 4(3), 232–239.

Tokar, A. S., and Markus, M. (1997). ''Artificial neural networks and conceptual models in water management of small basins in the central United States.'' *Proc., 3rd Int. Conf. on FRIEND*, International Association of Hydrological Sciences, Wallingford, U.K.

Towell, G. G., and Shafik, J. W. (1994). ''Knowledge-based artificial neural networks.'' *Artificial Intelligence*, 70, 119–165.

Wen, C.-W., and Lee, C.-S. (1998). ''A neural network approach to multiobjective optimization for water quality management in a river basin.'' *Water Resour. Res.*, 34(3), 427–436.

Yang, C. C., Prasher, S. O., and Lacroix, R. (1996a). ''Application of artificial neural networks to simulate water-table depth under subirrigation.'' *Can. Water Resour. J.*, Lethbridge, Canada, 21(1), 27–44.

Yang, C. C., Prasher, S. O., and Lacroix, R. (1996b). ''Application of artificial neural networks to land drainage engineering.'' *Trans., ASAE*, 39(2), 525–533, 1996.

Yang, C. C., Prasher, S. O., Lacroix, R., Sreekanth, S., Patni, N. K., and Masse, L. (1997). ''Artificial neural network model for subsurface-drained farmland.'' *J. Irrig. and Drain. Engrg.*, ASCE, 123(4), 285–292.

Zhang, E. Y., and Trimble, P. (1995). ''Forecasting water availability by applying neural networks with global and solar indices.'' *Proc., Engrg. Hydrol. Symp.*, ASCE, New York,

Zhang, M., Fulcher, J., and Scofield, R. A. (1997). ''Rainfall estimation using artificial neural network group.'' *Neurocomputing*, 16, 97–115.

Zhang, S. P., Watanabe, H., and Yamada, R. (1994). ''Prediction of daily water demands byt neural networks.'' *Stochastic and statistical method in hydrology and environmental engineering*, Vol. 3, K. W. Hipel et al., eds., Kluwer, Dordrecht, The Netherlands, 217–227.

Zhu, M., Fujita, M., and Hashimoto, N. (1994). ''Application of neural networks to runoff prediction.'' *Stochastic and statistical method in hydrology and environmental engineering*, Vol. 3, K. W. Hipel et al., eds., Kluwer, Dordrecht, The Netherlands, 205–216.