



# MODEL OF NEURONAL NETWORK WITH TRAINING OF PHYSICAL-CHEMICAL PARAMETERS PRESENT IN TILAPIA CULTIVATION

Johan Julián Molina Mosquera<sup>1</sup>, Diego F. Sendoya-Losada<sup>1</sup> and Arcelia Gutiérrez Méndez<sup>2</sup>

<sup>1</sup>Department of Electronic Engineering, Faculty of Engineering, Surcolombiana University, Neiva, Huila, Colombia

<sup>2</sup>Department of Software Engineering, Faculty of Engineering, Surcolombiana University, Neiva, Huila, Colombia

E-Mail: [diego.sendoya@usco.edu.co](mailto:diego.sendoya@usco.edu.co)

## ABSTRACT

In this paper, the neural network system is modeled, designed and simulated using the computer tool Matlab according to data obtained from measurements of physicochemical parameters such as temperature and dissolved oxygen present in tilapia cultivation in the Huila department.

**Keywords:** activation function, bias, layers, neural network, perceptron, weights.

## 1. INTRODUCTION

In freshwater lakes for tilapia cultivation, especially in their fattening stage, there is a high mortality rate of fish. According to the monitoring of the physical-chemical parameters such as temperature and dissolved oxygen of the water developed in the fish farm "Pezcomsi", located in the municipality of Campoalegre, in the department of Huila (Colombia) [1]; non-linearities that depend on the day and time are presented.

In this context, it was necessary to analyze these variables and record the values in a data table, which is used as the basis for the modeling of a neural network (NN) system.

The paper describes the design of NN layers, training functions, obtaining weights and bias. In addition, the Matlab code that performs the summations on each neuron with activation function "logsig" is presented. This code can be adapted to implement the neural network in any embedded system. Finally the results of the training of the NN with the data tabulated in Excel file are validated.

With this neural network design, it is expected to contribute to the reduction of tilapia mortality rates, due to sudden temperature and dissolved oxygen changes that occur daily in the ponds.

## 2. MATERIALS AND METHODS

### 2.1 Biological foundations

In human anatomy, a biological neuron is a cell specialized in processing information. The neuron is composed of the body of the cell known as soma and two types of branches: the axon and the dendrites as shown in Figure-1. The neuron receives the signals (impulses) from other neurons through its dendrites and transmits the signals generated by the body of the cell, through the axon [2].

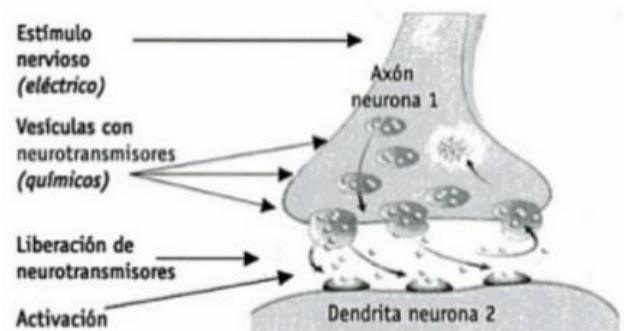


Figure-1. Structure of a biological neuron.

### 2.2 Artificial neural network

An artificial neural network can be defined as a connectionist model whose elements or nodes, connected to each other, simulate the functions performed by the brain cells which are called neurons [3].

They can also be defined as computer systems consisting of a large number of simple and highly interconnected elements that process information by responding dynamically to external stimuli.

The artificial neuron is a processing unit with four functional elements as shown in Figure-2.

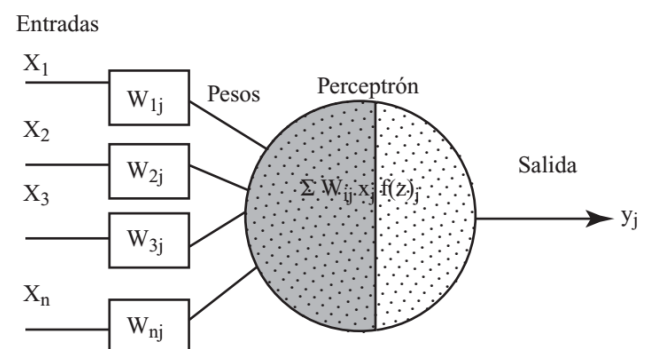


Figure-2. Structure of an artificial neuron.

**Receiver:** This is where one or more  $X_i$  input signals arrive. Generally they come from other neurons and are amplified or attenuated each according to a factor of weight  $W_i$ , which constitutes the connectivity between



the source neuron from which they originate and the target neuron.

**Adder:** It is responsible for effecting the algebraic sum of the input signals, penetrating them according to their weight, applying the following expression:

$$S = \sum W_i \cdot X_i$$

**Activation function:** Applies a function to the output of the simulator to decide whether the neuron is activated, firing an output or not.

**Output:** It produces the signal, according to the previous element, which in itself constitutes the output of the neuron.

Figure-3 shows the similarity between a natural biological neuron and its artificial simulation or representation.

Neurona biológica	Neurona artificial
- Señales que llegan a las sinapsis.	→ - Entradas a la neurona.
- Carácter excitador o inhibitor de las sinapsis de entrada.	→ - Pesos de entrada.
- Estimulo total de la neurona.	→ - $Net_j = \sum_{i=0}^n w_{ij}(t) \cdot x_i(t)$ .
- Activación o no de la neurona.	→ - Función de activación.
- Respuesta de la neurona	→ - Función de salida.

**Figure-3.** Analogy between biological and artificial neurons.

### 2.3 Main topologies

The architecture of a neural network consists of the organization and arrangement of neurons, forming layers or clusters of neurons more or less distant in the input and output of that network. The main fundamental parameters of the network are the number of layers, the number of neurons per layer, the degree of connectivity and the type of connections between neurons.

**Monolayer networks:** In the monolayer networks, connections are established between the neurons that belong to the single layer that constitutes the network. Monolayer networks are generally used in tasks related to what is known as self-association (regenerating incoming information that is presented to the network incompletely or distorted). For example: Perceptron, Adaline.

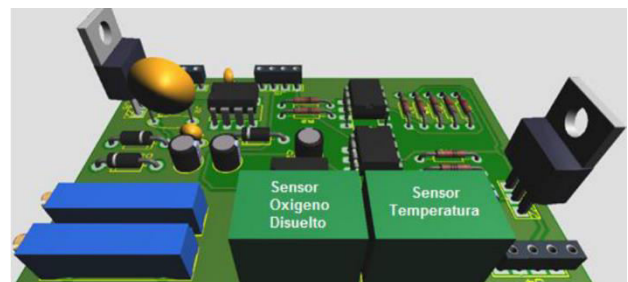
- **Perceptron:** The simplest mathematical model of a neuron is a perceptron. The perceptron can solve only problems that are linearly separable, that is, whose outputs are classified into two different categories and allow its input space to be divided into these two regions. The limitation of this algorithm is that if drawn on a plot, the "desired" and "unwanted" elements, must be separated in a hyperplane.

**Multilayer networks:** The neural network known as Multilayer Perceptron in a neuronal model with forward propagation, which is characterized by its organization in layers of disjoint cells, so that no neuronal output constitutes an input for neurons of the same layer or previous layers. This prevents backward or auto-recurrent connections. This allows solving problems that are not linearly separable, which is the main limitation of the perceptron, this network allows to establish regions of decision much more complex than those of two half-planes, as does the single-level perceptron [4].

- **Backpropagation:** This algorithm is the learning model of the multilayer perceptron network used in practice, due to its simplicity and efficiency in solving more complex problems. This algorithm is based on the descending gradient method, which is one of the oldest and most known multivalent function optimization methods. In a simplified way, this algorithm consists of learning a predefined set of input-output patterns, using a "propagation-adaptation" cycle with two differentiated phases, such as forward learning and backward learning.

### 2.4 Sensor adaptation

In the project different components were implemented to adapt the signal of the sensors, so that it can be processed by the microcontroller. To power this stage it is necessary to regulate the voltage of the batteries to 5V and -5V, which will power the amplifiers and other components of the system.



**Figure-4.** Circuit for adaptation of sensors.

For this reason and in order to obtain the required voltages, it was decided to implement the following elements:

**Regulator 7805:** It is a positive voltage regulator used in the system to regulate the voltage from 12 volts to 5 volts because it is the voltage with which the components operate.

**Regulator 7905:** This is a negative voltage regulator used mainly to obtain a voltage necessary for the operation of the instrumentation amplifier AD620 which needs a dual source. For this regulator to generate a negative voltage at the output (- 5 volts), it is necessary to have a higher negative voltage in the input signal; for which the following circuit was implemented.

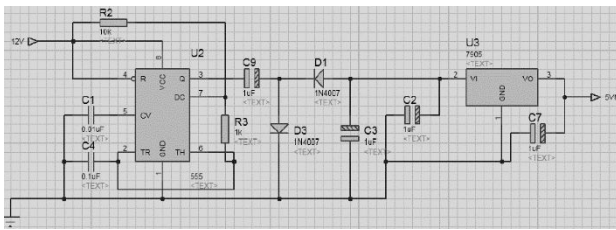


Figure-5. Negative voltage circuit.

This design consists of an integrated circuit 555 configured as an stable multivibrator which generates at the output a square wave with a voltage of 12 volts at a high level which charges the capacitor C9 through the diode D3. Then when at the output there is a voltage of 0 volts, the charge stored in the capacitor C9 is transferred through the diode D1 to the capacitor C3 generating there a negative potential difference, which is sufficient for the operation of the regulator 7905.

**Temperature sensor:** To monitor the system temperature, a PT100 RTD was used (see Figure-6). It is made of platinum and has an electrical resistance of 100 ohms at a temperature of 0°C. The adaptation of the sensor was performed as follows.



Figure-6. PT100 RTD sensor.

The PT100 is a resistive sensor, for this reason the way of balancing and sensing the sensor through a Wheatstone bridge (see Figure-7) was sought. This avoids erroneous measurements.

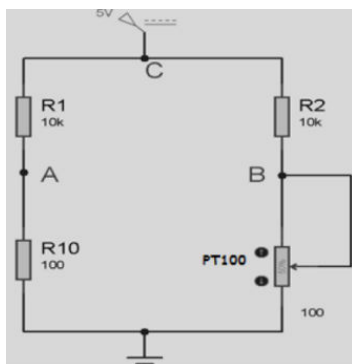


Figure-7. Wheatstone bridge.

After having the sensor in one of the branches of the bridge, the current source circuit was implemented (see Figure-8). The sensor connects to node C of the Wheatstone bridge to feed it in order to pass a very small

current (less than 1mA) through its branches, thus avoiding self-heating in the cables of the PT100 and improving its linearity.

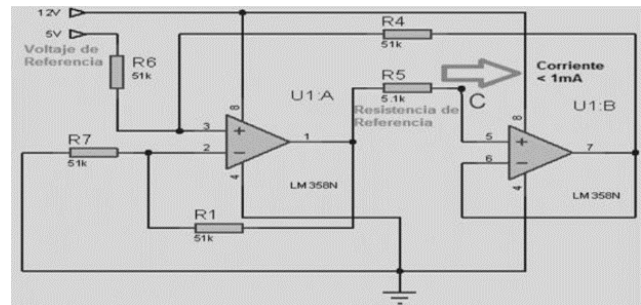


Figure-8. Current source circuit.

The current source circuit is made up of two operational amplifiers (LM358) and 5 resistors, which are configured such that the voltage at the output (resistor R5) is equal to the input or reference voltage. This keeps a very small and fixed current in the load.

Subsequently, and taking into account that the signal at the output of the Wheatstone bridge is very small and imperceptible to the system, amplification was performed. Then the obtained voltages were compared with a standard element for their calibration.

Amplification was performed using the instrumentation amplifier AD620 which offers significant gain and stability.

The equation to find the value of the resistance that gives the necessary amplification for the good operation of the system is facilitated by the manufacturer analog device in its datasheet and is shown below:

$$R_g = 49.4 \text{ K}\Omega / G - 1$$

Applying this equation and taking into account the gain required for the temperature sensor ( $G = 507$ ), the resistance value was found which was incorporated into the circuit by a trimmer to give it greater sensitivity and accuracy.

**Calibration with digital sensor:** In order to have greater reliability in the measurements, tests were carried out with a measurement standard (DS18B20 sensor). This is a high precision digital thermometer ( $\pm 0.5^\circ\text{C}$ ) with three terminals, two for power supply (red and black), and one DATA terminal (yellow) that transmits and receives information bits. In addition, it can be connected to the Arduino UNO board to read the temperature values as shown in Figure-9.

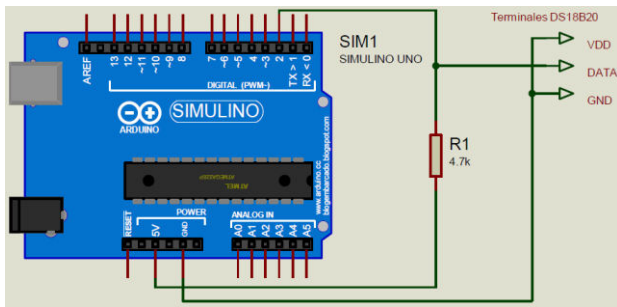


Figure-9. Digital sensor connection.



Figure-10. Digital temperature sensor DS18B20.

Table-1. Temperature test data.

Voltage PT100 (V)	Temperatura DS18B20 (°C)
0,6084	6
0,8619	9
0,9633	10
1,0647	11
1,1661	12
1,3689	14
1,4703	15
1,6731	17
1,8759	19
1,9773	20
2,1294	22
2,2308	23
2,4336	25
2,6364	27
2,7378	28
2,9913	30
3,3462	34
3,4476	35
3,8025	38
4,0053	40

Once the data were tabulated (Table-1), the linearization of the voltages obtained in the bridge Wheatstone was realized, finding the equation of linearization:

$$Y = 10.002X + 0.3498$$

Where  $Y$  is temperature in degrees Celsius and  $X$  is voltage in Volts.

Since the correlation coefficient is very close to 1 then it can be said that there exists a linear relationship between the two variables.

**Dissolved oxygen sensor:** To monitor the dissolved oxygen in the lake, the DO1200/T sensor was used. This is a device with an electrochemical galvanic probe that does not require external power to generate a control signal. This is a probe that generates a signal in millivolts that is proportional to the percentage of oxygen that is in the water. This probe indicates the percentage of oxygen in the water and does not measure the concentration of oxygen in ppm, therefore it is necessary to take into account a table supplied by the manufacturer where the concentration of oxygen is calculated from the temperature of the lake where the measurements are taken. For this the following procedure is performed.

First, the value generated by the outdoor sensor is taken, which will be at 100% saturation, this value will then be used in programming. Subsequently the sensor is immersed in the water to obtain the oxygen percentage and the following equation is applied to find the range.

$$r = \text{oxygen volt in the water} / \text{oxygen volt in the air}$$



Figure-11. Dissolved oxygen sensor.

The amplification of the oxygen sensor signal was performed by the operational amplifier AD620. Configured as a comparator, with a gain of 100, not requiring external power supply was connected directly as shown in Figure-12.

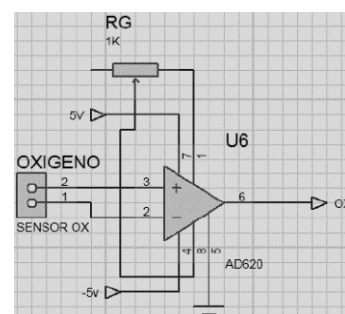


Figure-12. Amplification of the oxygen sensor signal.





### 3. RESULTS AND DISCUSSIONS

Daily records of the measurements obtained during each hour of the physicochemical variables sensed in the lake were taken.

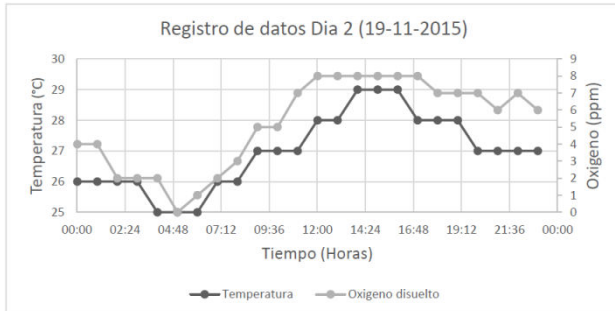


Figure-13. Data record of day # 2.

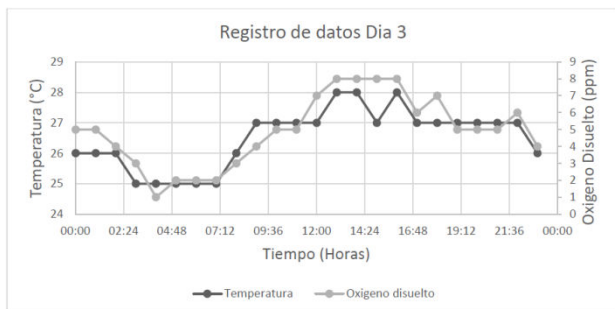


Figure-14. Data record of day # 3.

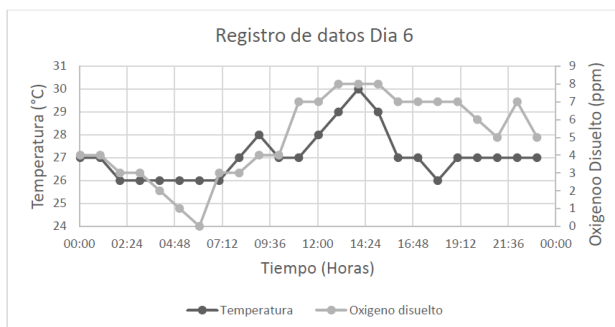


Figure-15. Data record of day # 6.

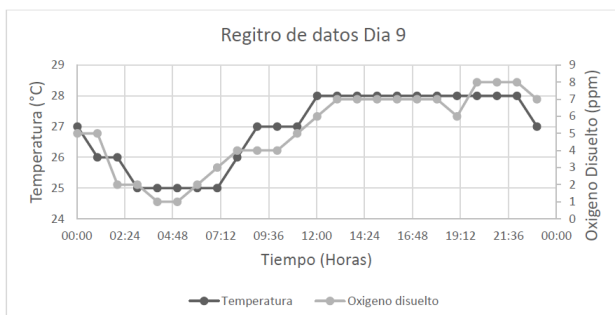


Figure-16. Data record of day # 9.

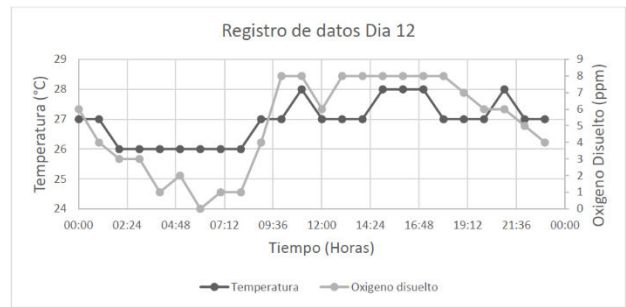


Figure-17. Data record of day # 12.

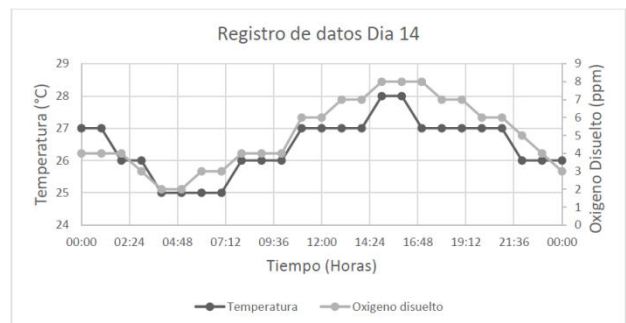


Figure-18. Data record of day # 14.

According to the records of the physical chemical parameters of temperature (°C) and dissolved oxygen (ppm) obtained hourly on different days, and verifying the "nonlinear" behavior of these variables, the database was constructed in an Excel file. The input fields are # Day, Hour, Temperature (°C), PTC temperature (Volts) and as output field, dissolved oxygen (ppm) and its conversion to decimal format in the range of 0.0 to 0.9, for purposes of neural network training and response of the activation function "logsig":

Table-2. database in Excel.

	A	B	C	D	E	F	G	H	I	J	K	L	M
3	ENTRADAS												
4	#Dia	2	2	2	2	2	2	2	2	2	2	2	2
5	Hora	0	1	2	3	4	5	6	7	8	9	10	11
6	T(°C)	26	26	26	26	25	25	25	26	26	27	27	27
7	PT100(V)	2,56	2,6	2,6	2,6	2,5	2,5	2,5	2,6	2,6	2,7	2,7	2,7
8	SALIDA												
9	O2 (ppm)	4	4	2	2	2	0	1	2	3	5	5	7
10	O2 (ppm)/10	0,4	0,4	0,2	0,2	0,2	0	0,1	0,2	0,3	0,5	0,5	0,7

	A	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
3	ENTRADAS													
4	#Dia	2	2	2	2	2	2	2	2	2	2	2	2	3
5	Hora	12	13	14	15	16	17	18	19	20	21	22	23	0
6	T(°C)	28	28	29	29	29	28	28	28	27	27	27	27	26
7	PT100(V)	2,8	2,8	2,9	2,86	2,9	2,8	2,8	2,8	2,7	2,7	2,7	2,66	2,56
8	SALIDA													
9	O2 (ppm)	8	8	8	8	8	8	7	7	7	6	7	6	5
10	O2 (ppm)/10	0,8	0,8	0,8	0,8	0,8	0,8	0,7	0,7	0,7	0,6	0,7	0,6	0,5

	A	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
3	ENTRADAS												
4	#Dia	3	3	3	3	3	3	3	3	3	3	3	3
5	Hora	1	2	3	4	5	6	7	8	9	10	11	12
6	T(°C)	26	26	25	25	25	25	26	27	27	27	27	27
7	PT100(V)	2,56	2,56	2,46	2,46	2,46	2,46	2,46	2,56	2,66	2,66	2,66	2,66
8	SALIDA												
9	O2 (ppm)	5	4	3	1	2	2	2	3	4	5	5	7
10	O2 (ppm)/10	0,5	0,4	0,3	0,1	0,2	0,2	0,2	0,3	0,4	0,5	0,5	0,7



A	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX
3 ENTRADAS												
4 #Dia	3	3	3	3	3	3	3	3	3	3	3	6
5 Hora	13	14	15	16	17	18	19	20	21	22	23	0
6 T(°C)	28	28	27	28	27	27	27	27	27	26	27	
7 PT100(V)	2,76	2,76	2,66	2,76	2,66	2,66	2,66	2,66	2,66	2,56	2,66	
8 SALIDA												
9 O2 (ppm)	8	8	8	8	6	7	5	5	5	6	4	4
10 O2 (ppm)/10	0,8	0,8	0,8	0,8	0,6	0,7	0,5	0,5	0,5	0,6	0,4	0,4

A	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ
3 ENTRADAS												
4 #Dia	6	6	6	6	6	6	6	6	6	6	6	6
5 Hora	1	2	3	4	5	6	7	8	9	10	11	12
6 T(°C)	27	26	26	26	26	26	26	27	28	27	27	28
7 PT100(V)	2,66	2,56	2,56	2,56	2,56	2,56	2,56	2,66	2,76	2,66	2,66	2,76
8 SALIDA												
9 O2 (ppm)	4	3	3	2	1	0	3	3	4	4	7	7
10 O2 (ppm)/10	0,4	0,3	0,3	0,2	0,1	0	0,3	0,3	0,4	0,4	0,7	0,7

A	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV
3 ENTRADAS												
4 #Dia	6	6	6	6	6	6	6	6	6	6	6	9
5 Hora	13	14	15	16	17	18	19	20	21	22	23	0
6 T(°C)	29	30	29	27	27	26	27	27	27	27	27	27
7 PT100(V)	2,86	2,96	2,86	2,66	2,66	2,56	2,66	2,66	2,66	2,66	2,7	2,7
8 SALIDA												
9 O2 (ppm)	8	8	8	7	7	7	7	6	5	7	5	5
10 O2 (ppm)/10	0,8	0,8	0,8	0,7	0,7	0,7	0,7	0,6	0,5	0,7	0,5	0,5

A	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH
3 ENTRADAS												
4 #Dia	9	9	9	9	9	9	9	9	9	9	9	9
5 Hora	1	2	3	4	5	6	7	8	9	10	11	12
6 T(°C)	26	26	25	25	25	25	25	26	27	27	27	28
7 PT100(V)	2,56	2,56	2,5	2,5	2,46	2,5	2,5	2,6	2,7	2,7	2,7	2,8
8 SALIDA												
9 O2 (ppm)	5	2	2	1	1	2	3	4	4	4	5	6
10 O2 (ppm)/10	0,5	0,2	0,2	0,1	0,1	0,2	0,3	0,4	0,4	0,4	0,5	0,6

A	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT
3 ENTRADAS												
4 #Dia	9	9	9	9	9	9	9	9	9	9	9	12
5 Hora	13	14	15	16	17	18	19	20	21	22	23	0
6 T(°C)	28	28	28	28	28	28	28	28	28	28	27	27
7 PT100(V)	2,8	2,8	2,8	2,8	2,8	2,8	2,8	2,8	2,8	2,8	2,7	2,66
8 SALIDA												
9 O2 (ppm)	7	7	7	7	7	7	6	8	8	8	7	6
10 O2 (ppm)/10	0,7	0,7	0,7	0,7	0,7	0,7	0,6	0,8	0,8	0,8	0,7	0,6

A	CU	CV	CW	CX	CY	CZ	DA	DB	DC	DD	DE	DF
3 ENTRADAS												
4 #Dia	12	12	12	12	12	12	12	12	12	12	12	12
5 Hora	1	2	3	4	5	6	7	8	9	10	11	12
6 T(°C)	27	26	26	26	26	26	26	26	27	27	28	27
7 PT100(V)	2,66	2,56	2,56	2,56	2,56	2,56	2,56	2,56	2,66	2,66	2,76	2,66
8 SALIDA												
9 O2 (ppm)	4	3	3	1	2	0	1	1	4	8	8	6
10 O2 (ppm)/10	0,4	0,3	0,3	0,1	0,2	0	0,1	0,1	0,4	0,8	0,8	0,6

A	DG	DH	DI	DJ	DK	DL	DM	DN	DO	DP	DQ	DR
3 ENTRADAS												
4 #Dia	12	12	12	12	12	12	12	12	12	12	12	14
5 Hora	13	14	15	16	17	18	19	20	21	22	23	0
6 T(°C)	27	27	28	28	28	27	27	27	28	27	27	27
7 PT100(V)	2,66	2,66	2,76	2,76	2,76	2,66	2,66	2,66	2,76	2,66	2,66	2,7
8 SALIDA												
9 O2 (ppm)	8	8	8	8	8	8	7	6	6	5	4	4
10 O2 (ppm)/10	0,8	0,8	0,8	0,8	0,8	0,8	0,7	0,6	0,6	0,5	0,4	0,4

A	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED
3 ENTRADAS												
4 #Dia	14	14	14	14	14	14	14	14	14	14	14	14
5 Hora	1	2	3	4	5	6	7	8	9	10	11	12
6 T(°C)	27	26	26	25	25	25	25	26	26	26	27	27
7 PT100(V)	2,7	2,6	2,6	2,5	2,5	2,5	2,5	2,6	2,6	2,6	2,7	2,7
8 SALIDA												
9 O2 (ppm)	4	4	3	2	2	3	3	4	4	4	6	6
10 O2 (ppm)/10	0,4	0,4	0,3	0,2	0,2	0,3	0,3	0,4	0,4	0,4	0,6	0,6

A	EG	EH	EI	EJ	EK	EL	EM	EN	EO
3 ENTRADAS									
4 #Dia	14	14	14	14	14	14	14	14	14
5 Hora	15	16	17	18	19	20	21	22	23
6 T(°C)	28	28	27	27	27	27	27	26	26
7 PT100(V)	2,8	2,8	2,7	2,7	2,7	2,7	2,7	2,6	2,6
8 SALIDA									
9 O2 (ppm)	8	8	8	7	7	6	6	5	4
10 O2 (ppm)/10	0,8	0,8	0,8	0,7	0,7	0,6	0,6	0,5	0,4

The red columns are the critical records when the low-oxygen dissolved event occurs.

For the design of the multilayer neural network, four layers were taken into account, distributed as follows:

Table-3. Multilayer neural network.

RED NEURONAL MULTICAPA			
# Capa	Neuronas	Pesos	Bias
Capa 1 (Entrada)	4	16	4
Capa 2 (Oculta)	7	28	7
Capa 3 (Oculta)	3	21	3
Capa 4 (Salida)	1	3	1
TOTAL	15	68	15

The connections of the multilayer NN are shown in Figure-19.

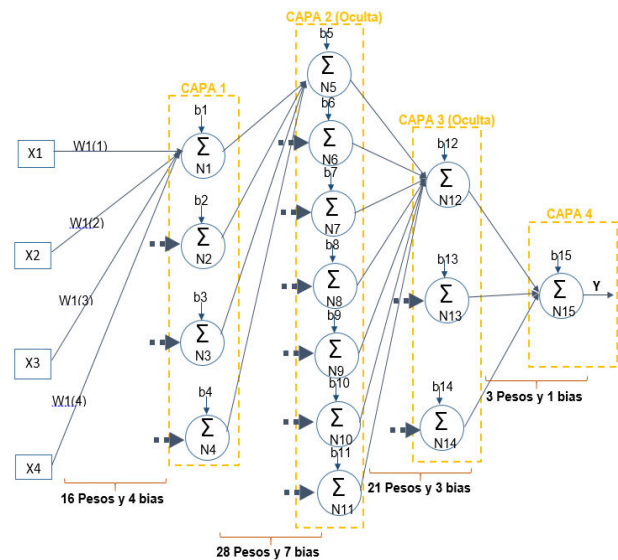


Figure-19. Multi-layer NN connections.

For the training of the multilayer NN, 576 input data and 144 output data were used, according to the values recorded in Table-2.

The Matlab code used for NN training is:

```

entrada=xlsread('varambientales.xlsx','B4:E07')
salida=xlsread('varambientales.xlsx','B10:E010')
net=newff(minmax(entrada),[4,7,3,1],{'logsig','logsig','logsig','logsig'},'trainlm');
net=init(net);

```



```
net.trainparam.goal=1e-10;
nete=train(net,entrada,salida);
sim(nete,entrada)
```

The screenshots of the training are as follows:

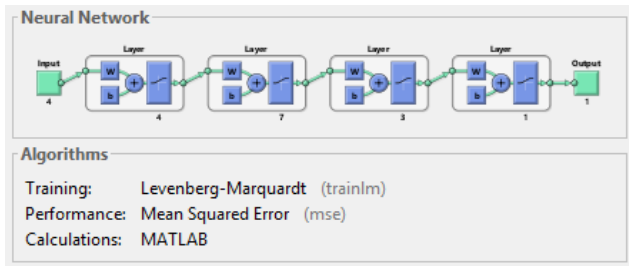


Figure-20. Neural network training.

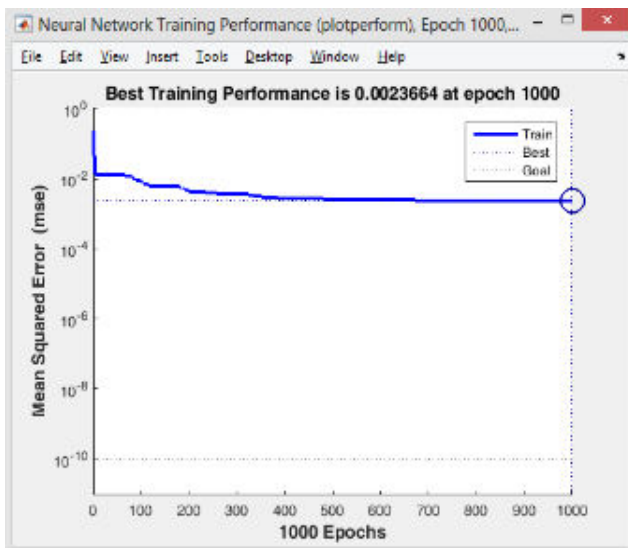


Figure-21. Performance.

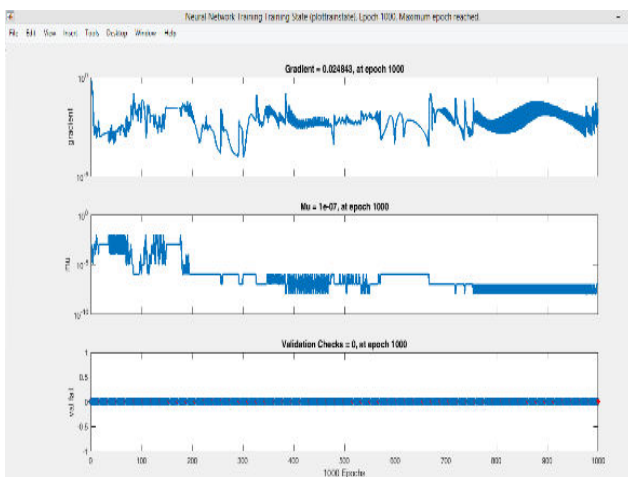


Figure-22. Training state.

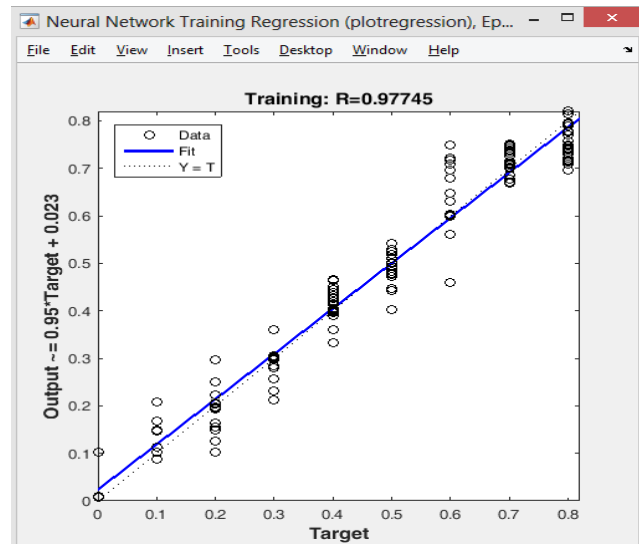


Figure-23. Regression.

The obtaining of the weights and bias of each layer are shown below.

For layer 1 (input) with 4 neurons, it contains 16 weights and 4 bias:

```
>> nete.iw{1,1}

ans =

    0.0048    0.5138   -10.7550    84.2594
    1.5110   -0.4143    12.5713   -100.6997
   -0.1855    0.2051   -0.7500     5.6777
   -0.2568   -0.3639     2.9032   -28.9913

>> nete.b{1,1}

ans =

    59.4938
   -64.8180
     3.0455
     7.0214
```

For layer 2 (hidden) with 7 neurons, it contains 28 weights and 7 bias:



```
>> nete.Lw{2,1}
```

```
ans =
```

```
    5.9967    23.1140    9.0983    10.3649
    3.2501    1.3655    3.6065    1.0119
   14.3192   -22.5154   -35.4510   44.3963
   15.3659  -143.0527   17.6864   12.8651
   -5.2376   -65.1208  -16.9292   21.8339
   11.5540  -190.6275    2.6541   17.1009
  -74.8212   14.6877   38.3661  -34.0310
```

```
>> nete.b{2,1}
```

```
ans =
```

```
 -36.6412
 -10.5984
  -6.3158
 125.0160
  48.8003
 173.6572
  -4.7686
```

For layer 3 (hidden) with 3 neurons, it contains 21 weights and 3 bias:

```
>> nete.b{3}
```

```
ans =
```

```
    8.6339
   -1.3708
    1.8439
```

```
>> nete.Lw{3,2}
```

```
ans =
```

```
 -0.6683  -133.3284  -8.8149  -4.2410  -4.8586    6.4157  -1.3871
 -11.8542   35.9908  -1.2008   1.0128  -2.1304   10.7918   2.2549
  13.8847  -40.1825   1.3650  -1.2440   2.0879  -12.6764   8.4983
```

For layer 4 (output) with 1 neuron, it contains 3 weights and 1 bias:

```
>> nete.Lw{4,3}
```

```
ans =
```

```
    7.6167  -54.6477  -49.9918
```

```
>> nete.b{4}
```

```
ans =
```

```
  48.0849
```

Once the respective weights and bias were known, the Matlab code was implemented, which allows to find the different sums of each neuron according to the activation function "logsig", in order to verify the results obtained in each neuron, starting from the data vector of (Layer 1) to the output of the neuron (Layer 4).

For layer 1 of the NN, the summations of each neuron with the activation function "logsig", are represented in the following Matlab code:

```
%Neurona 1 de la capa 1
%bias de la neurona 1
```

```
b1=59.4938;
```

```
%Entradas
```

```
de
```

```
#Dia,Tiempo(Hora),Temperatura(°C)
```

```
y
```

```
PT100(Voltios).
```

```
X=[2 0 26 2.56];
```

```
W1=[0.0048 0.5138 -10.7550 84.2594];
```

```
for j=1:4
```

```
    A1(j)=X(j)*W1(j);
```

```
end
```

```
%Sumatoria de entradas x pesos y bias1
en la neurona1 de capa1
```

```
N1=A1(1)+A1(2)+A1(3)+A1(4)+b1
```

```
Nlogsig1=1/(1+exp(-N1))
```

```
%Neurona 2 de la capa 1
```

```
%bias de la neurona 2
```

```
b2=-64.8180;
```

```
%Entradas
```

```
de
```

```
#Dia,Tiempo(Hora),Temperatura(°C)
```

```
y
```

```
PT100(Voltios).
```

```
X=[2 0 26 2.56];
```

```
W2=[1.5110 -0.4143 12.5713 -100.6997];
```

```
for j=1:4
```

```
    A2(j)=X(j)*W2(j);
```

```
end
```

```
%Sumatoria de entradas x pesos y bias2
en la neurona2 de capa1
```

```
N2=A2(1)+A2(2)+A2(3)+A2(4)+b2
```

```
Nlogsig2=1/(1+exp(-N2))
```

```
%Neurona 3 de la capa 1
```

```
%bias de la neurona 3
```

```
b3=3.0455;
```

```
%Entradas
```

```
de
```

```
#Dia,Tiempo(Hora),Temperatura(°C)
```

```
y
```

```
PT100(Voltios).
```

```
X=[2 0 26 2.56];
```

```
W3=[-0.1855 0.2051 -0.7500 5.6777];
```

```
for j=1:4
```

```
    A3(j)=X(j)*W3(j);
```

```
end
```

```
%Sumatoria de entradas x pesos y bias3
en la neurona3 de capa1
```

```
N3=A3(1)+A3(2)+A3(3)+A3(4)+b3
```

```
Nlogsig3=1/(1+exp(-N3))
```

```
%Neurona 4 de la capa 1
```

```
%bias de la neurona 4
```

```
b4=7.0214;
```

```
%Entradas
```

```
de
```

```
#Dia,Tiempo(Hora),Temperatura(°C)
```

```
y
```

```
PT100(Voltios).
```

```
X=[2 0 26 2.56];
```

```
W4=[-0.2568 -0.3639 2.9032 -28.9913];
```

```
for j=1:4
```

```
    A4(j)=X(j)*W4(j);
```

```
end
```

```
%Sumatoria de entradas x pesos y bias4
en la neurona4 de capa1
```

```
N4=A4(1)+A4(2)+A4(3)+A4(4)+b1
```

```
Nlogsig4=1/(1+exp(-N4))
```





The values of Nlogsig1, Nlogsig2, Nlogsig3 and Nlogsig4 are obtained from the application of the activation function "logsig", previous sum of products between weights and input. Then the summation is done with the bias of each of the neurons of Layer 1 (input). Also, the process is performed with layer 2 and 3 (hidden), until reaching the layer 4 of output.

The Matlab code shown serves as a reference model to implement the neural network in the embedded system that requires it. Then, some data obtained from the simulation in Matlab with the data represented in the Excel table are compared:

<code>&gt;&gt; sim(nete,[12; 6; 26; 2.56;])</code>	
<code>ans =</code>	
<code>0.0724</code>	ENTRADAS
	#Dia 12
	Hora 6
	T(°C) 26
	PT100(V) 2.56
	SAIDA
	O2 (ppm) 0
	O2 (ppm)/10 0
<code>&gt;&gt; sim(nete,[2; 4; 25; 2.46;])</code>	ENTRADAS
<code>ans =</code>	#Dia 2
<code>0.1985</code>	Hora 4
	T(°C) 25
	PT100(V) 2.46
	SAIDA
	O2 (ppm) 2
	O2 (ppm)/10 0.2
<code>&gt;&gt; sim(nete,[2; 0; 26; 2.56;])</code>	ENTRADAS
<code>ans =</code>	#Dia 2
<code>0.4000</code>	Hora 0
	T(°C) 26
	PT100(V) 2.56
	SAIDA
	O2 (ppm) 4
	O2 (ppm)/10 0.4
<code>&gt;&gt; sim(nete,[2; 5; 25; 2.46;])</code>	ENTRADAS
<code>ans =</code>	#Dia 2
<code>6.3681e-04</code>	Hora 5
	T(°C) 25
	PT100(V) 2.46
	SAIDA
	O2 (ppm) 0
	O2 (ppm)/10 0

The red columns are the critical records when the low-oxygen dissolved event occurs.

Due to the nonlinear behavior of the physical chemical parameters of dissolved oxygen and temperature during the monitoring by hours on certain days, it is necessary to design an optimal neural network system that evaluates the conditions of stable or critical parameters in a tilapia cultivation.

NN training required the maximum number of iterations and epochs (1000) given the amount of data processed, the number of layers and the types of activation function present in each layer.

The "logsig" activation function for the four layers of the NN was the one that best answered the training, because the dissolved oxygen level data ranged from zero (critical state) to 0.9 (optimal state).

This multi-layer NN design will allow to adapt to a fuzzy system with the purpose of strengthening the implementation in embedded systems.

With a greater amount of data for NN training, it would be possible to predict the environmental parameters.

Electronic systems exposed to the environment may be affected at certain times by critical conditions such as high temperature, noise, rain, and insects. For the case study, if on a given day the dissolved oxygen sensor "is

out of operation", the NN system will act intelligently delivering the training output data to the input of the controller that would perform the respective evaluation to activate or disable some type of actuator.

## REFERENCES

- [1] Sergio Andres Dussan, Oscar Vanegas Morales y Johan Julian Molina Mosquera. Diseño e implementación de un sistema para monitoreo de parámetros físico-químicos en cultivo de alevinos a través de aplicación móvil para disminuir tasa de mortalidad. 2016.
- [2] P. Ponce. Inteligencia Artificial, con aplicaciones a la ingeniería. <https://lelinopontes.files.wordpress.com/2014/09/inteligencia-artificial-con-aplicaciones-a-la-ingenierc3ada.pdf>.
- [3] R. Beltran, Bioinformática simulación, vida artificial e inteligencia\_artificial. <http://www.editdiazdesantos.com/libros/lahoz-beltra-rafael-bioinformatica-simulacion-vida-artificial-e-inteligencia-artificial-C03006450101.html#contenido>.
- [4] J. Damian. Redes neuronales: Conceptos básicos y aplicaciones. [https://www.fro.utn.edu.ar/repositorio/catedras/quimica/5\\_anio/orientadora1/monograias/mati-ch-redesneuronales.pdf](https://www.fro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/mati-ch-redesneuronales.pdf).