MONTHLY RAINFALL-RUNOFF MODELING USING ARTIFICIAL NEURAL NETWORKS IN THE CONTEXT OF CLARIS LPB PROJECT

Luiz Fernando Nascimento Ferreira
Universidade Federal do Paraná - lf_ferreira@onda.com.br.
Miriam Rita Moro Mine
Universidade Federal do Paraná – mrmine.dhs@ufpr.br
Heinz Dieter Fill
Universidade Federal do Paraná – heinzfill@yahoo.com
Fernando Weigert Machado
Universidade Federal do Paraná - fwm.civil@gmail.com

Abstract--This paper presents the results of an Artificial Neural Networks - ANN model for the monthly rainfall-runoff transformation in the context of CLARIS LPB Project. One of the objectives of the CLARIS LPB Project is to investigate how global climate changes will modify the guaranteed output of a system of interconnected hydroplants in La Plata Basin. In particular it is proposed to analyze the performance of the hydroplants system within the La Plata basin under a set of future climate scenarios. The methodology should be based on Monte Carlo simulations using synthetic streamflow series representing future scenarios of global climate changes. These series should be obtained from synthetic rainfall series using standard rainfall-runoff models at a monthly time scale. A monthly rainfall-runoff model available is Artificial Neural Networks - ANN for the rainfall-runoff transformation. The case study is being conducted for the interconnected power system South-Southeast of Brazil

Key-words: Artificial Neural Network, climate changes impact, hydropower

1. INTRODUCTION

Transforming rainfall into runoff is a process difficult to formulate due to the large number of variables that are relevant and modify both in space and time. Evaluating this process with accuracy is what allows rational management of the different water uses, such as: supply, irrigation, electric power generation, to forecast extreme flood events and dry periods, to generate scenarios of streamflow from precipitation scenarios resulting from climate change and others. Generally mathematical models known as rainfall-runoff models perform the evaluation of this process.

Rainfall-runoff models are divided into two major groups: conceptual and empirical models. The conceptual models describe mathematically the processes of the hydrologic cycle based on physical laws governing each of these processes. However, despite generally good results are achieved, some aspects of the conceptual models are challenging. Calibration is not easy and, in many cases, depends on field surveys of data often not available. Also the use of basin averages for relevant parameters together with the non linear character of those processes leads to additional difficulties. These characteristics often render the implementation of conceptual model difficult and financially burdensome.

Empirical models are an alternative to the conceptual models. The main characteristic of this type of model consists of establishing a stable relationship between input and output variables without accounting to the physical laws that govern the natural processes when rainfall is transformed into runoff. These models are easy to apply and supposedly cheaper. Examples of these models are multivariable equations with parameters estimated by Artificial Neural Networks - ANNs. This was the method chosen to generate scenarios of monthly average streamflow at CLARIS LPB Project.

One of the objectives of the CLARIS LPB project is to investigate how global climate changes will modify the guaranteed output of a system of interconnected hydropower plants. In particular it is proposed to analyze the performance of the hydropower plants system within the La Plata basin under a set of future climate scenarios. The methodology should be based on Monte Carlo simulations using synthetic streamflow series representing future scenarios of global climate changes. These series should be obtained from synthetic rainfall series using standard rainfall-runoff models at a monthly time scale. A monthly rainfall-runoff model available is Artificial Neural Networks - ANN for the rainfall-runoff transformation. The rainfall series obtained at monthly time scale need to be generated statistically by defining appropriate stochastic processes to represent the rainfall time series (Fill *et al.* 2011). This paper presents the ANN method to estimate the best rainfall-runoff model for the Rio Paranaíba basin in Emborcação station which drains an area of 29,050 km^2.

2. ARTIFICIAL NEURAL NETWORKS - ANN

According (Machado *et al.*, 2011) An ANN is a structure of elements formed by nodes or neurons, similar to the structure of the human brain, mathematically interconnected, representing a function. The coefficients and intercepts of the input variables of this function are called weights and biases.

There are different types of ANN, and the most common is the ANN Multilayer Perceptron - MLP, with the neurons distributed in layers, usually three of them (Galvão *et al*, 1999). From here on, the ANN nomenclature will be adopted to designate an ANN Multilayer Perceptron – MLP. Haykin (1994), Galvão *et al.* (1999) and Fernandes *et al.* (1996) mention that a three-layer ANN can approach any function with non-linear characteristics.

In water resources ANNs have been used to solve several problems: inflow forecasting and reservoir operation (Jain *et al.*, 1999), simulating and optimizing reservoir operation (Neelakantan & Pundarikanthan, 2000), fitting rating curves (Machado *et al.*, 2005) and many others.

One major application of ANN in hydrology has been related to streamflow or rainfall forecasting. Recent contributions of Cigizoglu (2003a, 2003b, 2005a, 2005b) and Jain *et al.* (1999) and Partal and Cigizoglu (2009) deal with this topic. Another application has been the estimation of sediment transport (Cigizoglu and Alp, 2006, Alp and Cigizoglu, 2007, Cigizoglu and Kisi, 2006). A comprehensive review of ANN applications in Hydrology can be found in ASCE Task Committee (2000). The modeling of the rainfall-runoff process by ANN has been used extensively since at least 15 years and is the main subject of this paper.

In order for an ANN to be able to model properly the rainfall-runoff process, it is should undergo a calibration or process called "training" in which its weights and biases are fitted. During the training paired sets of inputs and outputs are presented to the ANN. Based on a specific set of weights and biases the ANN outputs are calculated and compared to the observed output, if the deviation exceeds an allowable value, the weights and biases are corrected and new outputs are computed until the deviation is smaller or equal to the allowable value. This process is controlled by special optimization algorithms called back-propagation, including the descending gradient with momentum and the Levenberg-Maquardt method. During the training process, care should be taken with the ANN architecture, the number of iterations, the initialization of weights and the length of series for training.

Architecture is the way the neurons are distributed among the layers. It is the architecture that defines the functional form of the ANN. ANN architecture has been investigated by: Kadowaki & Andrade (1997), Ballini *et al.* (1997), Campolo *et al.* (1999), Tokar & Johnson (1999), Thirumalaiah & Deo (2000), Lima & Ferreira Filho (2003) and Ramos & Galvão (2001). Tokar & Johnson (1999) performed an evaluation comparing the length of the data series and the number of ANN inputs and suggest that any increase of the number of ANN inputs should be followed by an increase in the lengths of the data series. Lima & Ferreira Filho (2003), studying the semi-arid region of Ceará, Brazil, evaluated different combinations of the number of inputs and the number of neurons in the hidden layers. Twenty-four ANNs were evaluated, each of them trained with three different sets of data. They do not make recommendations about the ideal architecture, but present a good method to map it. Ramos & Galvão (2001) propose a methodology to determine the ANN architecture, based on the initialization of weights, changes in the transfer and training functions with data series of different lengths.

The number of iterations is the number of times the ANN is trained, or the number of iterations of the optimization algorithm to determine weights and biases. Anmala *et al.* (2000) analyzed the influence of the number of iterations, but did not make any special recommendation. It is known that in the case of an excessive number of iterations, the ANNs memorize the sample data and do not generalize the problem proposed. This process is call overfitting.

Initialization represents the set of initial values for the weights and biases at the beginning of training. Ramos & Galvão (2001) comment as a good practice that the initialization of weights and biases should be repeated. The length of the series represents the size of the data series used during ANN training. Different lengths of series were considered by Lima & Ferreira Filho (2003) and Sajikumar & Thandaveswara (1999). The latter applied temporal ANNs in rainfall-runoff modeling, in the Lee river basin, United Kingdom and Thuthapuzha, in Kerala, India and compared the results obtained with the results of other empirical models. The ANNs presented the best results.

3. THE METHOD

Basically an artificial neuron is constituted by three elements: i) set of weights and biases: responsible for ANN learning; ii) sum units computing the linear combination of the inputs; iii) transfer function which a response. The most often used transfer functions are the sigmoid and linear functions (Haykin, 1994). Usually, the values are normalized, which transforms the real data into a scale compatible with the characteristics of the transfer functions.

Although there is other ANN methods in the literature, the ANN employed in this study uses the "Feed Forward Back Propagation" method. In this method, the connection of several neurons is distributed with layers. Within the ANN, the data flows in a single direction, feedforward, i.e., the input data are propagated through the ANN, layer by layer, in the forward direction. The inputs in the input layer are multiplied by the weights of the respective connections. Each neuron in the middle layer receives a linear combination of the input elements. This combination generates a stimulus to the transfer function that emits an output. The responses of the transfer functions are the inputs to the next layer. The input to the output layer is the linear combination of the outputs from the middle layer. The output from the output layer is the ANN response (Haykin, 1994). In mathematical terms, the output from a three layer ANN is represented by equation (1).

$$y_k = \varphi\left(\sum_{j=1}^{q} w_{kj}\varphi\left(\sum_{i=1}^{p} w_{ji}x_i + b_j\right) + b_k\right) \qquad (1)$$

Where: x are the input elements, w are the weights between the connections, b the biases, p the number of neurons in the input layer, q the number of neurons in the middle layer, φ is the transfer function, y is the ANN output and i, j and k are neurons respectively of the input, middle and output layers.

Choosing an ANN to solve a problem consists of solving two sub-problems: the choice of the functional form off $f(x,w)$ and estimating the weights of vector $\boldsymbol{W}$ (Fernandes *et al.*, 1996). From a statistical perspective $f(x,w)$ is a regression function used to fit a vector of inputs $\boldsymbol{X}$ to a vector of outputs $\boldsymbol{D}$. The elements x are the exogenous variables and w the set of parameters. Thus, the function $f(x,w)$ represents a family of curves and the statistical problem is to obtain the optimum estimator $\boldsymbol{W^*}$, which will minimize the sum of square of the residues.

Several specific optimization algorithms for ANN to fit the weights are available in the literature. In this study the o Levenberg-Maqurdt method (Hagan & Menhaj, 1994) was used.

Levenberg-Maquardt Optimization

The Levenberg-Maquart optimization method is an extension of the Newton-Raphson method (Machado, 2005).

$$\Delta w_{jk} = -(\mathrm{H}_\varepsilon\big|_{w_{jk}})^{-1}\,\nabla\varepsilon\big|_{w_{jk}} \qquad (2)$$

The objective function is quadratic in the following form:

$$\varepsilon(n) = \sum_{1}^{n} e_k(n)^2 \qquad (3)$$

Where: $e_k(n)$ is the error between the desired output and the output calculated by ANN in the output layer and n is the lenght of the data series.

For this function Hagan & Menhaj (1994) proposed to compute the corrections of the weights and biases by:

$$\Delta w_{jk} = -[\boldsymbol{J}^T(n)\boldsymbol{J}(n) + \mu I]^{-1}\boldsymbol{J}^T(n)e(n) \qquad (4)$$

Where: J is the matrix of the first derivatives of the error function with respect to the weights called Jacobian matrix, μ is a parameter to equation (4) and should be always positive, and T denotes transpose.

CASE STUDY

The case study was carried out for the river basin Alto Paranaíba (basin 60). The stations considered were: i) Emborcação – streamflow station; ii) Monte Carmelo – rainfall station and iii) Goiânia – climatological station.

The Paraná river runs for 2,570 km to its estuary at the La Plata river, which added to the 1,170 km of the Paranaíba river, its main tributary, gives a total of 3,740 km, being the third longest river in the Americas. Paranaíba River is formed by many tributaries, of which the most northern is the São Bartolomeu, which rises in the Serra dos Pirineus highlands, in the vicinity of Brasília. (Mine *et al.*, 2010). The Figure 1 presents the Paranaíba River Basin with the stations data used in this study.

INPUT DATA FOR THE ANN MODEL

Rainfall data

The rainfall data collection (Monte Carmelo) for the rainfall-runoff simulation in the La Plata basin was based on two data sources: i) the National Operator of the Electric System (*Operador Nacional do Sistema Elétrico - ONS*) and ii) National Water Agency (*Agência Nacional de Águas-ANA*). The period of the study was January 1944 to December 2005 (01/44 to 12/01 obtained from ONS and 01/02 to 12/05 obtained from ANA). We performed a consistency analysis of these data checking the homogeneity and correcting the data (see Figure 2 – accumulated precipitation versus time).

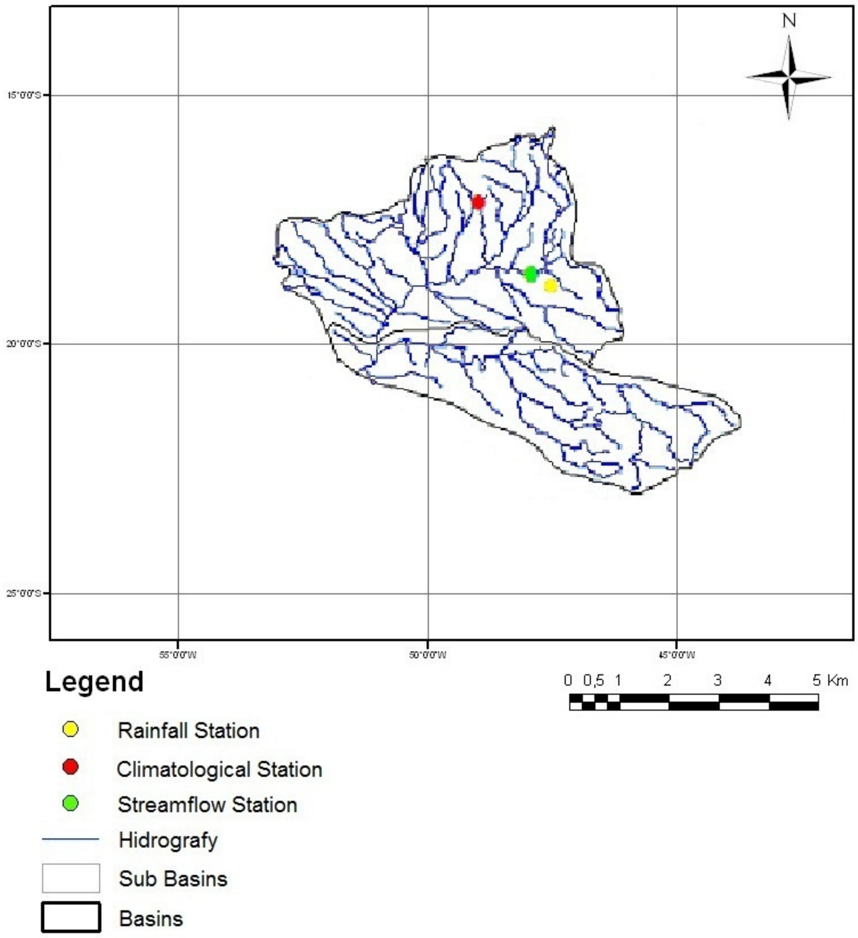
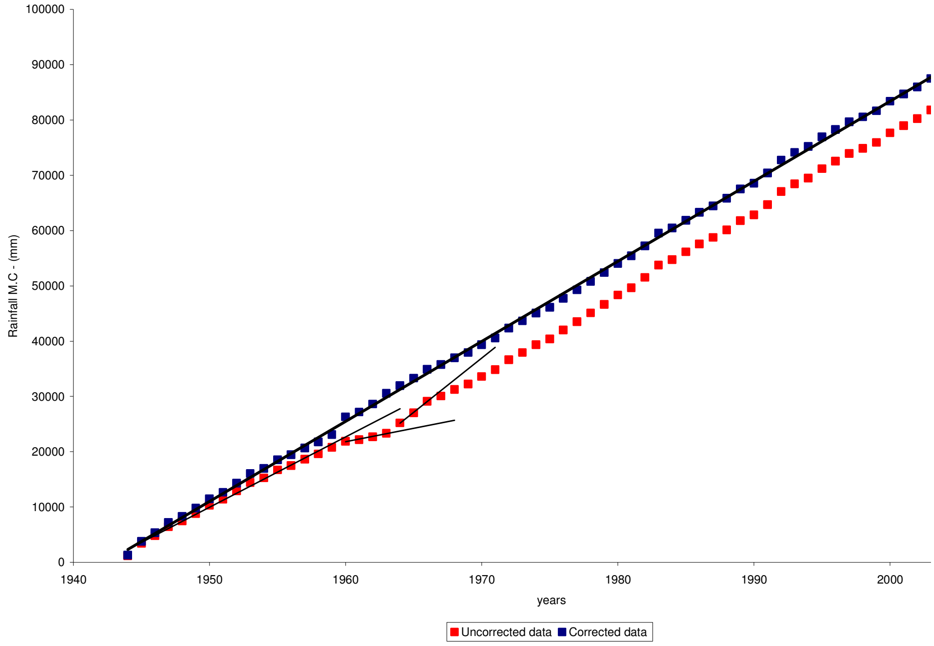


Figure 1 – Location of data stations – precipitation, streamflow and temperature

Streamflow data

The Paranaíba River at Emborcação station drains 29,050.00 km^2. The latitude, longitude and altitude of this station are respectively -18°04'12'', -47°18'07'' and approximately 530 m. Emborcação location for streamflow data are shown on the map of Figure 1. The average monthly streamflows at this station were provided by the National Operator of the System – ONS. According ONS, the monthly streamflows are reconstructed natural flows again.

We performed a consistency analysis of these data that consisted of i) analysis of linear trends; ii) verification of homogeneity through appropriate statistical tests; iii) data correction by cumulative streamflow curves. The monthly average streamflows for Emborcação are consistent dispensing any type of correction.



Rainfall MC – rainfall in Monte Carmelo station

Figure 2 – Correction of the inhomogeneity of Monte Carmelo rainfall station.

Evaporanspiration data

The evapotranspiration data were obtained using the model proposed by Blaney and Criddle (Daker, 1960). This model is based on average monthly temperature (to see Goiânia station in figure 1) and in the percentage of maximum hours of sunshine in the month. It originally was intended to estimate the amount of water needed for the irrigation of some kinds of crops, the so called consumptive use.

Consumptive use was a term first used in the design of the water supply systems in the Western United States. The Blaney-Criddle method is a widely used equation for its estimation. In this case the term "consumptive use" is considered the same as potential evapotranspiration.

The Blaney-Criddle equation contains coefficients accounting for plant type, plant growing season, mean monthly temperature, and seasonal and latitudinal variation in theoretical solar radiation. In metric units the equation may be written as:

$$U_C = K \times P \times (0{,}457 \times T_m + 8{,}13) \tag{5}$$

Where: U_c is the consumptive use (potential evapotranspiration), in millimeters per month; K is a crop use coefficient, dimensionless; P is the percentage of monthly sunshine hours within the year; T_m is the average monthly temperature, in ℃. The values of K and other information the reader finds in Mine *et al.* (2010).

ANN MODEL APPLICATION AND ANALYSIS OF RESULTS

ANN Training

The problem of training an ANN to solve a rainfall-runoff type problem is to fit a suitable function to a data sample. The rainfall-runoff process is non-linear, and the functional form for the fit is unknown. In this case, applying an ANN not only means to fit the best weights and biases to the sample of data observed, but also to investigate, by varying the ANN architecture, which is the best functional form for to observed data.

The ANNs used are of the three-layer MLP type. The input layer does not have transfer functions. All neurons of the middle layer and the output layer have a transfer function of the sigmoid and linear type, respectively.

In order to investigate the best functional form, 24 ANNs were created with variations in the number of inputs and the number of neurons in the middle layers. The output is always runoff. Each combination of inputs was called a model. Table 1 shows the input and output for each of these models. For each model 3, 5, 8 and 10 neurons were used in the middle layer, making up the total of 24 ANNs combinations. Varying both the number of neurons in the middle layer and the number of inputs, allows the evaluation of ANN sensitivity in terms of its architecture.

During the training of the 24 ANNs, the following parameters were considered: length of data series, number of iterations called epochs and initialization of weights, Figure 3 shows schematically the training process with variation of those parameters. Before the training, all inputs were normalized between 0.1 and 0.9 according to Sajikumar & Thandaveswara (1999).

All the ANNs were trained with three different sets of data. From the 221 months available, sets of 60, 120 and 180 items were used for training, and 161,101 and 41 items for validation, respectively. These lengths of the data series were chosen according to Lima & Ferreira Filho (2003).

An excessive number of iterations during training put the calculated values very close to the observed values and does not generalize the process. Because the Levenberg-Maquardt training algorithm converges rapidly, all 24 pre-established ANNs were trained arbitrarily over 30, 60 and 90 iterations.

The weights and biases were initialized at interval [-1,1]. During the training process, the values of the weights and biases change in order to reduce the error. When the initial weights and biases are on a point close to a local minimum, the optimization process will inevitably converge the solution to the local minimum. To avoid this problem, the ANN is initialized six times, at different values.

The combination, during the training, of the ANN architecture, the input sets, the number of initializations and the number of iterations generated a total of 1296 results for analysis. In order to evaluate the influence of all elements proposed in the ANN training, an algorithm was created in MATLAB software, that manipulates the data, trains, simulates, computes the statistics of the results and stores all responses in an output file. The statistics used at this time were the correlation coefficient and the percentage difference of the volumes.

Table 1 Models proposed.

MODEL	INPUTS	OUTPUTS
1	*P (t) EVT (t)*	*Q (t)*
2	*P(t) EVT(t) Q (t-1)*	*Q(t)*
3	*P(t-1) P(t) EVT(t-1) EVT(t)*	*Q(t)*
4	*P(t-1) P(t) EVT(t-1) EVT(t) Q(t-1)*	*Q(t)*
5	*P(t-2) P(t-1) P(t) EVT(t-2) EVT(t-1) EVT(t)*	*Q(t)*
6	*P(t-2) P(t-1) P(t) EVT(t-2) EVT(t-1) EVT(t) Q(t-2) Q(t-1)*	*Q(t)*

P: mean monthly precipitation (mm/month); *EVT*: potential evapotranspiration (mm/month); *Q*: mean monthly discharge (m^3/s).

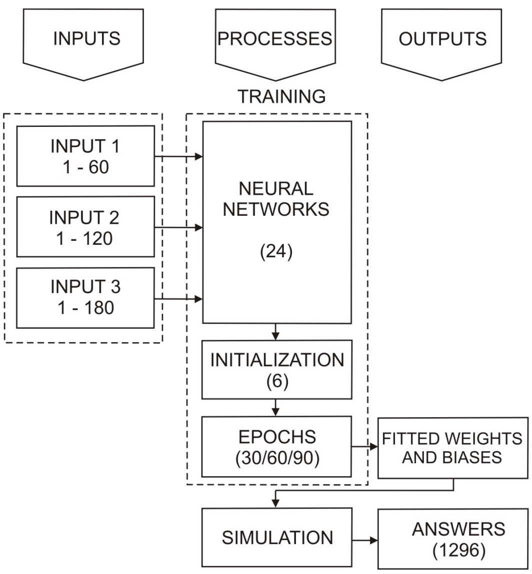


Figure 3 Training method

RESULTS OF ANN

Sensitivity Analysis

Only 108 of the 1296 results were selected for further analysis. They were selected based on the following criterion: for each initialization, model and input set the model with the best performance according to the correlation coefficient in the validation was selected. Each model is represented by four ANNs with 3, 5, 8 and 10 neurons in the middle layer and 30, 60 and 90 inputs during the training. This means 12 possible results for each model. The selection of the results for each model and the best of these have been selected.

Based on the 108 best results, five different analyses were performed:

a) Number of iterations for the trained models achieves the best results.

The frequency of the best result for each number of iteration was calculated. It was observed that 78 or about 72% achieved the best results when they were trained only for 30 iterations (19 or about 18% for 60 iterations, 11 or 10% for 90 iterations).

b) Relationship between the number of iterations and the length of data series.

The number of iterations for the best result was also evaluated considering the input data. It was believed that for the more input data, the number of iterations during training should be greater, Figure 4.

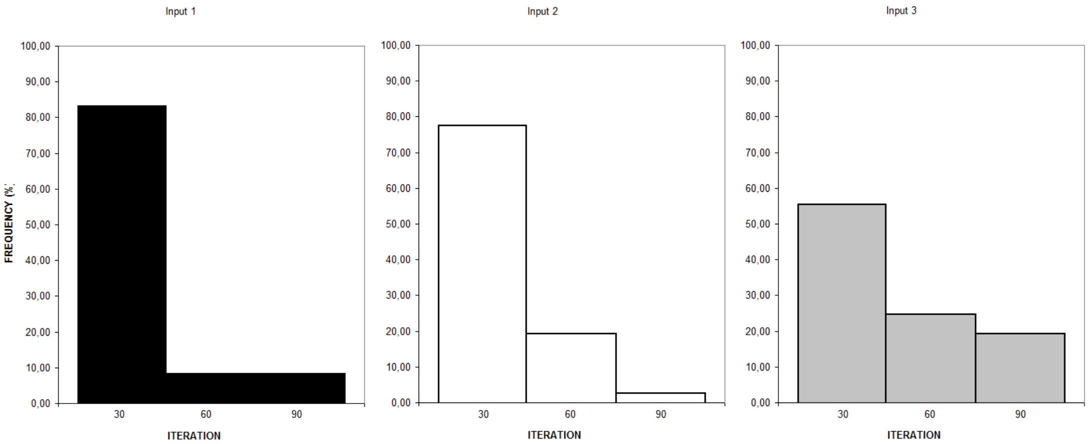


Figure 4 Frequency of the number of iterations

However, Figure 4 shows that the different input sets used in the training did not influence the number of iterations. For all inputs, the best results were achieved by the models trained only 30 times.

c) Evaluation of the relationship between the number of weights and biases of the ANN and the number of iterations.

It was also analyzed if the number of weights and biases, do influence the number of iterations. In order to determine the total number of weights and biases of an ANN, it is enough to know the number of connections, the number of neurons in the middle layer and the number of neurons in the output layer. The simplest ANN in this study has a total number of 13 weights and biases and the most complex has 101 weights and biases.

In order to evaluate the relationship between the number of iterations and the number of weights and biases of the ANN, they were divided into seven classes. For each interval, their frequencies were computed among the 108 best results and are shown in Table 2.

Independently of the number of iterations, the best results were most frequent for ANNs with weights and biases ranging between 31 and 60.

d) Analysis of the influence of reinitializing the weights and biases in the training process.

All 24 ANNs were initialized six times. The first initialization is represented by letter ‘A’ and the last by letter ‘F’. Initialization ‘A’ does not mean a particular set of values, but rather the first initialization for a particular ANN. Two ANNs with initialization ‘A’ of weights and biases, do not present the same values, among other reasons because the ANNs are different and the number of weights and biases is different.

Table 2 Frequency of the best results considering the number of parameters and the number of iterations.

WEIGHTS + BIAS	30	60	90	TOTAL
0-15	0	0	0	0
16-30	16	1	0	17
31-45	23	10	6	39
46-60	17	4	1	22
61-75	14	0	2	16
76-90	6	3	1	10
91-105	2	1	1	4
TOTAL	78	19	11	**108**

Table 3 shows the average, maximum and minimum R^2 values and the Nash-Sutcliffe (NS) efficiency coefficient for these groups of models both for training as for validation.

In order to evaluate the effect of initialization, the best model was chosen for each pair of initialization and input set. The results are shown in Table 5, together with the respective values of R^2 and Nash-Sutcliffe coefficient at validation period.

It may be observed that initialization influences the choice of model and architecture. Also for the same input the results present a certain consistency. For input 1, model 4 with 10 neurons in the middle layer presented the best performance. For input 2, model 2 with 10 neurons in the middle layer presented the best results and for input 3, independent of initialization, model 2 with 8 neurons in the middle layer has showed the best performance. Most of the models in Table 5 were trained over 30 iterations with the number of weights and biases varying from 15 to 57, consistent with the results of items 'a' and 'c' above.

According to Table 4, for all the inputs, model 2 presented in most cases the best result. This model presents only 3 inputs: P (t), EVT (t) and Q (t-1). Once the data available for this basin is not very large, a model with 3 inputs, as noticed, should work better.

Statistical Analysis

Three models were selected for statistical and graphical analysis of the results, one for each input set. For input 1 model 4 and model 6 were the best models. Considering the principle of parsimony, we prefer to choose the model 4. For inputs 2 and 3, model 2 performed best, see Table 4. The best performance for input 1 using model 4 was the one trained on initialization A, shown on Figure 5.

Table 3 – R^2 and Nash-Sutcliffe values

	TRAINING						VALIDATION					
EPOCH	30		60		90		30		60		90	
WEIGHT+BIAS	R^2	NS	R^2	NS	R^2	NS	R^2	NS	R^2	NS	R^2	NS
						AVERAGE						
0-15												
16-30	0.86	0.74	0.72	0.53			0.80	0.62	0.64	0.35		
31-45	0.82	0.69	0.77	0.60	0.82	0.69	0.74	0.53	0.72	0.51	0.79	0.63
46-60	0.88	0.78	0.86	0.73	0.94	0.88	0.81	0.64	0.79	0.59	0.84	0.70
61-75	0.86	0.74			0.80	0.64	0.77	0.56			0.71	0.44
76-90	0.90	0.81	0.78	0.78	0,88	0,77	0.81	0.64	0.79	0.58	0.76	0.55
91-105	0.91	0.83	0.81	0.81	0.87	0.75	0.83	0.68	0.86	0.71	0.79	0.61
						MAXIMUM						
0-15												
16-30	0.91	0.83	0.72	0.53			0.87	0.75	0.64	0.35		
31-45	0.91	0.83	0.91	0.83	0.91	0.83	0.88	0.78	0.88	0.76	0.89	0.78
46-60	0.96	0.92	0.91	0.83	0.94	0.88	0.87	0.75	0.87	0.76	0.84	0.70
61-75	0.92	0.85			0.82	0.67	0.87	0.76			0.73	0.51
76-90	0.92	0.84	0.93	0.86	0.88	0.77	0.88	0.76	0.84	0.71	0,76	0.55
91-105	0.92	0.84	0.90	0.81	0.87	0.75	0.83	0.68	0.86	0.71	0.79	0.61
						MINIMUM						
0-15												
16-30	0.69	0.47	0.72	0.53			0.62	0.32	0.64	0.35		
31-45	0.66	0.44	0.62	0.38	0.68	0.46	0.45	0.18	0.59	0.33	0.62	0.34
46-60	0.77	0.52	0.77	0.50	0.94	0.88	0.69	0.36	0.69	0.38	0.84	0.70
61-75	0.65	0.42			0.78	0.60	0.51	0.23			0.70	0.37
76-90	0.88	0.77	0.86	0.73	0.88	0.77	0.75	0.52	0.76	0.50	0.76	0.55
91-105	0.91	0.82	0.90	0.81	0.87	0.75	0.82	0.67	0.86	0.71	0.79	0.61

Table 4 Influence of initialization on the determination of the best model for each input set.

INIC.	INPUT 1				INPUT 2				INPUT 3			
	MOD	ARC	R^2	NS	MOD	ARC	R^2	NS	MOD	ARC	R^2	NS
A	4	10	0.87	0.76	2	10	0.87	0.75	2	8	0.88	0.76
B	4	3	0.84	0.71	2	3	0.85	0.73	2	8	0.89	0.78
C	4	8	0.84	0.71	4	5	0.86	0.73	2	8	0.89	0.78
D	4	8	0.84	0.70	2	10	0.87	0.76	2	8	0.87	0.75
E	6	8	0.87	0.76	2	3	0.87	0.75	2	8	0.88	0.78
F	4	10	0.84	0.71	2	3	0.87	0.75	2	8	0.89	0.78

MOD: model; ARC: Architecture of the middle layer; NS: Nash-Sutcliffe

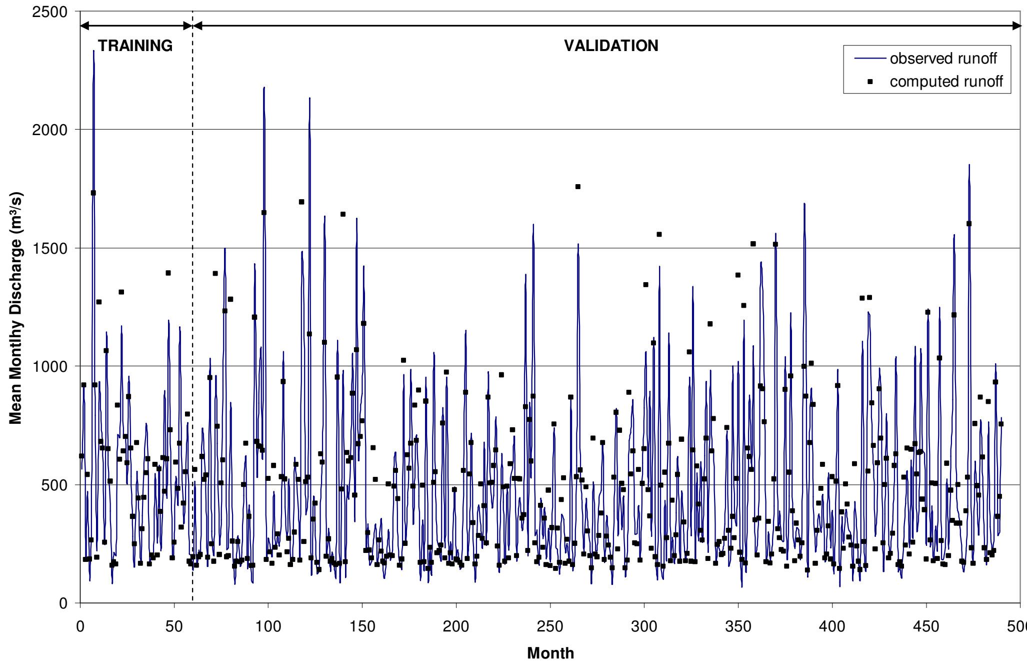


Figure 5 - Result for input 1 – ANN (model 4 – initialization A)

From Figure 5 it can be observed that during training model 4 showed a relatively good representation of low runoff, but some dispersion occurred from medium to high runoff values (to see Figure 6). The tendency was to minimize these flows. A particular feature of this model is that despite the ANN was trained using predominantly low runoffs, during validation it did very well in forecasting higher runoffs, although it preserved the tendency to minimize the latter. The difference of volume in training was 0.78% and in validation, 2.89%. The R^2 value and the Nash-Sutcliffe coefficient during validation were 0.87 and 0.76 respectively. During training the values were 0.91 and 0.83 respectively.

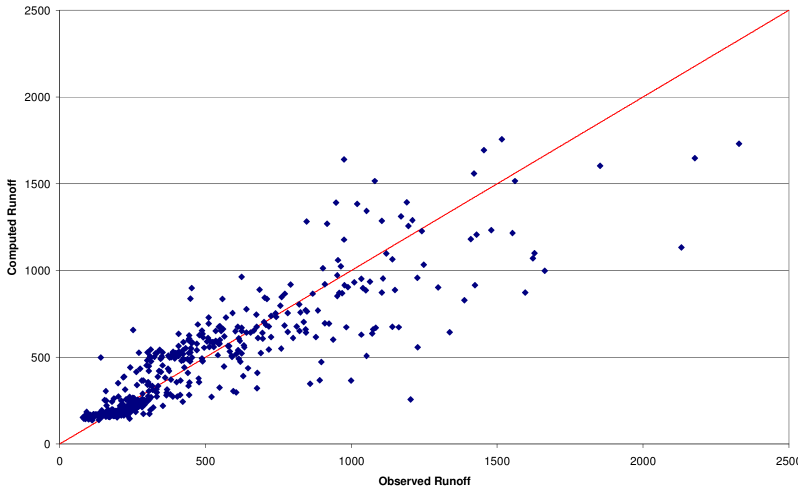


Figure 6 - Comparison between observed and calculated runoffs by model 4 – input 1– initialization A

For input 2, model 2, trained in ‘D’ initialization, presented the best result shown in Figure 7. For this case model 2 had a similar performance to model 4 (input 1) in terms of coefficients R^2 and the Nash-Sutcliffe. These coefficients were 0.91 and 0.82 respectively.The difference of volume in training was 0,15% and in validation was 1,38%. For this case, during training, both the low and high runoffs were slightly better represented than in the previous model. During the validation R^2 and the Nash-Sutcliffe coefficient were 0.87 and 0.76 respectively. Also the model presented a strong tendency to reduce the highest runoffs (Figure 8).

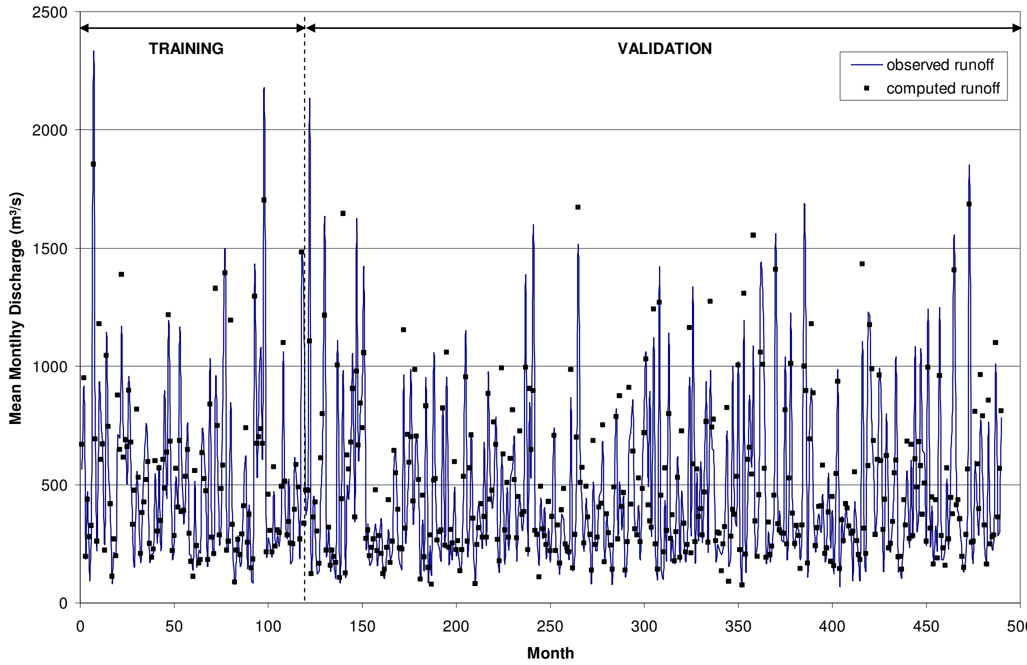


Figure 7 Best result for input 2 – ANN (Model 2 – initialization D)

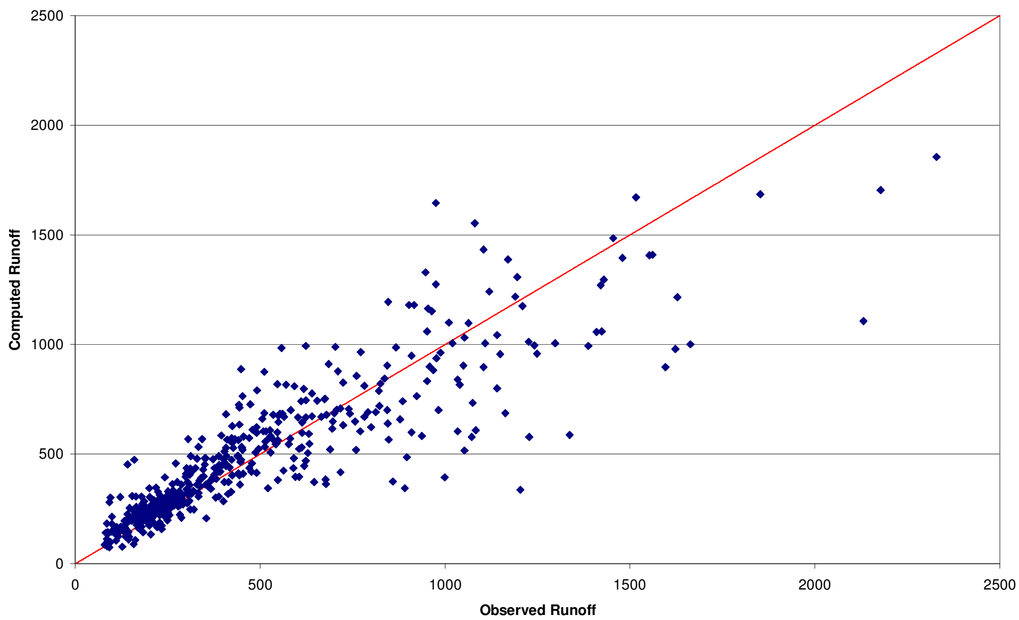


Figure 8- Comparison between observed and calculated runoffs by model 2 – input 2– inicialization D.

The best result based on R^2 value for input 3 was obtained with model 2 and initializations ‘B’, ‘C’ or “F’. The performance for this case is shown on Figures 9 and 10. The coefficients R^2 and Nash-Sutcliffe on validation were 0.89 and 0.78 respectively in all cases.

Among all models, this one based on R^2 and Nash-Sutcliffe coefficients was chosen as the model to represent the relationship between rainfall and monthly runoff through the ANNs. The model performed reasonably well, both for low and high runoffs. The R^2 values in training and validation were respectively equal to 0.89 and 0.89. The Nash-Sutcliffe coefficients were 0.79 and 0.78 during training and validation respectively. The difference in volume was 0,07% in training and 1,25% in validation.

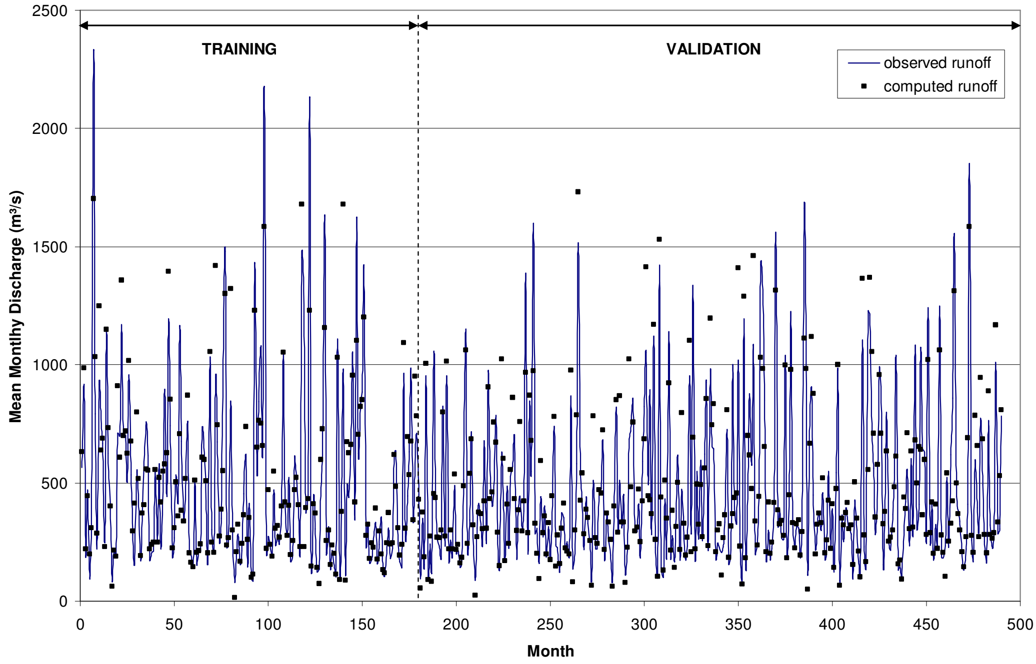


Figure 9 Best result for input 3 – ANN (Model 2 – initialization B, C or F)

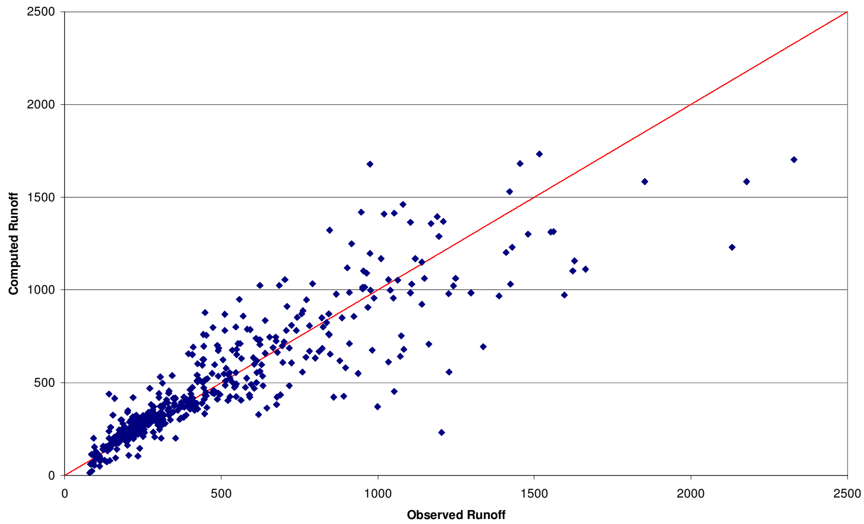


Figure 10- Comparison between observed and calculated runoffs by model 2, input – 3, inicialização B, C or F

CONCLUSION

The best result based on R^2 and Nash-Sutcliffe coefficients for input 3 was obtained with model 2 and initialization 'B', 'C' and 'F' that present the same result. Among all models, the model 2, input 3 and initialization B, based on R^2 values both for training and validation was chosen as the model to represent the relationship between rainfall and monthly runoff through the ANNs to the Emborcação station in Paranaíba basin.

ACKNOWLEDGEMENTS

The authors have received financial support for preparing this paper from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 212492 (CLARIS LPB - A Network Europe-South America for the Assessment of Climate Change and Impact on the La Plata basin). The second author thanks the *Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNP*q by the research scholarship.

REFERENCES

ALP, M., AND CIGIZOGLU, H.K., (2007) “Suspended sediment estimation by feed forward back propagation method using hydro meteorological data”, *Environmental Modelling & Software* 22(1) 2-13.

ANMALA, J., ZHANG, B. & GOVINDARAJU, R. S. (2000) Comparison of ANNs and empirical approaches for predicting watershed runoff. *J. Water Resour. Planning and Management*, Reston: ASCE, vol. 126. n.3, p. 156-166, May/June.

ASCE Task Committee, (2000) Artificial neural networks in hydrology II. *Journal of Hydrologic Engineering*, ASCE, 5(2), 124-132.

BALLINI, R., FRANCA, E., KADOWAKI, M., SOARES, S. & ANDRADE, M. (1997) Modelos de redes neurais e box e jenkins para previsão de vazões médias mensais. *XII Simpósio Brasileiro de Recursos Hídricos*. Goiana. Relação de trabalhos. Goiana: ABRH, 1997. 1CD ROOM.

CIGIZOGLU, H.K., (2003a)Incorporation of ARMA models into flow forecasting by artificial neural networks, *Environmetrics*, 14, 4, 417-427.

CIGIZOGLU, H.K., (2003b). Estimation, forecasting and extrapolation of flow data by artificial neural networks. *Hydrological Sciences Journal*, 48(3), 349-361.

CIGIZOGLU, H.K. (2005a) “Application of the Generalized Regression Neural Networks to Intermittent Flow Forecasting and Estimation”, ASCE *Journal of Hydrologic Engineering* 10(4), 336-341.

CIGIZOGLU, H.K.(2005b) “Generalized Regression Neural Network in Monthly Flow Forecasting”, Civil Engineering and Environmental Systems, 22 (2), 71-84

CIGIZOGLU, H.K., AND ALP, M.,(2006) “Generalized Regression Neural Network in modelling river sediment yield”, *Advances in Engineering Software*, 37(2), 63-68.

CIGIZOGLU, H.K. AND KISI, O., 2006. Methods to improve the neural network performance in suspended sediment estimation. *Journal of Hydrology*, 317 (3-4), 221-238.

CAMPOLO, M., ANDREUSSI, P. & SOLDATI, A. (1999) River flood forecasting with a neural network model. *Water Resour. Res.*, Washington: AGU, vol.35, n. 34, p. 1191-1198, Apr.

DAKER, A. *A água na agricultura: manual de hidráulica agrícola*. 3. Ed. Rio de janeiro. F. Bastos, 1960. V. 3 p. 42-9.

FERNANDES, L. G. L., NAVAUX, P. O. A. & PORTUGUAL, M. S. (1996) Previsão de Séries de Tempo: Redes neurais artificiais e modelos estruturais. *Pesquisa e Planejamento Econômico,* PPE: vol. 26, n. 2, p. 253-276, Ago.

FILL, H. D., MINE, M. R. M.; SANTANA, A. F. Monthly precipitation model based in hydrological scenarios. In: International Symposium on Climate Change and Water - ISCCW, 2011, Nanjing. Proceedings of the International Symposium on Climate Change and Water - ISCCW. Nanjing : NHRI, 2011. v. 1. p. 1-10.

GALVÃO, C O., VALENCA, M. J. S., VIEIRA, V. P. P. B., DINIZ, L. S., LACERDA, E. G. M. DE, CARVALHO, A. C. P. L. F. & LUDERMIR, T. B. (1999) *Sistemas Inteligentes: aplicações a recursos hídricos e ciências ambientais.* Porto Alegre: Ed. Universidade/UFRGS/ABRH.

HAGAN, M. T.& MENHAJ, M.B. (1994) Training feedforward networks with the Marquardt algorithm. Neural Networks, *IEEE Transactions on*. vol. 5, n 6, p. 989-993, Nov.

HAYKIN, S. (1994) *Neural Networks – A Comprehensive Foundation.* New York: Macmillan College Publishing Company.

JAIN, S. K., DAS, A. & SRIVASTAVA, D. K. (1999) Application of ANN for reservoir inflow prediction and operation. *Journal of Water Resour. Planning and Managemente*, Reston: ASCE, vol. 125, n.5, p.263-271, Sep./Oct.

KADOWAKI, M. & ANDRADE, M.G. (1997) Previsão de vazões mensais utilizando redes neurais multicamadas com algortimo backpropagation. *Simpósio Brasileiro de Redes Neurais*, Goiana.

LIMA, C.H.R. & FERREIRA FILHO, W. M. (2003) Análise de modelos de redes neurais aplicados ao processo chuva-deflúvio no semi-árido. *XV Simpósio Brasileiro de Recursos Hídricos.* Curitiba. Relação de trabalhos. Curitiba: ABRH, 2003. CD ROOM.

MACHADO, F. W. (2005) Modelagem chuva-vazão mensal utilizando redes neurais artificiais. MSc Thesis, Universidade Federal do Paraná, Curitiba, Paraná, Brasil.

MACHADO, F.W., SANTOS, I., PERREIRA FILHO, D. L. B. & MINE, M. R. M. (2005) Avaliação do ajuste e extrapolação de curvas de descarga através de redes neurais. *XX Congreso Nacional del Agua – CONAGUA e III Simposio de Recursos Hidricos del Cono Sur*. Mendonza, Mai.

MACHADO, F. W., MINE, M. R. M. ; KAVISKI, E ; FILL, H. D. (2011). Monthly rainfall-runoff modeling using artificial neural networks. *Hydrological Sciences Journal*, Publisher Taylor & Francis, London 56: 3, 349 — 361.

MINE, M. R. M.; FILL, H. F.; BESSA, M. R.; FERNANDES, Cristovão V S; WEIGERT, F. W., NASCIMENTO, L.F.F. *WATER RESOURCES IN LA PLATA BASIN IN THE CONTEXT OF CLIMATE CHANGE: Impact of the climate changes in hydropower.* CURITIBA: UFPR, 2010 (Partial Report 04).

NEELAKANTAN, T. R. & PUNDARIKANTHAN, N. V. (2000) Neural network-based simulation-optimization model for reservoir operation. *J. Water Resour. Planning and Management*, Reston: ASCE, vol. 126, n.2, p. 57-64, Mar/Apr.

PARTAL, T., CIGIZOGLU, H.K., (2009) Prediction of daily precipitation using wavelet-neural networks, *Hydrological Sciences Journal*, 54(2), 234-246.

RAMOS, A. M. & GALVÃO, C. O. (2001) Estabilidade de redes neurais em modelagem hidrometeorológica. *XIV Simpósio Brasileiro de Recursos Hídricos*, Aracajú. Relação de trabalhos. Aracajú: ABRH, CD ROOM.

SAJIKUMAR, N. & THANDAVESWAR, B.S. (1999) A non-linear rainfall-runoff model using artificial neural network. *J. Hydrol.*, London: Elsevier, vol. 216, n. 1-2, p. 32-55, Mar.

THIRUMALAIAH, K. & Deo, M. (2000) Hydrological forecasting using neural networks. *J. Hydrol. Enging.*, ASCE, Reston, vol.5, n.2, p.180-189, Apr.

TOKAR, A. S. & JOHNSON, P. A. (1999). Rainfall-runoff modeling using artificial neural networks. *J. Hydrol. Enging.*, ASCE, Reston, vol.4, n.3, p.232-239, Jul.