Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Hydrology 317 (2006) 31–48

Journal of Hydrology

www.elsevier.com/locate/jhydrol

Modeling a densely urbanized watershed with an artificial neural network, weather radar and telemetric data

Augusto José Pereira Filho[a,*], Cláudia Cristina dos Santos[b]

[a]*Universidade de São Paulo, São Paulo-USP/IAG/DCA, Rua do Matão, 1226, São Paulo, SP 05508-900, Brazil*
[b]*Instituto Nacional de Pesquisas Espaciais-INPE/DSR, Av. dos Astronautas,*
1758, Jardim da Granja, São José dos Campos, SP 12227-010, Brazil

Received 15 May 2003; revised 20 April 2005; accepted 4 May 2005

Abstract

Artificial neural networks (ANN) are widely used in a myriad of fields of research and development, including the predictability of time series. This work is concerned with one of such applications to simulate and to forecast stage level and streamflow at the Tamanduateí river watershed, one of the main tributaries of the Alto Tietê river watershed in São Paulo State, Brazil. This heavily urbanized watershed is within the Metropolitan Area of São Paulo (MASP) where recurrent flash floods affect a population of more than 17 million inhabitants. Flash floods events between 1991 and 1995 were selected and divided up into three groups for training, verification and forecasting purposes. Weather radar rainfall estimation and telemetric stage level and streamflow data were input to a three-layer feed forward ANN trained with the Linear Least Square Simplex training algorithm (LLSSIM) by Hsu et al. [Hsu, K.L., Gupta, H.V., Sorooshian, S., 1996. A superior training strategy for three-layer feed forward artificial neural networks. Tucson, University of Arizona. (Technique report, HWR no. 96-030, Department of Hydrology and Water Resources)]. The performance of the ANN is improved by 40% when either streamflow or stage level were input together with the rainfall. The ANN simulated flood waves tend to be dominated by phase errors. The ANN showed slightly better results then a multi-parameter auto-regression model and indicates its usefulness in flash flood forecasting.

Keywords: Urban hydrology; Artificial neural networks; Weather radar; Nowcasting

1. Introduction

Simplified rainfall-runoff models are usually applied to represent hydrologic processes over watersheds with scarce hydrometeorological data (Viessman Jr. and Lewis, 1996). Even watersheds with good monitoring systems and more elaborated models are limited in face of complex hydrological processes (Singh, 1988) such as precipitation, evaporation, infiltration and streamflow that are difficult to quantify with good accuracy. Moreover, some surface processes such as streamflow are strongly depended upon the physical characteristics of the watershed like topography, vegetation cover, soil type, hydraulic structures, urbanization, and many others.

* Corresponding author. Fax: +55 11 3091 4714.
E-mail addresses: apereira@model.iag.usp.br (A.J. Pereira Filho), claudiac@ltid.inpe.br (C.C. dos Santos).

0022-1694/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.jhydrol.2005.05.007

Watershed analysis and modeling require extensive time series of streamflow and rainfall data, not frequently available. Many analytical methods have been developed to estimate streamflow from rainfall measurements over the watershed (Viessman Jr. and Lewis, 1996; Franchini and Pacciani, 1991), as well as stochastic and statistical ones where no measurements are available. These simplified methods are usually applied to engineering projects, water resources management, flood forecasting and others. A great deal of the hydrological and hydraulic processes is neglected. Other more elaborated methods such as the Shamseldin et al. (1996) ensemble streamflow estimates from several models. Each of them represents certain important physical aspects.

More recently, the ANN method was successfully utilized to model time series in a variety of applications in science and engineering (Vemuri, 1994), also applied to convert rainfall into streamflow (Luk et al., 2000; Hsu et al., 1995; French et al., 1992) without prescribing the hydrological processes; non-linear systems are handled without explicitly solving differential equations (Hsu et al., 1995).

While a feasible contingency where limited rainfall and streamflow data are available, the ANN does not replace physically based rain-runoff methods, but Shamseldin (1997) indicates better ANN simulation results when compared to classical rainfall-runoff models. Additionally, Thirumalaiah and Deo (1998) were able to forecast flood waves with an ANN based on streamflow and physical features of the watershed. French et al. (1992) applied an three-layer ANN to predict the spatial and temporal distribution of rainfall. Xiao and Chandrasekar (1995) used an ANN to obtain rainfall rates from weather radar reflectivity and rain gauges measurements. Weights were assigned to radar reflectivity (input) and rainfall rate (output) measurements to obtain an optimum ZR relationship. These are some examples that indicate the usefulness of the ANN in hydrology and hydrometeorology studies.

Urban hydrology are not well represented by deterministic models since it is difficult to quantify processes with equations due to missing or incomplete data. Overall, processes and formulae are simplified and fewer parameters are used to keep the model and required data manageable. Simplified models can make modeling feasible and enhance insight in the processes. When the natural process is not sufficiently known, but data exist, a statistical approach can also be applied. Watersheds with long data time-series can be represented by stochastic models, but noise and changes can reduce modeling capabilities. Furthermore, some issues are not addressed: (1) unknown processes; (2) decision making; (3) data intensive tasks; (4) changing environments. The ANN can deal with such issues, namely, processing speed, fault tolerance, adaptability and learning capabilities (Loke, 1995).

As described by Loke (1995), ANNs relate an input vector (e.g., rainfall accumulation) to an output vector (streamflow) by means of a large number of highly interconnected, simple input and output devices termed neurons. These neurons can be connected in several ways to form special groups of neurons with distinct architectures. The ANN has to be trained through a learning procedure that uses a data set of input and output vectors to adjust the networks internal parameters. After training, the ANN is verified against an independent data set. If the verification procedure is not satisfactory, the network has to be retrained, using different data sets or modified internal settings. Training and verification are repeated until the ANN performs adequately. As the performance of an ANN is influenced by many parameters (e.g. network structure or learning algorithm), numerous types of ANNs exist, all with their specific application purposes. ANNs are divided into prediction, simulation, classification, optimization and identification problems. ANNs can be used in urban hydrology applications for runoff and flow forecast, flow and pollution simulation, control strategy definition or system parameter identification.

Therefore, the objective of this work is to apply and ANN to simulate and to forecast flash floods in the Tamanduateí river watershed, a heavily urbanized area in Eastern São Paulo State (Fig. 1). Rainfall data were obtained from the São Paulo weather radar—SPWR (Pereira Filho, 1999). Streamflow were estimated from data measured by an automatic stage level station at the outlet of the watershed (Fig. 1). A three-layer feed forward ANN (Hsu et al., 1996) was trained with the LLSSIM (Hsu et al., 1995). Noteworthy, feed forward ANNs have been widely used (Karunanithi et al., 1994; Crespo and Mora, 1993). Their structure is post-defined. Simple structures might not have enough degrees of freedom to learn

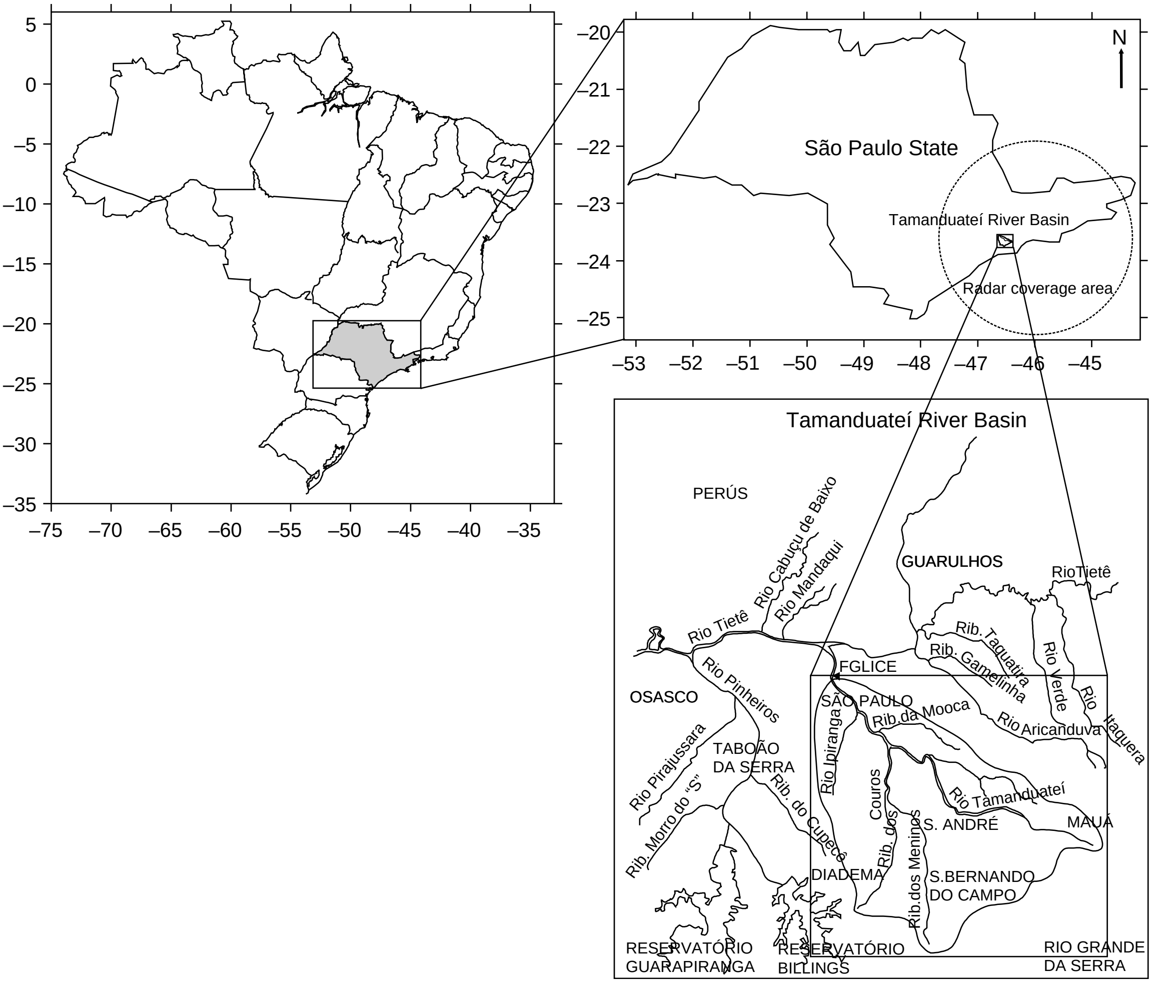


Fig. 1. Map of the Tamanduateí Watershed in São Paulo State (upper right), in Brazil (left). The circumference indicates the radar surveillance area. The outlet is indicated (FGLICE).

the process adequately, while complex ones might not converge to realistic patterns.

2. Methodology

2.1. Artificial neural networks

The artificial neurons are very similar in structure and operation to the biological neurons. Fig. 2 shows input signals $X_1, X_2, \ldots, X_N$ that represent the dendrites and weights $W_1, W_2, \ldots, W_N$ that represent the synaptic processing of the input signal. They can be positive or negative. A neuron is activated by a synapse, the weight that emulates the strength of or the degree to which the neuron should allow shot signals into the connection. The signal is processed by an activation function that determines the strength of the output signal, equivalent to a synapse.

Artificial neurons are composed of an integration and a transfer activation function. The integration function adds up individual weighted signals. Specific weights are assigned to each synapse. The activation function yields the intensity of the signal to be transmitted to the connections in adjacent layers. Mathematically, a neuron's signal processing can be

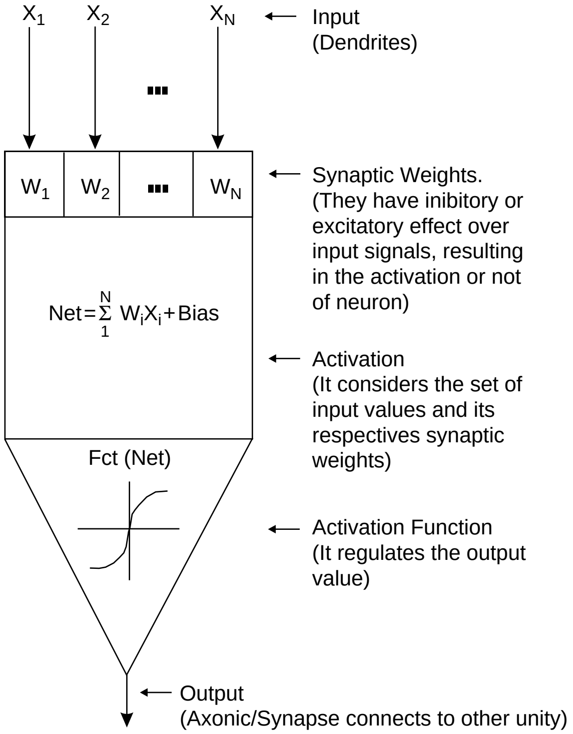

X1
X2
XN
Input
(Dendrites)
W1
W2
WN
Synaptic Weights.
(They have inibitory or excitatory effect over input signals, resulting in the activation or not of neuron)
Net=Σ WiXi+Bias
Activation
(It considers the set of input values and its respectives synaptic weights)
Fct (Net)
Activation Function
(It regulates the output value)
Output
(Axonic/Synapse connects to other unity)

Fig. 2. Example of an artificial neuron. After Osório and Vieira (1999).

described as follows:

$$\mu_k = \sum_{j=1}^{p} w_{kj} x_j \tag{1}$$

and

$$y_k = \varphi(\mu_k - \theta_k) \tag{2}$$

where,

$x_1, x_2, \ldots, x_p$ are the input signals;
$w_{k1}, w_{k2}, \ldots, w_{kp}$ are the synaptic weights of neuron k;
μ_k is the integrated output;
θ_k is a threshold;
ϕ is the activation or transfer function;
y_k is the neuron output signal.

A group of artificial neurons form a neural network. Inputs can be linked to many neurons. Each one can receive several weights that result in a series of outputs (Fig. 3). These synaptic connections are similar to dendrites in a biological system.

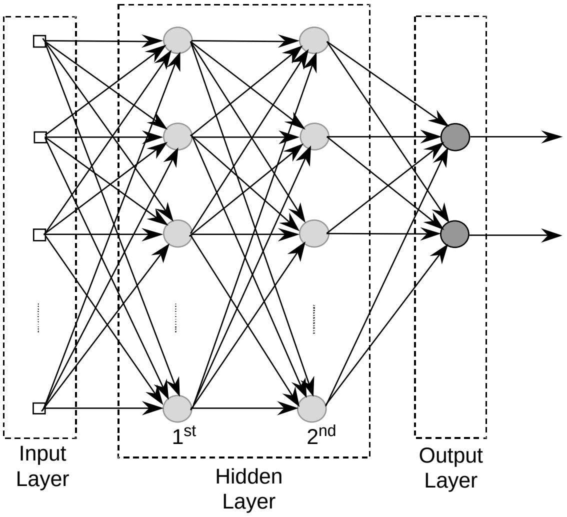


Fig. 3. Example of a two hidden layers perceptron ANN. Adapted from Haykin (1994).

The transfer function ϕ defines the input signal activity level of the processing element (Fig. 3). Thus, the transfer function specifies the signal to be transmitted to other neurons. It is known as the logistic or sigmoid function. There are many types of transfer functions. The most common ones are the sigmoid and the hyperbolic functions. Both are symmetric, continuous and with continuous derivatives, important mathematical characteristics.

A neuron can be totally ($\phi = 1$) and partially activated ($0 < \phi < 1$) or deactivated ($\phi = 0$). For instance, in a given hydrological system, $\phi = 1$ would represent complete surface runoff, $\phi = 0$ would be equivalent to complete infiltration, and $0 < \phi < 1$ would be intermediate states (Pedrollo, 1999).

The topology of an ANN comprises the number of layers in the network, the number of nodes in each layer, the type of connection between nodes and the overall structure. A multi-layer ANN has the following general characteristics: (1) an input layer that shares external information with the internal layers; (2) hidden layers that process the input signal by means of an integrator and activation functions that extract patterns; (3) output layer that assemble the response pattern (Fig. 3).

In vector notation, a three-layer ANN can be written as ANN(l,m,n), where the indices l, m, and n are the number of nodes in the input, hidden and output layers, respectively.

In a feed forward ANN, signals propagate from the input to the output layer one way (Fig. 3). It can have more than one hidden layer. The layers are fully connected so that a neuron shares information with all neurons of the next layer to solve even complex mapping. Moreover, it can also classify complex patterns even when they are not used in the training process.

Training performance depend on the information content of problem to be solved. An optimization algorithm is usually applied to estimate weights, the number of layers and neurons or nodes per layer. An ANN learns to extract information from given patterns to generate specific solutions to the problem at hand. This training process ends when an overall solution is found. Afterwards, the rules are applied to solve similar problems.

Several training procedures are available. Basically, they differ on how the weights are obtained. For instance, in the back propagation algorithm (Gupta et al., 1997), an input pattern is processed layer-by-layer to yield an output signal that is compared to an observed pattern. The difference between the ANN output and the observation are propagated backward to modify the response. This procedure is repeated until an arbitrary overall minimum difference is obtained. The optimum solution might not be reached due to errors and lack of convergence caused by local minima. An efficient training algorithm is briefly presented in the next section The ANN structure is nonlinear and can be used to model complex processes. It has an advantage upon more traditional methods such as auto-regressive model briefly describe in the next section. Furthermore, the ANN model can be easily retrained to include changes in to be adapted to other watersheds. Additional advantages of the ANN are: (1) they more adequate to dynamic prediction problems since the weights can be modified as new information are made available; (2) small errors are not amplified because the processing is distributed; (3) do not require storage of all data previously processed; (4) only input and output data are required for the training process.

2.2. The training algorithm

The linear least squares simplex or the LLSSIM algorithm (Hsu et al., 1995; Gupta et al., 1997) is a hybrid method that optimizes the training of feed forward ANNs. Basically, the LLSSIM splits the weights into two training strategies to minimize the search for the global minima (Scalero and Tepedelenlioglu, 1992). The simplex algorithm (Duan et al., 1992) is a random multi-initialization procedure that reduces trapping by local minima (Hsu et al., 1996). For hydrological applications, the data sets are normalized:

$$N_i = N_{\min} + \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}(N_{\max} - N_{\min}) \tag{3}$$

where,

X_i hydrological variable at the ith time step;
$X_{\min}$ lowest value of X in the hydrological time series;
$X_{\max}$ highest value of X in the hydrological time series;
N_i normalized value of X at the ith time step;
$N_{\min}$ lowest value of N allowed;
$N_{\max}$ highest value of N allowed;

The imposed lower and upper limits of the normalized variables prevent over-shooting (Smith, 1993). The time series of precipitation and streamflow were normalized in the range of 0 to 1 and 0.1 to 0.9, respectively. Additional details are in Appendix A.

2.3. Auto-regressive (AR) models

Runoff time series can be regarded as a random process (Fiering; Jackson, 1971), and so can be described by a stochastic model through probabilistic laws. One of such models is the auto-regressive (AR) model. This parametric model can also be used in prediction. One basic assumption in AR modeling is that the time series is random, stationary (the long term average is constant) and in stable equilibrium (Morettin, 1982). In general, the ith term of the time series, x_i, is decomposed into two parts:

$$x_i = d_i + e_i \tag{4}$$

where d_i is the deterministic component that is obtained from a parametric function based on the time series. Normally, d_i *is a function of the mean and runoff variance and past runoff* (e.g., x_{i-1}, x_{i-2}). The random component in Eq. (4) is obtained from a probabilistic function or a pattern. Thus, only the deterministic component is time dependant. Considering the model below:

$$x_i = \phi_1 x_{i-1} + \phi_2 x_{i-2} + \cdots + \phi_p x_{i-p} + a_i \qquad (5)$$

where ϕ_1, ϕ_2, …, ϕ_p are weights and a_i a white noise. Eq. (5) is known as a p-order auto-regressive process or simple a AR(p) process. Defining the AR(p) operator as $(1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p)x_i = a_t$, then Eq. (5) can be written as:

$$\phi(B)x_i = a_t \qquad (6)$$

Such that x_i is determined by x_{i-1} and the noise at time t. It is a Markov process (Fiering; Jackson, 1971). The AR(p) is stationary (Box and Jenkins, 1976) if the roots of $\phi(B)=0$ are not equal to 1 (Moretin and Toloi, 1982).

AR models are normally used in the analysis of data time series. For instance, in a simple linear regression model, an independent variable X (rainfall) is used to model a dependent variable Y (runoff). It can be used to predict future states of the dependent variable. The model can be verification through the analysis of residues. The AR model can be readjusted when new data sets are available, though a new complete regression analysis is required to include previous and new data sets. Sometimes, outliers can significant affect model performance as well as changes in hydrological processes implicit in the regression analysis. Moreover, nonlinear processes can not de handled by AR models.

2.4. The study area

Fig. 1 shows the Tamanduateí watershed, located in Eastern São Paulo State, Brazil. It is an important tributary of the Alto Tietê River (Water and Electrical Energy Department (DAEE), 1988). This densely urbanized watershed has a drainage area of 310 km^2 with an estimated time of concentration of about 4 h. More than 80% of its area is impermeable, especially upstream from the outlet (CTO, 1997). The Tamanduateí river channel has a regular concrete cross-section that was projected to support a peak flow of around 485 m^3 s^{-1} for a return period of 500 years (CTO, 1997). As the population grows every year without any effective public policy, so do the urbanization and the flood waves that cause severe inundation.

The Tamanduateí watershed is a part of a much larger urban area termed the Metropolitan Area of São Paulo (MASP) with a population of more than 17 million inhabitants. Thermodynamically, it constitutes a huge heat island. Under quiescent synoptic conditions during the rainy season from October to March it tends to induce stronger convection and so more rainfall associated with the local sea breeze circulation which brings moisture from the Atlantic Ocean (Pereira Filho, 1999). With increased precipitation and lower infiltration rates, flash floods are becoming more and more severe with increasing social and economical loses. Thus, hydrological models such the one proposed in the present work are of great assistance to give support to the civil defense and to the local and state governments.

2.5. Data sets

Precipitation accumulations were estimated from radar measurements of reflectivity. The São Paulo weather radar (SPWR) is located in the Upper Alto Tietê River (Fig. 1). Radar volume scans were made every 10 min. Reflectivity measurements were transformed into rainfall rates by the classical Marshall and Palmer (MP, 1948) ZR relationship. Afterwards, rainfall rates were interpolated to a constant altitude of 3.0 km with a horizontal resolution of 2 km × 2 km. Precipitation accumulation were obtained as follows:

$$P = \sum_{i=1}^{n} \frac{R}{6} \qquad (7)$$

where,

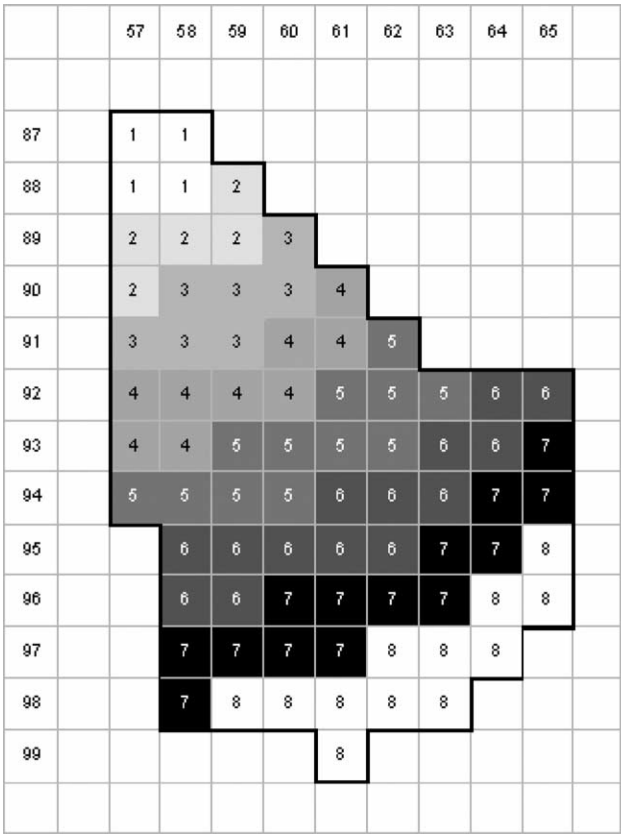
P precipitation accumulation (mm);
R rainfall rate (mm h^{-1});
n number of time steps of ten minutes each.

The SPWR rainfall estimates are susceptible to many sources of errors such as electronic calibration, the use of the MP relationship, bright band and range effect (Pereira Filho, 2003). In this work, radar rainfall estimates were not analyzed together with rain gauges to minimize errors since the available network of rain gauges is sparse and presented many data gaps.

Moreover, rain gauge measurement errors can be large because of the rainfall spatial variability and wind effects for intense convective systems as the ones associated with floods in the MASP and so could not be used as ‘ground truth’. In the present work, the SPWR data were used without any corrections because of the limitations above. In this work, flood events with radar rainfall estimations not consistent with the respective flood waves were arbitrarily eliminated. Radar rainfall estimates can be improved by integrating both radar estimates and rain gauge measurements through a statistical objective analysis scheme (Pereira Filho et al., 1998). Pereira Filho and Crawford (1999) have shown that a 30% error in radar rainfall estimation can result in a peak flow error greater than 300% for small flood waves.

The Tamanduateí watershed was mapped to the SPWR matrix coordinate system (Fig. 4) to form a mosaic of radar rainfall estimations of 2 km×2 km horizontal resolution. The area of the watershed was divided up into eight isochrones (Ponce, 1989) of thirty minutes each. Thus, aerial precipitation averages were obtained by simple average of all corresponding 2 km×2 km rainfall accumulations. This was done to reduce the number of nodes in the ANN input-layer so to improve the convergence rate of the ANN training.

Stage level data were measured every ten minutes at the outlet of the watershed (Fig. 1). Since the river channel has a regular cross-section, an empirical rating curve was used to estimate the streamflow. Time series of both radar derived rainfall and streamflow estimates were taken from the data sets available between 1991 and 1995. All selected flood events are plotted in Fig. 5 one after the other side-by-side.



	57	58	59	60	61	62	63	64	65
87	1	1							
88	1	1	2						
89	2	2	2	3					
90	2	3	3	3	4				
91	3	3	3	4	4	5			
92	4	4	4	4	5	5	5	6	6
93	4	4	5	5	5	5	6	6	7
94	5	5	5	5	6	6	6	7	7
95		6	6	6	6	6	7	7	8
96		6	6	7	7	7	7	8	8
97		7	7	7	7	8	8	8	
98		7	8	8	8	8	8		
99					8				

Fig. 4. Radar based coordinate system over the area of the Tamanduateí watershed shown in Fig. 1. The squares represent 2 km×2 km grid cells divided up into eight isochrones.

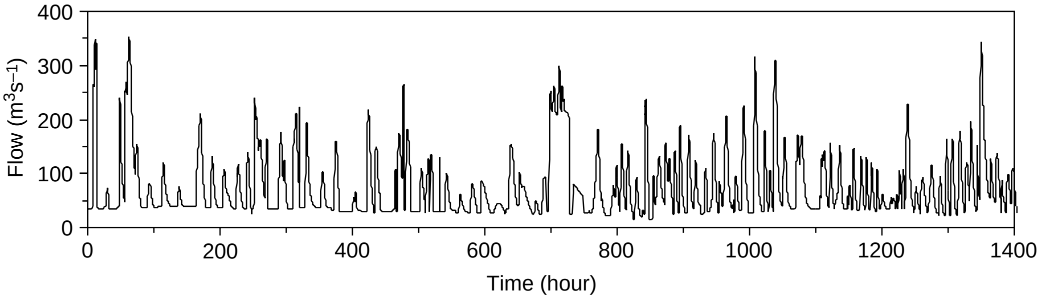


Fig. 5. Time series of streamflow measurements at the outlet of the Tamanduateí watershed (Fig. 3). A total of one-hundred flood events were selected from 1991 to 1995.

The feed forward ANN was trained with the LLSSIM method for different input data (Table 1):

1. Rainfall for each isochrone at time steps t to $t-7\Delta t$ and streamflow as the output at time step t.
2. Similar to (1) for streamflow at time step $t-\Delta t$ as an input.
3. Similar to (1) for stage level.
4. Similar to (2) for stage level.

Initially, a data quality control was performed with the ANN to identified flood events with poor quality due to precipitation and streamflow measurement errors. Thus, the ANN was initially training with all available data sets in a total of 100 events (Fig. 5) with the configuration (8,1,1). Most events with very larger discrepancies between observed and estimated streamflow were eliminated. This data quality control procedure reduced the number of flood events by 25%. The selected events were randomly divide up into three groups for training, verification and forecast. The later used the observed rainfall as forecasted (perfect forecast).

2.6. The verification procedure

Observed and simulated streamflow and stage level time series were compared by means of the mean square difference (MSD). Difference instead of absolute error because the observations such streamflow or stage level are not free of errors. The MSD can be written in terms of phase and amplitude differences (Takacs, 1985):

$$\mathrm{MSD} = \mathrm{MSD_a} + \mathrm{MSD_p} \tag{8}$$

where,

$\mathrm{MSD_a}$ amplitude difference between observed and estimated time series;

$\mathrm{MSD_p}$ phase difference between observed and estimated time series.

Table 1
The ANN training options

Training option	Input			Output	
	Rainfall	Streamflow	State level	Streamflow	Stage level
ANN1	Yes	No	–	Yes	–
ANN2	Yes	Yes	–	Yes	–
ANN3	Yes	–	No	–	Yes
ANN4	Yes	–	Yes	–	Yes

Half-hour rainfall accumulations for isochrones 1 to 8 (Fig. 4) for time steps t to $t-7\Delta t$, respectively, are input to the ANN at time step t. Streamflow or stage level input and output to the ANN are given at time steps $t-\Delta t$ e t, respectively.

Mathematically,

$$\mathrm{MSD_a} = [\sigma(Q_\mathrm{o}) - \sigma(Q_\mathrm{e})]^2 + (\langle Q_\mathrm{o}\rangle - \langle Q_\mathrm{e}\rangle)^2, \quad \mathrm{MSD_p} = 2(1-\rho)\sigma(Q_\mathrm{o})\sigma(Q_\mathrm{e}) \tag{9}$$

where,

$Q_{\mathrm{o(e)}}$	observed (estimated streamflow);
σ	standard deviation operator;
$\langle\,\rangle$	average operator;
ρ	correlation coefficient between Q_o and Q_e

Simulated flood waves with the ANN were compared to the ones obtained with a multi-parameter auto-regression (AR) model developed by Ferraz et al. (1999), a more traditional approach applied to watersheds with predominant surface runoff. The calibration of the AR model and its verification were performed with the same data sets utilized with the ANN.

3. Results

3.1. Training

Fig. 6 shows time series of normalized observed and simulated streamflow or stage level for training performed with ANN1(8,1,1), ANN3(8,1,1), ANN1(8,5,1), ANN2(9,1,1) and ANN4(9,1,1), respectively. The results indicate that the training performed without streamflow (Fig. 6a) or stage level (Fig. 6b) data as an input to the ANN tend to overestimate the lower output values while some higher flood wave peaks are underestimated. These amplitude differences between observations and simulations are most likely related to the use of a single radar ZR relationship. In fact, the MP relationship tends to overestimate lower rainfall rates and underestimate higher ones. Moreover, it is more likely to have larger errors in radar rainfall estimation than in stage level or streamflow since the later represents an integrate measurement of the rainfall over the watershed plus other superficial hydrological processes. The increase in the number of nodes in the hidden-layer (e.g. Fig. 6c) slightly improves results, though not always, as discussed later in this section.

The addition of streamflow (Fig. 6d) or stage level (Fig. 6e) at time step $t-\Delta t$ in the input layer improved the ANN performance by more than 40% variance-wise. This addition works as a memory of the watershed since it is the result of all hydrological processes. One might ask whether the radar data is need at all. With the radar data one can perform quantitative short-term rainfall forecast (Pereira Filho et al., 1999). It can be input to the hydrological model so increasing its lead-time. Furthermore, improving radar rainfall estimates was not a goal in this work. Since for these two configurations the ANN well simulated all flood waves, one can infer that the quality of the data sets and the arbitrary subdivision of the watershed into isochrones limited the performance of the ANN. In this sense, notice that the stage level data yielded slightly better results because no additional uncertainty is introduced by the use a rating curve.

Normally, the first guess weights are selected in between specified limits. An appropriate selection can reduce the search time significantly while the opposite can lead to a premature saturation of the ANN. Initially, the weights are chosen randomly close to zero. Sometimes, it might lead to a local minimum. The ANN training process is completed when the solution yields an error less than an arbitrary small value.

3.2. Verification

A larger number of independent flood events depicted in Fig. 5 were utilized to verify the ANN training. Fig. 7 shows the simulated and observed time series of flood waves for the same options used in the training (Fig. 6). An inspection of Fig. 7 indicates that the ANN yielded fairly good flood wave simulations. The effect of errors and uncertainties are similar to the ones in the ANN training. The highest observed flood wave in between 300 and 350 h in Fig. 7a–c was not simulated at all most probably due to radar rainfall underestimation. Again, the inclusion of streamflow (Fig. 7d) or stage level (Fig. 7e) as an input variable greatly improved the simulations.

Table 2 shows variance coefficients of each of the training and verification experiments performed with the ANN shown in Figs. 6 and 7. More than 95% of

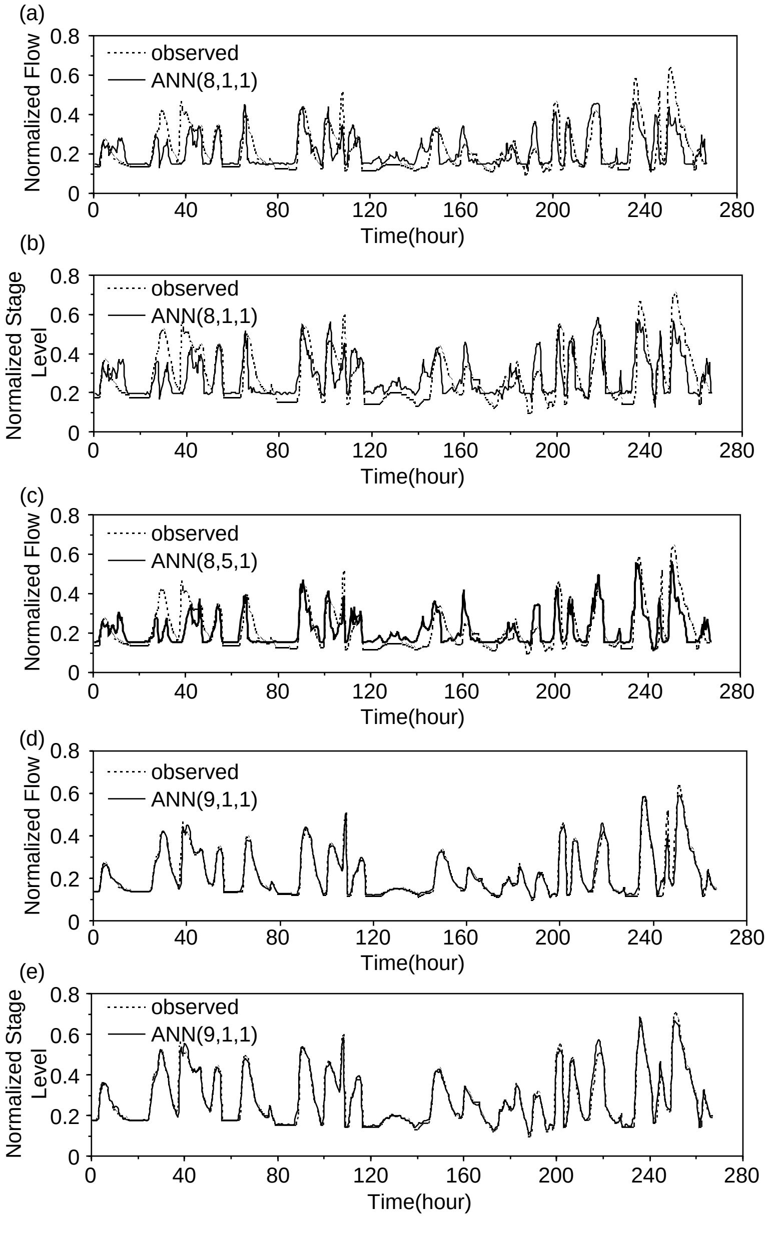


Fig. 6. Times series of normalized streamflow (a) and stage level (b) obtained with the ANN trained with options 1 and 3 (Table 1), respectively, as well as with option 1 for five nodes in the hidden-layer (c). Time series of normalized observations are indicated in each plot by the dashed lines.

the variance is explained with the inclusion of streamflow or stage level at the input-layer. On the other hand, variance coefficients of the ANN verification are 6 to 8% lower than the ones for training without the addition of either streamflow or stage level at the input-layer. As in the training, the use of stage level yields slightly better results than with streamflow.

The AR model (Ferraz et al., 1999) simulations were compared to the ones of the ANN. Results

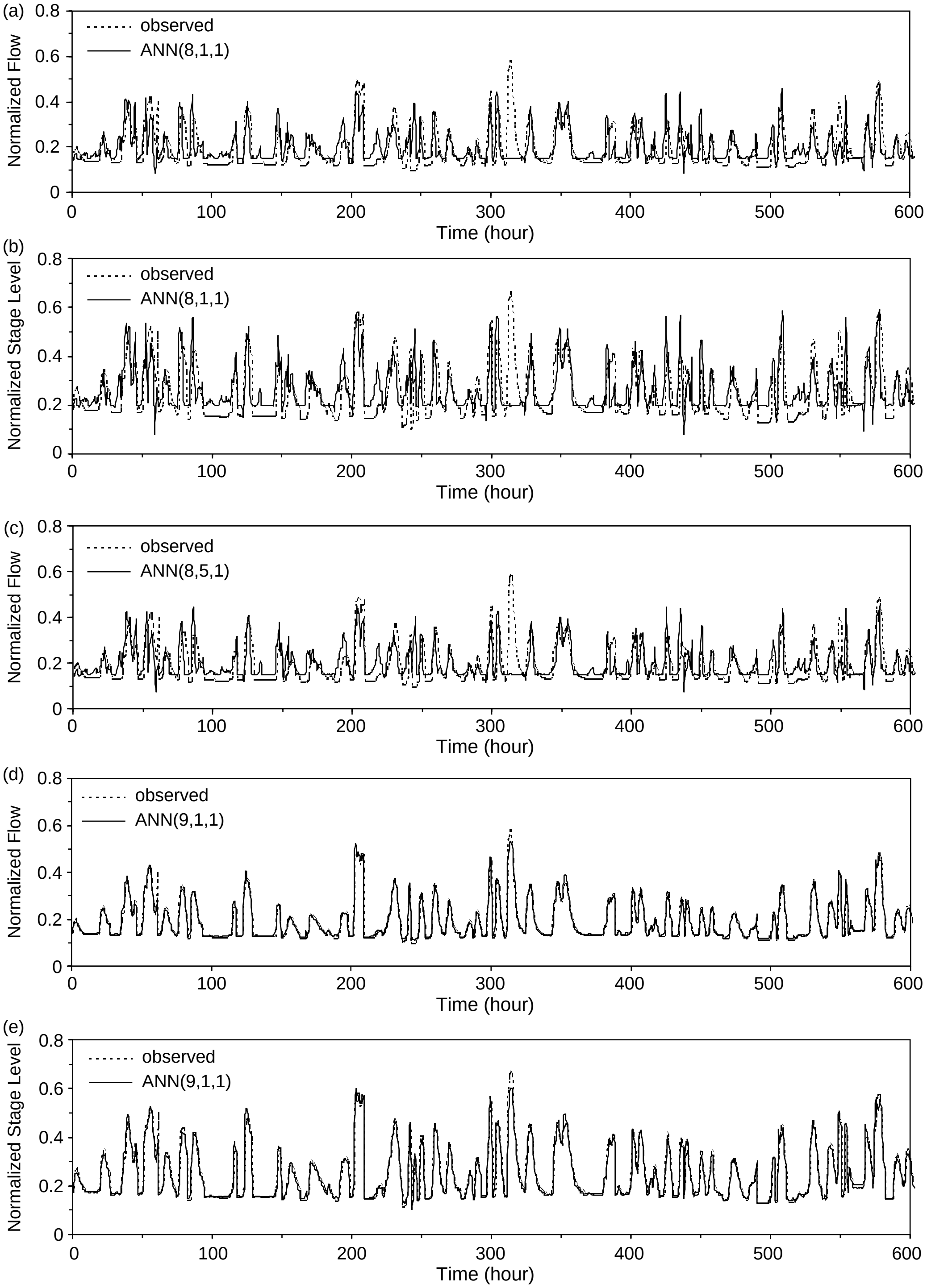


Fig. 7. Similar to Fig. 6, except for the verification data set.

Table 2
Variance coefficients between observed and ANN obtained stage level or streamflow for training (left) an verification (right) options specified in Table 1

Option	r^2 Training	r^2 Verification
ANN1 (8,1,1)	0.53	0.45
ANN3 (8,1,1)	0.54	0.49
ANN1 (8,5,1)	0.57	0.45
ANN2 (9,1,1)	0.95	0.95
ANN4 (9,1,1)	0.96	0.95

The 3D vector indicates from left to right the number of nodes in the input, hidden and output layers.

shown in Fig. 8 indicates better performance for the ANN, though both methods did not adjusted well for some events. Again, most likely due to radar rainfall estimation errors. Both the AR and the ANN overestimate lower streamflows and stage levels when they are not input to the input-layer. Noteworthy, the AR requires a much large number of flood events as well as more parameters for proper calibration. In the experiments with the AR (not shown) the weight given to streamflow or stage level is much higher than the ones given to precipitation.

3.3. MSD, phase and amplitude differences

Table 3 shows phase and amplitude differences for training and verification of the ANN configurations 1 to 4 (Table 1) and the AR model. In general, phase differences are larger than amplitude differences. Option 3 for both the ANN and the AR model yielded the largest phase differences for training and

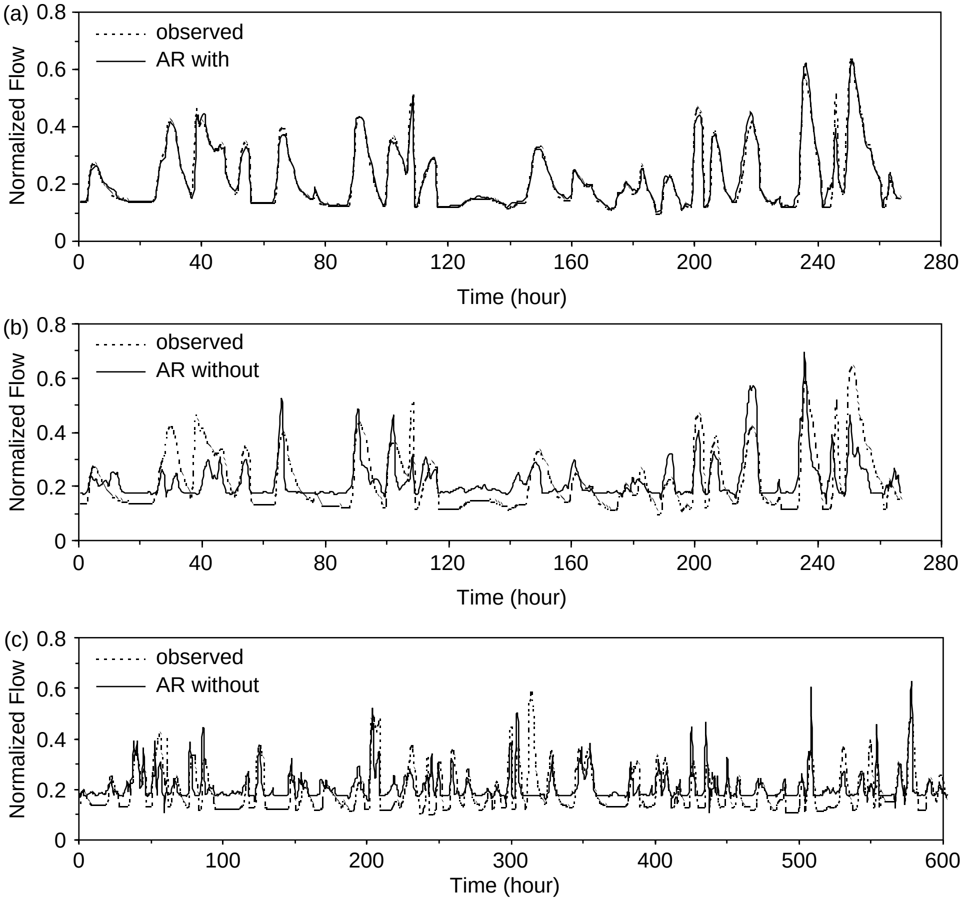


Fig. 8. Time series of normalized streamflow obtained with the AR model calibrated with (a) and without (b) observed streamflow at time step $t-\Delta t$, as well as the verification of the AR model without (c) observed streamflow at time step $t-\Delta t$. It is also indicated the time series of normalized observations in each plot by the dashed lines.

Table 3
Relative MSD phase and amplitude differences for training and verification of the ANN (Table 1) and AR models

Model	Phase		Amplitude	
	Training	Verification	Training	Verification
AR without	1.24×10^{-03}	3.40×10^{-03}	5.28×10^{-03}	1.12×10^{-03}
ANN1 (8,1,1)	4.94×10^{-03}	3.96×10^{-03}	9.17×10^{-04}	2.69×10^{-04}
ANN1 (8,5,1)	4.49×10^{-03}	3.97×10^{-03}	8.01×10^{-04}	2.82×10^{-04}
ANN3 (8,1,1)	5.86×10^{-04}	6.01×10^{-03}	1.14×10^{-03}	4.48×10^{-04}
ARR with	6.26×10^{-03}	3.44×10^{-04}	8.32×10^{-06}	63.5×10^{-06}
ANN2 (9,1,1)	6.26×10^{-04}	3.66×10^{-04}	6.13×10^{-06}	2.77×10^{-06}
ANN4 (9,1,1)	7.68×10^{-04}	6.30×10^{-04}	4.92×10^{-06}	3.86×10^{-05}

Table 4
Relative MSD and variance coefficients between observed and ANN obtained streamflow for the indicated ANN configurations (Table 1)

Configuration	MSD		r^2	
	Training	Verification	Training	Verification
ANN2 (9,1,1)	7.13×10^{-05}	0.033	0.987	0.959
ANN2 (9,2,1)	5.90×10^{-05}	0.046	0.980	0.955
ANN2 (9,5,1)	5.81×10^{-05}	0.031	0.982	0.958
ANN1 (8,1,1)	0.0578	2.602	0.300	0.473
ANN1 (8,5,1)	0.0526	2.408	0.323	0.437
ANN1 (8,10,1)	0.0506	2.252	0.342	0.284
ANN1 (8,20,1)	0.0464	2.221	0.370	0.262

verification, respectively. The inclusion of streamflow or stage level in the input-layer of the ANN or in the AR as predicting variable reduces amplitude differences. Amplitude differences tend to be caused by rainfall errors while phase differences can also be related to changes in the hydrological dynamics of the watershed.

The number of nodes in the hidden-layer was also varied to analyze its impact on the training and verification of the ANN. Table 4 shows MSD and variance coefficients for several configurations. The training improves very little with the increase of nodes. The verification experiments with the ANN without the streamflow at the input-layer tend to augment phase differences as the number of nodes in the hidden-layer increases.

3.4. Flood forecasting

The ANN was utilized to forecast flood waves with the configuration (9,1,1). Observed precipitation was used as forecasted together with stage level data. Since from the previous section it yielded slightly better results. Besides, it is straightforward to estimate flood levels in this manner. Observed stage level and precipitation at the initial time step were input to the ANN to forecast the stage level at the second time step. It was then input to the ANN with the observed precipitation as if it was forecasted to estimate the stage level at the third time step. This procedure was used up to the six time levels or up to 3 h in advance. Thirty-four independent events were utilized in the

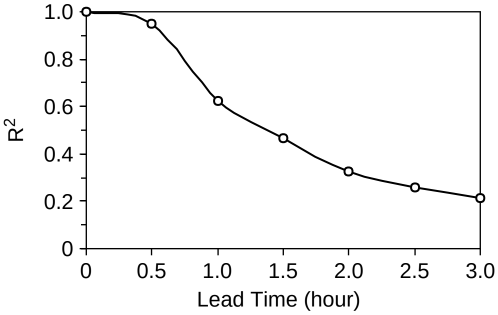


Fig. 9. Covariance coefficients between observed and the ANN4 forecasted stage level for 30, 60, 90, 120, 150 and 180 min lead time. Curve fitted by the least square method.

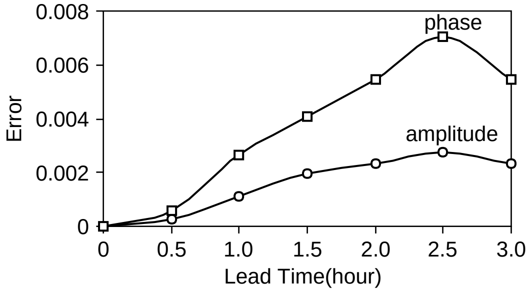


Fig. 10. Phase and amplitude differences (m^2) between observed and forecasted stage levels for 30, 60, 90, 120 and 180 min forecast lead time.

forecast (not shown). Stage level forecasts were compared to measurements.

Covariance coefficients (r^2) between observed and forecasted stage levels at 30-minute time steps are shown in Fig. 9. The covariance decreases with lead-time. Considering that the forecast has skill down to $r^2=0.5$, the ANN forecast is useful almost up to 1.5 h in advance. Of course, the skill is rapidly reduced by errors caused by data errors and hydrological uncertainties. Spatial and temporal variations in rainfall distribution are in general very significant. As mentioned before, radar rainfall estimation is subject to several sources of errors. Unfortunately, no independent rainfall data sets were available to perform a more in depth analysis of errors in radar rainfall estimation. Though, even with these limitations, the ANN performed better than current AR models used in the MASP (Pereira Filho and de Barros, 1998).

The MSD and its phase and amplitude components were estimated for lead-time up to 3 h in advance (Fig. 10). Phase differences are twice as larger as amplitude differences for all lead times. Again, uncertainties in rainfall estimation tend to affect phase differences. But, the averaging of rainfall into isochrones might cause this behavior.

4. Conclusions

An three-layer feed forward ANN was applied to model a urban watershed using weather radar and telemetric streamflow data. Results indicate its usefulness for hydrologic simulation and forecast without explicitly resolving important surface hydrology processes. Even with better spatial and temporal data resolution the performance of the ANN was limited by data uncertainties and errors.

As in the case of AR models, the inclusion of the memory of the watershed greatly improved results. Its inclusion in the input layer improves results by 40%. But, since it carries a much greater weight, even the ANN has its limitations for not resolving many of the hydrological processes that produce the streamflow at the outlet.

The MDS differences between observed and forecasted stage levels grow with increasing lead time and are caused by data errors since observed radar rainfall estimations were used as predicted and the ANN performed well in its verification. Furthermore, the radar data was used without any correction for potential errors. In addition, discrepancies could also be attributed to the ANN model itself. Stage level tends to yield better results. It seems reasonable because no errors are introduced by the theoretical rating curve. Therefore, the hydrological modeling by means of an ANN is a good alternative to flash flood forecasting in the MASP. Seemingly, data quality issues rather than hydrological modeling ones limited the performance of the ANN.

Some of the advantages of the ANN in hydrological applications are: (1) can handle processes poorly define or not well understood; (2) previous knowledge of processes in the training not needed; (3) no specific solution are available; (4) weights are readily adjusted as new data are available; (5) small errors are not amplified since the processing is distributed; (6) relevant information is stored so no need to keep previous processed data records; and (7) no need for additional input and output data except those specified in the training. The ANN can be effectively applied to model nonlinear systems such as the transformation of rainfall into runoff without explicitly specifying hydrological process of the watershed, but it is not a substitute for conceptual models, based on physical process. It can be a feasible alternative for hydrological forecasting when the runoff is predicted from input and output time series.

The ANN is a black box model though procedures can be created to explicitly determine its performance in some instances. Moreover, the ANN training process can be a limiting factor. In some applications, millions of interactions are need before the error is reduced to an acceptable level, especially so in a serial CPU that obtain the functions for each processing unit and respective connections separately. It might be a problem for a large ANN or for application with a large number of data sets. Furthermore, it is difficult to define an ideal ANN structure such that it is as complex as needed yet simple to reduce the time of training. There are no clear rules to determine the ideal size of the ANN structure such as the number of processing units per layer and the number of connections between units. In general, the training starts with just one hidden-layer. If necessary, more hidden-layers can be used. Finally, the number o processing units is proportional to the size of the data set.

Acknowledgements

We would like to thank two anonymous reviewers for their thorough review of this manuscript to enrich it substantially. The authors also wish to thank the Department of Water and Electrical Energy (DAEE) for supplying both radar and stage level data. Also, Dr Kuo-lin Hsu for providing the LLSSIM algorithm and the technical assistance. This work has been partially supported by the National Counsel for Research and Development—CNPq under grant 302419/2002-0.

Appendix A

Mathematically, neuron k can be written by a set o two equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

and

$$y_k = \varphi(u_k + b_k)$$

where, x_1,x_2, …, x_m are input signals; w_1, w_2, …, w_3 are the synaptic weights of neuron k; u_k is the output of the linear combiner due to the input signals; b_k is the bias; $\phi(\cdot)$ is the activation function; and y_k is the output signal from the neuron.

The bias b_k performs a transformation to the output u_k:

$$v_k = u_k + b_k$$

The activation function represented by $\phi(\upsilon)$, defines the neuron output in terms of a induced local field v. The sigmoid function below was used in the present study:

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

where a is a tilting parameter.

The *Linear Least Squares SIMplex*—LLSSIM—is a hybrid training algorithm for three-layer feed forward ANNs (Hsu et al., 1995). The LSSIM is a combination of the Linear Least Squares Method with the Simplex optimization for multi initializations. It obtains the global minimum after few interactions. Furthermore, the algorithm uses a partition of the weight space to implement two training strategies where the weights used in the input-hidden layer are estimated using a multi initialization version of the Simplex nonlinear optimization algorithm (Duan, et al., 1992). The weights of the hidden-output layer are estimated with the linear least squares—LLS method (Scalero and Tepedelenlioglu, 1992). The weight partition reduces the search space and, thus, the training process. The Simplex algorithm improves the search for the global minimum while avoiding local minima (Hsu et al., 1996).

Given m sets of input-output patterns with n_0 input and n_2 output variables and n_1 hidden neurons in a cyclic training (Gupta et al., 1997), the input pattern is represented by $[x_1(p),x_2(p),\ldots,x_{n0}(p)]^{\mathrm{T}}$, the network output by $[z_1(p),z_2(p),\ldots,x_{n2}(p)]^{\mathrm{T}}$, and the target output by $[t_1(p),t_2(p),\ldots,t_{n2}(p)]^{T}$. Then, the cost function to be minimized is:

$$F[w] = \frac{1}{2}\sum_{p=1}^{m}\sum_{k=1}^{n_2}(t_k(p) - z_k(p))^2$$

where,

$$z_k(p) = f\left(\sum_{j=0}^{n_1} w_{kj}^{0} y_j(p)\right)$$

$$y_j(p) = f\left(\sum_{i=0}^{n_0} w_{ji}^h x_i(p)\right)$$

where,

$x_i(p)$ input value of input node i of training pattern p;
x_0 input tendency;
wjih connection weight from input node i to hidden node j;
$y_j(p)$ output value of hidden node j of training pattern p;
wkj0 connection weight from hidden node j to output node k;
$z_k(p)$ output value of output node k of training pattern p;
$t_k(p)$ target value of output node k of training pattern p;
$f()$ monotonic nonlinear transfer function, limited between 0 and 1.

The cost function can be rewritten:

$$F[w] = \frac{1}{2}\sum_{p=1}^{m}\sum_{k=1}^{n_2}(t_k(p) - z_k(p))^2$$
$$= \frac{1}{2}\sum_{p=1}^{m}\sum_{k=1}^{n_2}\left[t_k(p) - f\left(\sum_{j=0}^{n_1}\left(w_{kj}^0 f\sum_{i=0}^{n_0} w_{ji}^h x_i(p)\right)\right)\right]^2$$

and the transfer function $f(\,)$:

$$f(u) = \frac{1}{1+\exp(-u)}$$

The goal of the ANN training is to obtain the weights w_{ji}^{h} and w_{kj}^0 that minimize the cost function. The LLSSIM training strategy splits the weights into two groups, the one associated with the input-hidden layer w_{ji}^{h} and the one with the hidden-output layer w_{kj}^0. Defining $\mathrm{TS}_k(p)$ as the value of the target p transformed backward through the logistic function of output node k:

$$\mathrm{TS}_k(p) = \ln\{t_k(p)/[1 - t_k(p)]\}$$

The above equation is used to define a new error function:

$$F[w] = \frac{1}{2}\sum_{p=1}^{m}\sum_{k=1}^{n_2}(\mathrm{TS}_k(p) - s_k^{\mathrm{o}}(p))^2$$
$$= \frac{1}{2}\sum_{p=1}^{m}\sum_{k=1}^{n_2}\left[\mathrm{TS}_k(p) - \sum_{j=0}^{n_1} w_{kj}^0 f\left(\sum_{i=0}^{n_0} w_{ji}^h x_i(p)\right)\right]^2$$

In this representation, the transformed targets $\mathrm{TS}_k(p)$ are linear in the hidden-output weights, w_{kj}^0, and nonlinear in the input-hidden weights, w_{ji}^{h}. If the values of the input-hidden weights are known, the optimal hidden-output weights can be estimated explicitly using the LLS method:

$$\frac{\partial F_1}{\partial w_{kj}^0} = -\sum_{p=1}^{m}(\mathrm{TS}_k(p) - s_k^0(p))y_j(p) = 0$$

The above equation can be rewritten:

$$\sum_{p=1}^{m} TS_k(p)y_j(p) = \sum_{p=1}^{m}\sum_{l=0}^{n_1} w_{kl}^0 y_l(p)y_j(p)$$
$$= \sum_{p=1}^{m} y_j(p)\sum_{l=o}^{n_1} y_l(p)w_{kl}^0$$

Defining:

$$R_{jl} = \sum_{p=1}^{m}\sum_{l=0}^{n_1} y_j(p)y_l(p)$$

$$Q_j = \sum_{p=1}^{m}\mathrm{TS}_k(p)y_j(p)$$

The hidden-output layer weights, w_k^0, are obtained from a system of linear equations:

$$w_k^0 = R^{-1}Q$$

where, $w_k^0 = [w_{k0}^0, w_{k1}^0, \ldots, w_{kn1}^0]^{\mathrm{T}}$ are conditionally optimal hidden-output weights since their values depend on the values selected for the input-hidden weights.

The ANN model structure is applied for modeling input–output nonlinear systems such as the transformation of rainfall (input) over a watershed (nonlinear system) into runoff (output). In the present work, rainfall and runoff stream flow data sets were normalized because of the use of sigmoid functions. The rainfall time series was normalized between [0,1], while the stream flow time series was normalized between [0.1,0.9] to avoid the saturation of the ANN output (Smith, 1993). The stream flow output $z(t)$ at time t can be represented by a nonlinear model written in terms of inputs of rainfall $x(i-j)$ and stream flow $z(t-j)$. The streamflow output at t, $z(t)$:

$$z(t) = g_{\mathrm{non}}(z(t-1) - \cdots, z(t-n_{\mathrm{a}}), x(t-1), \ldots,\\ x(t-n_{\mathrm{b}})) + e(t)$$

where,

$g_{\mathrm{non}}(\;)$ unknown nonlinear mapping function;
$e(t)$ unknown mapping error to be minimized;
n_{a} and n_{b} number past inputs and outputs contributing to the present output;
t time.

This model is represented by the notation ANN (n_{a}, n_{b}, n_{h}, n_0), where $n_{\mathrm{a}}+n_{\mathrm{b}}$ is the numbers of nodes in the input layer, n_{h} number of nodes in the hidden layer, n_0 number is the number of nodes in the output layer. The ANN is identified by selecting n_{a}, n_{b}, and n_{h} and the ANN weights w_{ji}^{h} e w_{kj}^{0} are estimated such that the prediction error, e, is minimized (Hsu et al., 1995).

References

Crespo, J.L., Mora, E., 1993. Drought estimation with neural networks. Advances in Engineering Software 18, 167–170.

Duan, Q., Sorooshian, S., Gupta, V.K., 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resources Research 28 (4), 1015–1031.

Ferraz, M.I.F., Safadi, T., Lage, G., 1999. Use of series models in forecasting the series of monthly rainfall in the city of Lavras, state of Minas Gerais, Brazil. Revista Brasileira de Agrometeorologia 7 (2), 259–267 (in portuguese).

Fiering, M.B., Jackson, B.B., 1971. Synthetic streamflows. 2$^{\mathrm{nd}}$ ed, American Geophysical Union, Washington (Water Resources Monograph Series).

Franchini, M., Pacciani, M., 1991. Comparative analysis of several conceptual rainfall-runoff models. Journal of Hydrology 122, 161–219.

French, M.N., Krajewski, W.F., Cuykendall, R.R., 1992. Rainfall forecasting in space and time using a neural network. Journal of Hydrology 137, 1–31.

Gupta, V.H., Hsu, K., Sorooshian, S., 1997. Superior training of artificial neural networks using weight-space partitioning. Proceed. IEEE (ICNN'97) 3, 1919–1923.

Haykin, S., 1994. Neural Network A—Comprehensive Foundation. Prentice-Hall International, Inc., New Jersey.

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

Hsu, K.L., Gupta, H.V., Sorooshian, S., 1996. A superior training strategy for three-layer feed forward artificial neural networks. Tucson, University of Arizona. (Technique report, HWR no. 96-030, Department of Hydrology and Water Resources).

Karunanithi, N., Grenney, J.W., Whitley, D., Bovee, K., 1994. Neural networks for river flow prediction. Journal of Computing in Civil Engineering 8 (2), 1994.

Loke, E., 1995. Introduction to Artificial Neural Networks in Urban Hydrology, MATECH, Institute of Environmental Science and Engineering. Technical University of Denmark, Lyngby pp. 1–47, (report).

Luk, K.C., Ball, J.E., Sharma, 2000. A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. Journal of Hydrology 227, 56–65.

Marshall, J.S., Palmer, W.M.K., 1948. The distribution of raindrops with size. Journal of Meteorology 5, 165–166.

Morettin, P.A., 1982. Time series. In Short course of the National Congress of Applied Mathematics and Computational Sciences—SBMAC, 5th, João Pessoa, Brazil (in Portuguese).

Morettin, P.A., Toloi, C.M.C., 1982. Models for the prediction of time series. Vol 1, Rio de Janeiro, IMPA-CNP$_{\mathrm{q}}$ (in Portuguese).

Operational Technical Center (CTO), (1997). Hydraulic and Hydrologic Study of the Tamanduateí watershed. Technical Report (in portuguese).

Osório, F.S., Vieira, R., 1999. Tutorial of systems hybridos inteligentes. In: Cong. Da Soc. Bras. Computação-ENIA'99-Encontro Nacional de Inteligência Artificial, Rio de Janeiro.

Pedrollo, O.C., 1999. Real time hydrological forecasting with effective rainfall estimated from the logistic function. RBRH—Revista Brasileira de Recursos Hídricos 4 (2), 19–30 (in portuguese).

Pereira Filho, A.J., 1999. Radar measurements of tropical summer convection: urban feedback on flash floods. 29th radar conference, AMS, Montreal, Canada, July 1999. Paper 17.4, 939–940.

Pereira Filho, A.J, de Barros, M.T.L., 1998. Flood warning system for Megacities: a Brazilian Perspective. In: Wheater, H., Kirby, C.J. (Eds.), Hydrology in a Changing Environment, vol. III. Wiley, pp. 331–337 (ISBN 0-471-98680-6).

Pereira Filho, A.J., Crawford, K.C., 1999. Mesoscale precipitation fields: part I: statistical analysis and hydrologic response. Journal of Applied Meteorology 38 (1), 82–101.

Pereira Filho, A.J., Crawford, K.C., Hartzell, C., 1998. Improving WSR-88D hourly rainfall estimates. Weather and Forecasting 13 (4), 1016–1023.

Pereira Filho, A.J., Crawford, K.C., Stensrud, D.J., 1999. Mesoscale precipitation fields: part II: hydrometeorologic modeling. Journal of Applied Meteorology 38 (1), 101–125.

Pereira Filho, A.J., Negri, A., Nakayama, P.T., 2003. An intercomparison of gauge, radar and satellite rainfall in the tropics. First Workshop on Precipitation Measurements, IPWG/CGMS/WMO, Madrid, Spain. Proceedings. 275–283.

Ponce, V.M., 1989. Engineering Hydrology—Principles and Practices. Prentice Hall p. 640.

Scalero, R.S., Tepedelenliogiu, N., 1992. A fast new algorithm for training feed forward neural networks. IEEE Trans. Signal Process. 40 (1), 202–210.

Shamseldin, A.Y., O'Connor, K.M., Liang, G.C., 1996. Methods for combining the outputs of different rainfall-runoff models. Journal of Hydrology 197, 203–229.

Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modelling. Journal of Hydrology 199, 272–294.

Smith, M., 1993. Neural Networks for Statistical Modeling. Van Nostrand Reinhold, New York.

Singh, V.P., 1988. Hydrologic Systems-Rainfall-Runoff Modeling, vol. I. Prentice-Hall, Inc., New Jersey.

Takacs, L.L., 1985. A two-step scheme for the advection equation with minimized dissipation and dispersion errors. American Meteorological Society 113, 1050–1065.

Thirumalaiah, K., Deo, M.C., 1998. River stage forecasting using artificial neural networks. Journal of Hydrologic Engineering 3 (1).

Vemuri, V.R., 1994. Artificial neural networks–forecasting time series. Los Alamitos, IEEE Computer Society Press.

Viessman Jr.., W., Lewis, G.L., 1996. Introduction to Hydrology, 4th ed. HarperCollins College Publishers, New York.

Water and Electrical Energy Department (DAEE), 1988. The Tamanduateí Channel Engineering. Water and Electrical Energy Magazine 12, 30–41 (in portuguese).

Xiao, R., Chandrasekar, V., (1995). Multiparameter radar rainfall estimation using neural network techniques. In: Conf. On Radar Meteorology, 27a, Colorado, 199–201.