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 hydropower plants in La Plata Basin. In particular it is proposed to analyze the performance of the hydropower 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 Paranaiba basin in Emborcação station which drains an area of 29,050 km².
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).

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

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, \( \varphi \) 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 \( f(x, w) \) and estimating the weights of vector \( W \) (Fernandes et al., 1996). From a statistical perspective \( f(x, w) \) is a regression function used to fit a vector of inputs \( X \) to a vector of outputs \( D \). 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 \( W^* \), 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 Levenberg-Maquardt method (Hagan & Menhaj, 1994) was used.

**Levenberg-Maquardt Optimization**

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

\[ \Delta w_k = -(H_1 \left|_{w_k} \right) \nabla \varepsilon_{w_k} \]

The objective function is quadratic in the following form:

\[ \varepsilon(n) = \sum_{i} e_i(n)^2 \]

Where: \( e_i(n) \) is the error between the desired output and the output calculated by ANN in the output layer and \( n \) is the length of the data series.

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

\[ \Delta w_k = -[J^T(n)J(n) + \mu I]^{-1} J^T(n)e(n) \]

Where: \( J \) is the matrix of the first derivatives of the error function with respect to the weights called Jacobian matrix, \( \mu \) 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 Paranaiba river, its main tributary, gives a total of 3,740 km, being the third longest river in the Americas. Paranaiba 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](image)

Streamflow data

The Paranaiba River at Emborcação station drains 29,050.00 km². 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.
Evaporanspiration data

The evaporanspiration 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:

\[
U_c = K \times P \times (0.457 \times T_m + 8.13)
\]

Where: \( U_c \) is the consumptive use (potential evaporanspiration), 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 °C. 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 the 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.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$P(t)$ $EVT(t)$</td>
<td>$Q(t)$</td>
</tr>
<tr>
<td>2</td>
<td>$P(t)$ $EVT(t)$ $Q(t-1)$</td>
<td>$Q(t)$</td>
</tr>
<tr>
<td>3</td>
<td>$P(t-1)$ $P(t)$ $EVT(t-1)$ $EVT(t)$</td>
<td>$Q(t)$</td>
</tr>
<tr>
<td>4</td>
<td>$P(t-1)$ $P(t)$ $EVT(t-1)$ $EVT(t)$</td>
<td>$Q(t)$</td>
</tr>
<tr>
<td>5</td>
<td>$P(t-2)$ $P(t-1)$ $P(t)$ $EVT(t-2)$ $EVT(t-1)$ $EVT(t)$</td>
<td>$Q(t)$</td>
</tr>
<tr>
<td>6</td>
<td>$P(t-2)$ $P(t-1)$ $P(t)$ $EVT(t-2)$ $EVT(t-1)$ $EVT(t)$</td>
<td>$Q(t)$</td>
</tr>
</tbody>
</table>

$P$: mean monthly precipitation (mm/month); $EVT$: potential evapotranspiration (mm/month); $Q$: mean monthly discharge (m³/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.

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.
Three models were selected for statistical and graphical analysis of the results, one for each input set. For Statistical Analysis model with 3 inputs, as noticed, should work better.

Once the data available for this basin is not very large, a model with 3 inputs: P (t), EVT (t) and Q (t-1). For input 2, model 2 with 10 neurons in the middle layer presented 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.

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

<table>
<thead>
<tr>
<th>WEIGHTS + BIAS</th>
<th>30</th>
<th>60</th>
<th>90</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16-30</td>
<td>16</td>
<td>1</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>31-45</td>
<td>23</td>
<td>10</td>
<td>6</td>
<td>39</td>
</tr>
<tr>
<td>46-60</td>
<td>17</td>
<td>4</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>61-75</td>
<td>14</td>
<td>0</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>76-90</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>91-105</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>TOTAL</td>
<td>78</td>
<td>19</td>
<td>11</td>
<td>108</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>INIC</th>
<th>INPUT 1</th>
<th>INPUT 2</th>
<th>INPUT 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MOD</td>
<td>ARC</td>
<td>R²</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>10</td>
<td>0.87</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>3</td>
<td>0.84</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>8</td>
<td>0.84</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>8</td>
<td>0.84</td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td>8</td>
<td>0.87</td>
</tr>
<tr>
<td>F</td>
<td>4</td>
<td>10</td>
<td>0.84</td>
</tr>
</tbody>
</table>

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

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

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)](image1)

![Figure 8- Comparison between observed and calculated runoffs by model 2 – input 2– inicialization D.](image2)

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.

**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 Emboração station in Paranaiba 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 - CNPq by the research scholarship.

REFERENCES


