Development of an Application to Determine

Quality Indexes in Rivers through Artificial Neural Networks Applied to Data from the Bogotá River

Luis Enrique Méndez López
Universidad Distrital “Francisco José de Caldas”
Bogotá, Colombia

Octavio José Salcedo Parra
Universidad Distrital “Francisco José de Caldas”
Universidad Nacional de Colombia, sede Bogotá
Bogotá, Colombia

Miguel J. Espitia R.
Universidad Distrital “Francisco José de Caldas”
Bogotá, Colombia

Abstract

The quality of water in a river is a factor that must be influenced by the system that surrounds it, in this work we try to determine through a historical set of measurements on the Bogota river between years 2008 to 2015 supplied by the Autonomous Regional Corporation of Cundinamarca (CAR). We want to know the variables with the greatest impact on changes in the water index of the ICA and with them to build a model using artificial neural networks to predict water quality indexes in any type of river in the same Bogota river.
Keywords: Bogotá river, artificial intelligence, multilayer perceptron, artificial neural networks, machine learning, water quality index (ICA)

1 Introduction

The Bogotá river, with a distance of 375 km from its birth on the Guacheneque moor, to its mouth on the Magdalena River is a water source of great importance in the city and country, but historically it has presented a deterioration in its quality of water, making risky some uses of it, such as human consumption, animal feed, agricultural use, strategic irrigation or industrial use. For this reason, authorities such as CAR have carried out monitoring at 81 stations, which take data to determine studies on water quality and contribute to environmental protection plans. The waters of the Bogota River are highly contaminated daily, it is necessary to begin to determine which components affect more and at increasing speeds, through the ICA (Water Quality Index) has a basis to determine quality indices, but the entities in charge of collecting these data give little elaborated results for their treatment in order to obtain new knowledge, for which the data must be treated and transferred to a database, from which real data analysis can be carried out, but yet given the number of real variables that determine water quality an algorithm such as the multilayer perceptron provides the tools to achieve more accurate indexes in less time, in addition to being increasingly accurate and have the advantage of being able to use the model determined in the artificial neural network to calculate the ICA of any river at a high speed and precision.

2 Background

Contrasting seasonal variations occur in river flow and water quality as a result of short-term storms, severe-intensity storms, and typhoons in Taiwan. Sudden changes in river flow caused by imminent extreme events can lead to severe degradation of river water quality and fatal impacts on ecosystems [1]. Urban dust pollution has become an outstanding environmental problem due to rapid urbanization in China. However, it is very difficult to construct an inventory of urban dust, due to its small horizontal scale and its strong temporal / spatial variability [2]. A multilayer Perceptron (MLP) is an automatic learning algorithm capable of sorting large amounts of data and finding patterns in complex data sets. In this article, the MLP algorithm is used to perform intrusion detection based on the Knowledge Discovery and Datamining (KDD) dataset. The results show the minimization of errors during training, as well as the classification accuracy for a number of different perceptron topologies. These topologies differ in their number of hidden layers, and in the number of neurons within each hidden layer [3]. Earthing due to urbanization triggers a series of negative environmental impacts with direct effects on the quality of life of people living in cities. Changes in ecosystem services are associated with land use, among which is the immediate loss of C due to conversion of land use [4]. Anterior Cruciate Ligament (ACL) injury
is an injury to the knee joints that occurs in many athletes. It has a significant impact on the movement of patients in their daily lives and sports activities. Therefore, it is important to detect ACL injury at an early stage. For that, the proper treatment can be operated on time. This article proposes a novel method to look for differences in gait patterns between ACL reconstructed patients and healthy individuals, in addition to classifying the class to which gait data belongs [5]. The paper designed a research platform for medical information analysis and data mining. Based on OpenStack, a cloud management platform was created that can perform virtual distribution and management. It could also improve the utilization of resources and otherwise reduce the requirements for personal PC work. In addition, the application of Hadoop technology provided a better solution for mining and medical data analysis, since a task can be divided into small subtasks [6]. Image restoration is the method of fixing degraded images that can not be retaken, and the process of getting the image back is expensive. Image restoration is performed in two fields: spatial domain and frequency domain. In the spatial domain, the refining action to restore the images is performed by operating directly on the pixels of the digital image. The adequacy of the restoration was verified taking into consideration the peak signal-to-noise ratio (PSNR) and the mean square error (MSE) [7]. Humans constantly modify their environment to better suit their needs. These changes are even more important in the small Mediterranean islands, where the flow and type of ecosystem services (SE) is limited by insularity and heavily exploited by economic activities. We evaluated the SE dynamics from 1954 to 2007 linked to the landscape changes of the Island of Vulcano (southern Italy) and related this transformation to the perception of the local communities. We estimate the changes in the total economic value of SE and link this objective assessment with a survey among the inhabitants to measure the perception of the driving forces and SE [8]. In this paper, we suggest a new approach for the prediction of cardiac arrhythmia classes using Particle Swarm Optimization (OEP) and the Multi-layer Perceptron (PMC). The PMC structure is optimized to improve the performance of the classifier for predicting heart disease. Linear and non-linear methods are used to extract the characteristics of the time series of the heart rate. Nonlinear and linear parameters such as the largest Lyapunov exponent (LLE), the spectral entropy (EE), the Hurst exponent (H), the SD1 / SD2 ratio, the low frequency normalized power components (nLF) and high frequency (nHF) types of cardiac arrhythmias [9]. The multi-layer power perceptron (MLP) with the Cuckoo (CS) search algorithm, called CS-MLP, is implemented to predict the 7-hour water level of the Ping River in central Chiang Mai, Thailand. The prediction performance of the CS-MLP model is compared with the regular multilayer perceptron (MLP) and the results of the previous work. The CS-MLP is the best among them with the mean absolute error in the blind test data set of 6,836 cm [10].

3 Methodology

Perceptron Multilayer
It is an artificial neural network, which solves problems that are not linearly separable
and highly complex, applying machine learning concepts such as self-regulation, has an input layer corresponding to the neurons or impulses that introduce the information to the network, also has a hidden layer where the processes are performed and level the weights of a neuron with respect to its previous layer, finally there is the output stage where the desired information is displayed and it contrasts the reliability of the network.

\[ A_i^k = f \left( u_i^k + \sum a_{ji}^{k-1} w_{ji}^{k-1} \right), \]
\[ i = 1, 2, 3 \ldots m_k \]
\[ k > 1 \]

Where \( k \) is the number of the layer.
\( J \) the subscript of the neuron in a layer \( k \)
\( I \) is the number of layers related to that neuron.

From the works done by David Rumelhart in [11] we can see the use of the sigmoid function and its qualities for a neural network. In this case it is more convenient to follow your model and use the same function:

\[ f(x) = (1 + e^{-x})^{-1} \]

\( W \) is the input value of the last layer \( k-1 \), to the node of the current layer.

Cleaning of data: The CAR provided the data in a PDF but it was necessary to convert the records to a plain text format to be entered into a PostgreSQL v9.2 database engine, this was possible thanks to a program in Java and the application pdfBox.

The relational model of the database is presented in Figure 1.

Fig 1. Logical schema of the relational model of the database. Source: Authors.

It was mainly thought of 4 entities to perform the entity-relationship model, due to the complexity of the data in the PDF, for the artificial neural network, the monitoring point was not used since, given the variables, the ICA calculation is independent of the point where the data were taken.
Measurement on the other hand is the relevant entity in this study, provides given a type of variable the respective ICA value that corresponds to it, which in the entity corresponds to the attribute "Clasificacion_segun_ica ".

![Fig. 2. Variable entity of the database. Source: Authors.](image)

Not all of the variables will be used, because it can harm the model, add complexity and subtract reliability, so we chose to choose 6 variables with greater influence in the calculation of ICA, later we will talk about these 6 variables. It also presents the corresponding calculations of the ICA that will be the expected output in the artificial neural network, while the variables selected will be the different inputs.

**Implementation**

Once the database has been polished, the most important variables in the database will be taken as inputs to the neural network, the 25,939 records from 2008 to 2013 will be used by the CAR to calibrate all weights, this process it will take until there is little or no improvement in the reliability process and finally with the records of 2014 and 2015 the quality of the network will be verified, the output of the network being indexes determining the quality of the water with respect to the time.

**4 Design**

The first is to determine which of the variables in the database are relevant, which thanks to a study of the ICA and its main variables were determined 6 variables with greater impact, which are: Dissolved Oxygen, Chemical Oxygen Demand, Suspended Solids Total, Total Nitrogen / Total Phosphorus, electrical conductivity and pH.
TABLE I. Selected variables as inputs

<table>
<thead>
<tr>
<th>Selected variables (inputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissolved oxygen</td>
</tr>
<tr>
<td>Chemical oxygen demand</td>
</tr>
<tr>
<td>Total</td>
</tr>
<tr>
<td>Suspended Solids</td>
</tr>
<tr>
<td>Total nitrogen / Total</td>
</tr>
<tr>
<td>phosphorus</td>
</tr>
<tr>
<td>Electric conductivity</td>
</tr>
<tr>
<td>Ph</td>
</tr>
</tbody>
</table>

The outputs are then determined, which will be a quality index between 0-1, which will tell us how good the water quality is at that point. Values between 0-0.4 are low quality indexes, between 0.41 and 0.7 are of average quality and if they are between 0.71 and 1.0 it is a good index of quality.

We will use an artificial neural network using the multilayer perceptron, with 6 inputs already named level 1, 2 hidden layers each 1 with 7 neurons and in the last layer our output that will be the ICA (water quality index).

Fig. 3. Neural network design with its levels. Source: Authors.

The (x) correspond to the input level where the 6 variables selected from the database are the most important, and the (y) corresponds to the output as water quality index.

We chose 2 hidden layers because this greatly reduces the dispersion of the data and offers a more stable artificial neural network, however if too many connections are used the network will lose precision due to the lack of dispersion in the results.

The changes in each connection are given by the partial derivative of the output and with respect to a weight w, where according to the level the equation may vary.
Development of an application to determine quality indexes

Level 1:
\[
\frac{\partial y_i}{\partial w_{jk}} = x_j^1 a_k^2 (1 - a_k^2) \left( \sum_{p=1}^{n_2} w_{kp} a_p^3 (1 - a_p^3) w_{pi}^2 \right) y_i (1 - y_i)
\]
\[
i = 1, 2, \ldots, n_4
\]
\[
j = 1, 2, \ldots, n_3
\]
\[
k = 1, 2, \ldots, n_2
\]

Level 2:
\[
\frac{\partial y_i}{\partial w_{jk}} = a_j^2 a_k^3 (1 - a_k^3) w_{ki}^3 y_i (1 - y_i)
\]
\[
i = 1, 2, \ldots, n_4
\]
\[
j = 1, 2, \ldots, n_2
\]
\[
k = 1, 2, \ldots, n_3
\]

Level 3:
\[
\frac{\partial y_i}{\partial w_{ji}} = a_j^3 y_i (1 - y_i)
\]
\[
i = 1, 2, \ldots, n_4
\]
\[
j = 1, 2, \ldots, n_3
\]

With these weights it is possible to determine the error in each neuron by comparing it with the expected values, as can be seen in the theory discussed in [11].

It is also necessary to take into account the changes in the thresholds of each neuron, but these can be calculated in the same way with these proposed equations.

With the different changes that can be presented the network can calculate the weights of the network, with them begin the implementation of the system and the input of data for its calibration.

5 Implementation

It was thought to perform the logic in java or MatLab, but given the amount of matrices that are possessed and the facility to perform functions and calculations was chosen by MatLab v.9.2.0.518641.

It was necessary to establish the connection between PostgreSQL and Matlab, for this is necessary to use an ODBC driver (Open DataBase Connectivity) in this case was used the devart ODBC PostgreSQL, is a paid driver to make the connection, but has a trial period of 1 month.

Now with the driver installed it is necessary to go to the ODBC data source on the computer and add the database using the driver.

Once this is done, it is only necessary to call the database from MatLab and save queries to matrices.
And finally, the test was performed using a select, obtaining the data matrix of size 2996x5.

![Instance database in command window](image)

Fig. 4. Instance database in command window. Source: authors

The Figure 5 shows how it is possible to extract data from the database, save it and print it from MatLab matrix so that it is already possible to process all the data of the 6 variables.

![MatLab data table](image)

Fig. 5. MatLab data table. Source: Authors.

Already with the data in a matrix of 757x6 with all the system inputs, it was divided into the calibration matrix and the verification matrix, from early 2008 until the end of 2013 are 559 records that correspond to the calibration matrix and from the beginning from 2014 until the end of 2015 there are 198 records that correspond to the data for the verification of the artificial neural network.

To begin, it is necessary to adapt the initial conditions for the 6 inputs, which must be normalized or in other words must be in a range of values between 0 and 1, since the neural network only accepts values in this range. This is done through the following formula:
Development of an application to determine quality indexes

\[ Nuevo\text{Valor} = \frac{Valor - Valor\text{Maximo}}{Valor\text{Maximo} - Valor\text{Minimo}} \]

For the case of the database that we use the minimum value is 0 in all the variables, for which the formula is directly:

\[ Nuevo\text{Valor} = \frac{Valor}{Valor\text{Maximo}} \]

Already with the 6 standard inputs the initial conditions of the neural network are created.

```matlab
nivel1=6;
nivel2=7;
nivel3=7;
nivel4=1;
w1=rand(nivel1,nivel2);
b1=rand(nivel2,1);
w2=rand(nivel2,nivel3);
b2=rand(nivel3,1);
w3=rand(nivel3,nivel4);
b3=rand;
```

Fig. 6. Initial conditions for the neural network. Source: authors

The dimensions of the levels are given, for the thresholds and weights of the networks are determined a random initial weight, this is explained in [11]. With these determined conditions the connections of the neurons (a) for each layer begin to realize making use of the sigmoidal function as follows:

```matlab
function a=sigmoid1(x)
    a= (1+exp(-x)).^(-1);
end
```

Fig. 7. Sigmoidal function. Source: Authors.

The functions for the value of the sigmoidal function in each neuron for each layer are performed.

Following are the values of the errors and alphas that will help to calibrate all the weights of the neurons.

We start with layer 4, due to the back propagation procedure, which tells us that the calculation of the errors is made from the last layer to the first, because it is easier.
The alpha of layer 4 is determined with the expected values and the obtained, this determines the correction index that is applied in the calibration of the neural network and is distributed by all the artificial neurons.

For layers 2 and 3 it is also necessary to determine the errors, this process is carried out in a similar way to layer 4.

Once all errors in the artificial neural network are determined, it is necessary to calibrate all the values of the weights and thresholds again for all the layers.

With all the ready functions you can begin to calibrate the neural network, for this purpose the 559 records from 2008 to 2013 will be repeated 50 times, calibrating the network a total of 27,950 times and then entering the 198 records from 2014 to 2015 for check the reliability of the artificial neural network.

6 Functionality test

Once the neural network was calibrated, the 198 registers were entered to check their reliability, with a margin of error of 0.1, the artificial neural network response was accepted, which reached 162 hits of 198 registers, which gives us a reliability of 82%, which is not the 90% expected in the objectives, but it is an equally optimal and quite good result.

The following table gives some expected results and the response of the artificial neural network.

<table>
<thead>
<tr>
<th>Expected</th>
<th>Obtained</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3500</td>
<td>0.5502</td>
</tr>
<tr>
<td>0.5600</td>
<td>0.5518</td>
</tr>
<tr>
<td>0.7900</td>
<td>0.5524</td>
</tr>
<tr>
<td>0.5500</td>
<td>0.5502</td>
</tr>
<tr>
<td>0.5200</td>
<td>0.5503</td>
</tr>
<tr>
<td>0.2800</td>
<td>0.5465</td>
</tr>
</tbody>
</table>

From the table at a glance it can be observed that for some results, the response obtained was extremely positive, but for another very imprecise, this is because the
neural network is too rigid, precluding dispersion and presenting an average range of values among all its solutions.

7 Discussion of Results

In [3] they test the artificial neural network that they pose with more than 2 hidden layers, obtaining for 5 hidden layers a reliability of 99.78%, this proves that among more hidden layers greater precision presents the artificial neural network and the results are so high already which are applied in a computational environment where the number of variables is less than a problem in an environmental context. On the other hand in [7] it talks about the treatment of restoration of images through artificial neural networks, in this case the results are in the range of 75% to 80% of precision, because the exit function fulfills many variables and obey a few, even so in the article they emphasize that the results are good. For [10] it is a good example of when a local optimum is reached and not the global one for the output function, in this it was necessary to perform 2 times the calibration in the first obtaining a reliability of 12% and in the second one was the 89%. For this case it was not necessary to do more calibrations due to this problem, this can be caused to that the output function has only 1 dimension (1 output).

8 Conclusions

The neural network was too rigid, so there is very little dispersion in the solutions obtained, it takes care to remove neurons from the hidden layer 2 until increasing the dispersion at an acceptable point. Even so the reliability rate is at 82%, although it is not the proposed value in the objectives is quite good for a first version and able to be increased by improving the design and implementation of the artificial neural network.

References


[3] Francisco Palenzuela, Melissa Shaffer, Matthew Ennis, Jeffrey Gorski, Derek McGrew, Daniel Yowler, Daniel White, Logan Holbrook, Chris Yakopcic,
https://doi.org/10.1109/naecon.2016.7856806


representations by back-propagation errors, *Nature*, 323 (1986), 533-536. https://doi.org/10.1038/323533a0

Received: November 8, 2017; Published: December 3, 2017