




# Enhancing Electoral Surveys with Artificial Neural Networks

Optimización de Encuestas Electorales mediante Redes Neuronales Artificiales

Yessica Yazmin Calderon-Segura\* , Gennadiy Burlak  y José Antonio García Pacheco 

Centro de Investigación en Ingeniería y Ciencias Aplicadas, Universidad Autónoma del Estado de Morelos  
Avenida Universidad 1001, Colonia Chamilpa, 62209, Cuernavaca, Morelos, México

\*[ycalderons@uaem.mx](mailto:ycalderons@uaem.mx)

## KEYWORDS:

Artificial Neural Network,  
Conservative, Algorithm

## ABSTRACT

The objective of this study is to search for the main factors that can influence to predict the results of voting surveys. A system is developed that allows the optimization of Artificial Neural Networks to identify the factors that affect the electoral result, through a computational method that allows the evaluation of the characteristics that influence a successful electoral vote. An Artificial Neural Network with three layers and a back propagation learning algorithm is used. The first phase loads the system by developing a random synthetic database. This will contain the data that will serve as input to the Artificial Neural Network to optimize the most outstanding attributes that affect a vote. The system identifies the inputs to the Artificial Neural Network, and the iterations that can be carried out to optimize its outputs.

## PALABRAS CLAVE:

Red Neuronal Artificial,  
Conservadora, Algoritmo

## RESUMEN

El objetivo de este estudio es buscar los principales factores que pueden influir para predecir los resultados de las encuestas de votación. Se desarrolla un sistema que permite la optimización de Redes Neuronales Artificiales para identificar los factores que afectan el resultado electoral, a través de un método computacional que permite evaluar las características que influyen en un voto electoral exitoso. Se utiliza una Red Neuronal Artificial con tres capas y un algoritmo de aprendizaje de retro propagación. La primera fase carga el sistema desarrollando una base de datos sintética aleatoria. Éste contendrá los datos que servirán de entrada a la Red Neuronal Artificial para optimizar los atributos más destacados que afectan una votación. El sistema identifica las entradas a la Red Neuronal Artificial y las iteraciones que se pueden realizar para optimizar sus salidas.

• Recibido: 18 de julio de 2023 • Aceptado: 18 de mayo de 2024 • Publicado en línea: 1 de junio de 2024

## 1. INTRODUCTION

Artificial Neural Networks arise from the interpretation of the functioning of a human brain. Although the first to relate computing to the human brain was Alan Turing in 1936, it was Warren McCulloch and Walter Pitts who created the theory of how a neuron works [10].

Joel W. Johnson in 2012 presents the main factors affecting vote inequality among incumbent cohorts (members of the same party and district), indicating the strong

influence of vote splitting incentives on electoral environments focused on the candidates [11].

The study developed by Ching-Hsing Wang in 2014 indicates that awareness and emotional stability can significantly increase female participation in electoral votes, but have no effect on male participation [18]. Furthermore, openness to experience has opposite effects on male and female participation. As openness to experience increases, men are more likely to vote, while women are less likely to cast ballots.

However, extraversion and agreeableness are not associated with participation, regardless of gender [18]. Orlando D'Adamo in July 2015, makes a study of the usefulness and scope of the use of social networks during electoral campaigns. The authors [5] present results of an investigation that analyzes the use of social networks made by candidates for deputies and senators for the city of Buenos Aires in the legislative elections.

In 2017, Dimitrios Xefteris studied several factors influencing electoral voting, including religion, race, culture, and others, making use of optimized data access for the Data Warehouse maximizes differentiated voting participation [8]. Artificial Neural Networks or ANN for its acronym in English, have numerous publications made each year. Several researchers have done studies on the different neural networks, the simplest being the multilayer perceptron which has a pattern recognition architecture, which means that its neurons are only connected from one layer to the next [13]. Artificial Neural Networks can be used to predict the difficulties of the electoral process, since they have been used for different complex problems with adaptive and cognitive mechanisms of human learning. The literature tells us that training a neural network is an NP-hard optimization problem that represents several theoretical and computational limitations [7]. Dat Thanh Tran in January 2019, proposes a library to avoid the bottleneck in Machine Learning using the perceptron [15], Recurrent Neural Networks for Sequential Data Modeling have also been published, for voice recognition considering the morphology of words, the syntax and semantics [16]. The literature shows that the information that influences the electoral choice is uncertainty. it is clear that various factors and attributes influence to know the winner of the elections, it should be noted that doing this manually represents a very laborious workload and knowing the main attributes that influence is always of great importance for both voters and the nominated candidates.

## 2. PROBLEM STATEMENT

A database with artificial data is randomly generated with which the Artificial Neural Network can evaluate the attributes that affect a vote. These must be safe to avoid inconsistencies since the data must be managed by software which, according to the restrictions of the problem and the input information obtained from the database, will show the efficiency of the proposed algorithm. This software is an algorithm which will give a solution to find the attribute more efficiently with a back propagation method. The database will have the necessary initial information that is required for the Artificial Neural Network. These represent the different possibilities to make a choice when voting, among which are the economy, socio-cultural movements, and work. Taking that into account, the input data to the Artificial Neural Network are the following:

1. Age
2. Economic income
3. Education
4. Debt

And the output variables are the political parties to take into consideration, in this case we will use the most common political ideology patterns among which are usually classified as Left, Right and agnostic or central. With which we will create three fictitious political parties called:

1. Conservative (right)
2. Moderate (center)
3. Liberal (left)

To understand the way in which supporters of each party are classified, Fig. 1 shows the schematic distribution of the political spectrum.

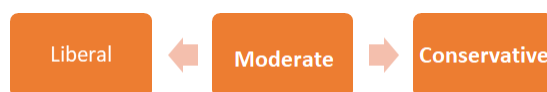


Figure 1. Political spectrum.

This political distribution Fig. 1 has its origin in 1789 in France during the National Constituent Assembly in which the revocation of political power to the monarchy was discussed. Those who were against it were on the right and those who promoted a change, seeking national sovereignty, were on the left. Péronnet in 1985 Said distribution was modified while preserving the same political bases since at the beginning of the 19th century the aristocracy was supplanted by the bourgeoisie as the predominant class [12]. With which we can say that liberal or left politics seeks political equality, for the progress of the people. Without imposing on the most the law of the least. Rightwing or conservative politics represents maintaining the current political order, in which it represents those who possess power and wealth seeking the individual good without taking into account all social classes.

Moderate politics has gained popularity in recent years because it represents the union between liberal politics and conservative politics trying to get the best of both parties. The idea of joining and discerning the vote on the scale from Left to Right entails acceptance to the way that group of people work, taking into account the way in which they deal with problems, that is, the means used to resolve conflicts. Thus, generating the right with the other part of the scale, attributing pejorative connotations to the identity of the opposition and vice versa [3]. Recently there is a tension due to not so clear circumstances between political geometries, leaving out the debate by differentiating political thought, focusing on the discrediting of the adversary. These acts cause confusion in the voters, frustrating the reasoning behind their vote, this means that the electoral decision in many cases is overwhelmed by the economy, socio-cultural movements and work, when making a decision. The disturbance in our representation Politics is represented by sociocultural movements and other events not initially considered as natural phenomena, electoral fraud, among others.

### 3. BACKPROPAGATION NEURAL NETWORK

An Artificial Neural Network is a complex mathematical function inspired by the operation of its biological namesake. But it's also the interaction of many simpler parts called neurons, working together, which have numerical inputs and outputs. And its goal is to solve problems in a way similar to the human brain. The neural network is the integration of many neurons in which each neuron performs a weighted sum whose weighting is given by the weight assigned to each of the connections of entry. This means that each connection that reaches the neuron will have an associated value that will define the intensity with which the input variable will affect the neuron and therefore will influence the result that the output layer throws [2]. Backpropagation networks use feedback as a supervised method, consisting of three layers: input, hidden, and output. Having better precision because the error propagates inversely, that is, it starts from the output layer passing between the hidden layer to reach the input layer [14].

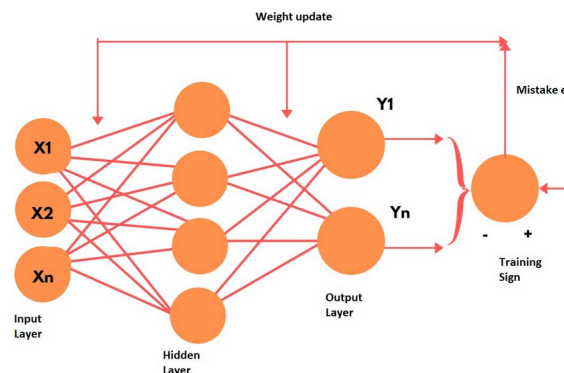


Figure 2. Representation of a backpropagation neural network.

As shown in Fig. 2, the  $X$  variables from  $X_1$  to  $X_n$  represent the inputs to the network, and the  $Y$  variable from  $Y_1$  to  $Y_n$  represents the result obtained from the neurons in the output layer.

#### 4. KOHONEN NEURAL NETWORK

Unlike the backward propagation Neural Network, the Kohonen Neural Network, as shown in Fig. 3, is simpler as it has only one layer, which uses an unsupervised method, therefore it does not have a specific vector to be trained, among others. reasons that affect the veracity of the output result [1].

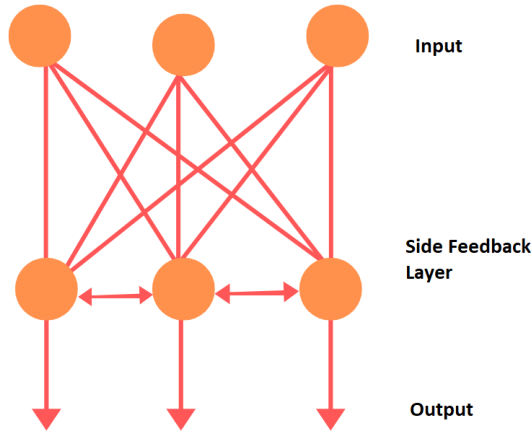


Figure 3. Representation of a Kohonen neural network.

#### 5. MODEL OF AN ARTIFICIAL NEURON

It is the base unit of a neural network, basically it is an elementary processor which processes a vector  $x$  and produces an output resulting from the weighted sum [9]. The model of an artificial neuron is an imitation of the process of a biological neuron, as seen in Fig. 4.

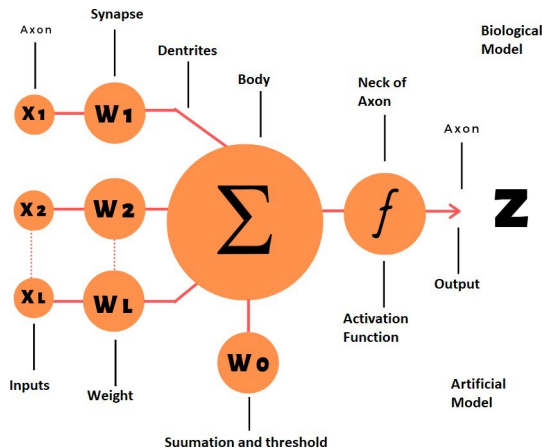


Figure 4. Homonymous representation of a neuron.

Where the  $X_i$  are the inputs (through the dendrites) to the neuron. These undergo a multiplying effect on the weight  $W_i$ , due to their communication to the nucleus of the neuron, and  $b$  is bias. Thus, obtaining the following formula as can be seen in Fig. 4.

$$Z = \text{activation function} \left[ \sum (\text{weight} * \text{input}) + \text{bias} \right] \quad (1)$$

The basic characteristic of a neural network is that it is composed of three layers. The input layer is in charge of receiving the input values and sending these values to the second layer called the hidden layer and these carry out their process and transfer the information to the output layer, the network can contain more layers if required, they can be modified by adding or removing input or output variables or by changing the learning or training process [4]. A conventional neural network is composed of three characteristics:

- The interconnection model between the different layers of the network.
- Development of learning in the variation of weights between the interconnections.
- The activation function modifies the weighted result format of the network in the output activation value.

In this case, a Trained Neural Network will be used with a backward propagation algorithm oriented towards gradient descent and the use of the chain-rule.

#### 6. ACTIVATION FUNCTION

The activation function is used to modify the data and enter it within a shorter range to make a simpler calculation. Next, we have the activation function that we will use in our backpropagation algorithm [19]. This function modifies the input values. where the high values are close to 1 and the very low values are close to 0 and is represented in Eq. 2. called Sigmoid function.

$$f(x) = \frac{1}{1 - e^{-x}} \quad (2)$$

### 6.1. BACKPROPAGATION

Backpropagation is an algorithm widely used in the training of forward neural networks for supervised learning. It works by calculating the gradient of the loss function with respect to each weight by the chain rule, iterating backwards one layer at a time from the last layer to avoid redundant calculations of middle terms in the chain rule, and is based on partial derivatives of calculus. Each weight and bias value have an associated partial derivative. You can think of a partial derivative as a value that contains information about how much and in which direction a weight value should be adjusted to reduce error. The collection of all partial derivatives is called a gradient. However, for simplicity, each partial derivative is commonly called a gradient [17].

### 6.2. CHAIN RULE

If a variable  $y$  depends on a second variable  $u$ , which in turn depends on a third variable  $x$ , then the rate of change of  $y$  with respect to  $x$  can be calculated as the product of the rate of change of  $y$  with respect to  $u$  multiplied by the rate of change of  $u$  with respect to  $x$ . If  $g(x)$  is differentiable at the point  $x$  and  $f(x)$  is differentiable at the point  $g(x)$ , then  $f$  is differentiable in  $x$ . Also, let  $y = f(g(x))$   $y = g(x)$ , then Eq. 3 is obtained. What is the chain rule.

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx} \quad (3)$$

### 6.3. COST FUNCTION

The cost function tries to determine the error between the estimated value and the real value, in order to optimize the parameters of the neural network. In this case we will use the root mean square error. In regression analysis, Mean Square Error refers to the mean of the squared deviations

of the predictions from the true values, over one space outside the test sample, generated by a model estimated over one sample space particular. Its formula is shown in the Eq. 4 root mean square error.

$$c(d_j) = \frac{1}{2} \sum_j (y_j - d_j)^2 \quad (4)$$

### 6.4. MATHEMATICAL MODEL OF OUR ARTIFICIAL NEURAL NETWORK

Initially the network values are randomly generated. With this, it is very probable that the error obtained is very high, for which reason the network must be trained to obtain the minimum possible error. We will begin by calculating the derivative of the parameters in the last layer, in which the result obtained from the weighted sum that is show below in Eq. 5  $z^l$  is the result of the weighted sum, is the weight 1 is the representation of the bias.

$$z^l = w^l x + b^l \quad (5)$$

Subsequently, the activation and cost function are added to this result, resulting in the error obtained from the network represented in the Eq. 5 where Result of the weighted sum activation function  $F$ . cost. What we will look for will be the partial derivative of the cost with respect to the weight and bias parameters, with which we will have to calculate two derivatives. As we have said, we are going to start working from back to front, therefore we begin to calculate the derivative of the parameters of the last layer. The number of the layer to which the parameter belongs if our neural network has layers. To calculate this derivative it is important to analyze which is the path that connects the value of the parameter and the final cost in the last layer of this path is not very long, although it still has several steps, previously we saw that in the operation of the neuron the parameter participated in a weighted sum now which we will refer to as which would then be passed by the activation function represented in the Eq. 2 and the

result of the activations of the neuron in the last layer would conform to the result of the network that would later be evaluated by the cost function Having to determine the network error. With this, a composition of functions is formed and we will use a mathematical calculation tool called the Chain Rule represented in the Equation Eq. 6 To calculate the derivative of composition of functions, what it tells us is that to calculate the derivative of a composition of functions we simply have to multiply each of the intermediate derivatives. Considering the functions Eq. 7 and Eq. 10 Represents the weighted sum of the last layer.

$$z^l = w^l a^{l-1} + b^l \quad (6)$$

$$c(a(z^l)) = error \quad (7)$$

We will obtain the derivative of the weight with respect to the cost Equation Eq. 8 and the derivative of the bias with respect to the cost Eq. 9 Derivative of weight with respect to cost.

$$\frac{\partial c}{\partial w^l} = \frac{\partial c}{\partial a^l} \cdot \frac{\partial a^l}{\partial z^l} \cdot \frac{\partial z^l}{\partial w^l} \quad (8)$$

$$\frac{\partial c}{\partial b^l} = \frac{\partial c}{\partial a^l} \cdot \frac{\partial a^l}{\partial z^l} \cdot \frac{\partial z^l}{\partial b^l} \quad (9)$$

Thus, obtaining three partial derivatives where the derivative of the activation with respect to the cost Eq. 9 to obtain the cost variation of the network when the output of the activation of the neurons in the last layer is varied, that is to say that a derived from the cost function with respect to the output of the neural network. The cost function that we will use in this case will be the root mean square error described in Eq. 4 with the parameters of our network Eq. 10 partial derivative of activation with respect to cost.

$$\frac{\partial c}{\partial a^l} \quad (10)$$

Thus, the derivative of the function with respect to the output of the network would be represented Eq. 12 indicates the derivative

of the activation function with respect to the output of the network.

$$c(d_j) = \frac{1}{2} \sum_j (y_j - d_j)^2 \quad (11)$$

$$\frac{\partial c}{\partial a_j} = (d_j - y_j) \quad (12)$$

$$a^l(z^l) = \frac{1}{1 + e^{-z}} \quad (13)$$

$$\frac{\partial a^l}{\partial z^l} = a^l(z^l) \cdot (1 - a^l(z^l)) \quad (14)$$

$$z^l = \sum_i a_i^{l-1} w_i^l + b^l \quad (15)$$

$$\frac{\partial z^l}{\partial b^l} = 1 \quad (16)$$

$$\frac{\partial z^l}{\partial w^l} = a_i^{l-1} \quad (17)$$

$$\frac{\partial c}{\partial b^l} = \frac{\partial c}{\partial a^l} \cdot \frac{\partial a^l}{\partial z^l} \cdot \frac{\partial z^l}{\partial b^l} \quad (18)$$

$$\frac{\partial c}{\partial z^l} = \delta^l \quad (19)$$

We continue with the activation function of the weighted sum Eq. 13 Activation function of the weighted sum of and its Derivative of the activation function with respect to the weighted sum in Eq. 14. This reveals the output variation of the neuron when the weighted sum of the neuron is varied, thus calculating the activation function. This derivative is calculated depending on the type of activation function, in this case we will use the sigmoid function Eq. 2 with all that we would only be missing two partial derivatives with respect to bias Eq. 16 Partial derivative with respect to bias and weight Eq. 17. Partial derivative with respect to weight. These are obtained by deriving the weighted sum of the neuron as shown below in Eq. 15 Derivation of the weighted sum of the neuron. By applying the chain rule and using two partial derivatives in the derivative of the bias with respect to cost Eq. 18, the error is obtained as a function of the value of represented in Eq. 19 What is the weighted sum calculated within the neuron, that is, what this derivative tells us is to what degree the cost error is modified when there is a small

change in the sum of the neuron if this derivative is large, before a small change in the value of the neuron this will be reflected in the final result and on the contrary if the derivative is small it does not matter how we vary the value of the sum since this will not affect the error of the network, that is, the derivative from here is the one that will tell us what responsibility the neuron has in the final result and therefore in the error, this is what we said before, if the neuron is a part responsible for the final error then we should use this information to extract part of that mistake for this one. We will need the derivation of the bias with respect to the cost with error imputed to neuron 1 is represented by Eq. 20 which will be the error imputed to the neuron or derivation of the bias with respect to the cost with error imputed to neuron 2 shown in the Eq. 21, which is calculated in Eq. 22, which is the error imputed to the neuron.

$$\frac{\partial c}{\partial b^l} = \delta^l \cdot \frac{\partial z^l}{\partial b^l} \quad (20)$$

$$\frac{\partial c}{\partial b^l} = \delta^l \cdot 1 \quad (21)$$

$$\frac{\partial c}{\partial b^l} = \delta^l \quad (22)$$

Later we will do the same, but with the partial derivative of the weight with respect to the cost of the error imputed to neuron 1 Eq. 23 which reduces to Eq. 24 Derivative of the weight with respect to the cost with error imputed to neuron 2.

$$\frac{\partial c}{\partial w^l} = \delta^l \cdot \frac{\partial z^l}{\partial w^l} \quad (23)$$

$$\frac{\partial c}{\partial w^l} = \delta^l \cdot a_i^{l-1} \quad (24)$$

We have deduced three different expressions that allow us to obtain the partial derivatives that we are looking for the last layer, one that tells us how to calculate the error of the neurons in the last layer, and another for each of the partial derivatives and thus we obtain the result of the last layer. To obtain the result of the previous layer, we

apply the Chain Rule again to the following composition. Eq. 25 Indicates the Error in the penultimate layer that with the chain rule generates two derivatives; derivative of weight with respect to cost in the penultimate layer Eq. 26 and derivative of bias with respect to cost in the penultimate layer Eq. 27.

$$c(a^l(w^l a^{l-1}(w^l a^{l-2} + b^{l-1}) + b^l)) \quad (25)$$

$$\frac{\partial c}{\partial w^{l-1}} = \frac{\partial c}{\partial a^l} \cdot \frac{\partial a^l}{\partial z^l} \cdot \frac{\partial z^l}{\partial a^{l-1}} \cdot \frac{\partial a^{l-1}}{\partial z^{l-1}} \cdot \frac{\partial z^{l-1}}{\partial w^{l-1}} \quad (26)$$

$$\frac{\partial c}{\partial b^{l-1}} = \frac{\partial c}{\partial a^l} \cdot \frac{\partial a^l}{\partial z^l} \cdot \frac{\partial z^l}{\partial a^{l-1}} \cdot \frac{\partial a^{l-1}}{\partial z^{l-1}} \cdot \frac{\partial z^{l-1}}{\partial b^{l-1}} \quad (27)$$

Calculated that is the error of the layer and these derivatives are operated the same as before minus 1 and the activation of the previous layer and this here is the derivative of the function of the activation in this expression. The only thing that would need to be calculated would be this derivative that tells us about how the weighted sum of a layer varies when the output of a neuron in the previous layer is varied. This derivative is also simple to calculate and is basically the parameter matrix Eq. 28, which connects both layers, what it does is move the error from one layer to the previous layer, distributing the error based on the weights of the connections, with this we would again have an expression from which to obtain the derivatives partials we are looking for. Again, the block highlighted in Eq. 29. It becomes this derivative Eq. 30. Which again represents the error of the neurons in this layer.

$$\frac{\partial c}{\partial z^{l-1}} = \delta^{l-1} \quad (28)$$

The effectiveness of the back propagation algorithm lies in the fact that what we have done in this layer is already extensible to the rest of the layers of the network, applying the same logic, we take the error from the previous layer, we multiply it by the weight matrix in a transformation that comes to represent the back propagation of the errors Eq. 28 and we calculated the partial

derivatives with respect to the parameters and so on, going through all the layers of the network until the end, with a single step we had calculated all the errors and the partial derivatives of our network using only four expressions.

With what in the end we will obtain four expressions to calculate error starting with the last layer Eq. 30 that computationally indicates the error of the last layer and later perform a retro propagation of the error of the previous layer Eq. 31 to do the retro propagation from the error to the previous layer and the calculation of the derivatives of each layer Eq. 32 the derivative bias of the layer is developed using the error with the , Eq. 28 of our network the weight of the house is calculated using the error.

$$\delta^l = \frac{\partial c}{\partial a^l} \cdot \frac{\partial a^l}{\partial z^l} \quad (29)$$

$$\delta^{l-1} = w^l \delta^l \cdot \frac{\partial a^{l-1}}{\partial z^{l-1}} \quad (30)$$

$$\frac{\partial c}{\partial b^{l-1}} = \delta^{l-1} \quad (31)$$

$$\frac{\partial c}{\partial w^{l-1}} = \delta^{l-1} a^{l-2} \quad (32)$$

To calculate the partial derivatives that we are looking for in the Eq. 32 and Eq. 31 expressions are quite intuitive because we are simply telling ourselves how we have to use the error from the previous layer to calculate the error in this layer there are two different cases. one is in the last layer where the error already belongs to the cost function Eq. 30 and others are the rest of the layers of our network that depend on another layer Eq. 31 and of course once we have these two expressions that tell us how we can calculate the error in the current layer with respect to the previous one.

## 7. IMPLEMENTATION OF THE ARTIFICIAL NEURAL NETWORK

The program randomly creates an artificial database, it starts by randomly generating

1,000 synthetic data items.

```

Generating 1000 artificial data elements with 4 features
Generating all input data
Generating weights and biases
5.027 -7.785 -0.660 5.432 3.150 -1.344 -2.918 8.877 -7.975 2.849
-9.427 -5.039 -3.598 9.795 3.642 3.098 -4.345 2.307 4.085 4.036
8.996 -8.132 -6.782 -2.361 5.959 -6.611 5.876 -3.855 6.461 7.729
1.126 4.335 3.986 -9.728 9.838 6.126 7.175 -8.750 0.288 0.505
-4.543 9.876 3.822
Neural network created: Local
Leaving all input data
Generation of all input data performed
    
```

Figure 5. Artificial data generation.

Each data item has four input values and three output values as can be seen in Fig. 5. The four input values are all between -10.0 and +10.0 and correspond to predicted values that have been normalized, so that values below zero are less than the average, and values above zero are greater than the average. The three output values correspond to a variable to predict that can take one of the three categorical values. To predict a person's political inclination: conservative, moderate or liberal.

[ 0 ]	-1.09	-9.10	0.85	5.52	0.00	0.00	1.00
[ 1 ]	-1.92	8.16	6.53	-1.54	0.00	1.00	0.00
[ 2 ]	7.94	3.66	-2.63	-3.38	0.00	1.00	0.00
[ 799 ]	1.61	2.00	-5.77	-7.44	0.00	1.00	0.00

Figure 6. Input and output values to the artificial neural network.

The program randomly divides the data into a training set of 800 items and a test set of 200 items Fig. 6. The training set is used to create the neural network model, and the test set is used to estimate the accuracy of the model. After the data is partitioned, the program creates an instance of a neural network with n hidden nodes. The number of hidden nodes is arbitrary and must be determined by trial and error. To finish the program, it generates a final neural network with the optimal weights and bias with values generated during the previous training of the network to obtain the final result.

## 8. ALGORITHM

In Fig. 7, we present a flowchart illustrating the application of the backpropagation



algorithm within an artificial neural network. The primary goal is to develop an algorithm capable of predicting electoral votes. The process commences by defining the parameters of the network, followed by the generation of a randomized database, as depicted in Fig. 5. Subsequently, an artificial neural network is created.

The input data extracted from the randomized database is then introduced into the input layer of the neural network. After traversing hidden layers, the processed data ultimately reaches the output layer. The ensuing step involves a comparison between the neural network's output and the anticipated results derived from the training data. This assessment yields an error or loss metric, which quantifies the network's performance.

If the calculated error falls within an acceptable range, the results are displayed, as illustrated in Fig. 6, and the process concludes. However, if the error remains outside this acceptable threshold, the data is looped back to the input layer, where it undergoes further processing iterations. This iterative procedure continues until the error converges to an acceptable value.

### 9. RESULTS

These results are tuned in order to know if the dependence of the parameters of the Artificial Neural Network finds the normalized average value. And know if the number of hidden Layers affects the results of the voting. We can see in Table Fig. 8 the optimal number of hidden layers (NumHidden), obtained when there is a hidden layer, because it is the smallest value with a mean of 0.0667240612695. We also see the worst result when having 8 hidden layers in the network, with an average of 0.116324855868.

### 10. CONCLUSIONS

In conclusion, the parameters with the best performance in our Network Artificial Neural are those expressed below in Fig. 9.

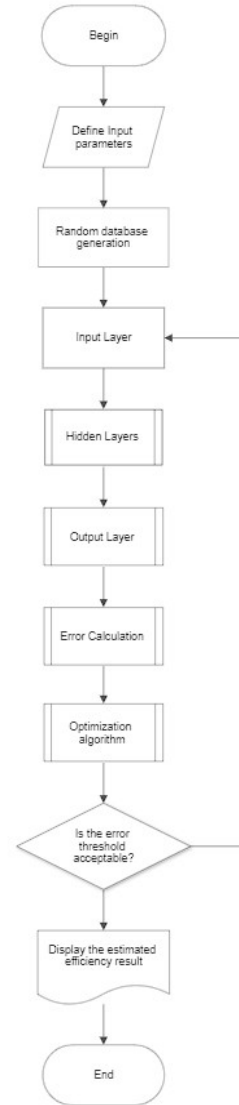


Figure 7. Algorithm flowchart.

It only remains for us to take the approach based on electoral votes, with experimental data, and the disturbance as a determining event in the alteration, or not, of the final result. We have three possible outcomes which are on a scale of 0 to 10. If you are representing a person who is younger than average, has a much lower income than

average, is somewhat more educated than average, and has more debt than average. The person has a liberal political view. Likewise, if another person on the same scale is older than average, has a higher-than-average income, is slightly less than, equal to, or slightly more educated than average, and has less than average debt, that person has a conservative political vision. But when a person is within the average or a little below or a little above in all the parameters, he is in a moderate ideology.

Parametro	Media	Varianza $\sigma^2$	DesvEst $\sigma$
1	0.066724061269486	0.044457338648618	0.21084908975051
2	0.078388657565341	0.042771050500891	0.20681163047781
3	0.090033319857806	0.041068010963467	0.20265243882931
4	0.080320990205377	0.042636895162207	0.20648703388399
5	0.090428669427933	0.041056037753971	0.20262289543379
6	0.087554860608494	0.041524859910155	0.20377649486938
7	0.070790801802586	0.043888701151205	0.20937693557602
8	0.11632485586751	0.037476119064442	0.19358749718007
9	0.094901702846709	0.04052670337977	0.2013126184271
10	0.088486127323551	0.041378803004582	0.20341780405014

Figure 8. Results in the hidden layers parameter.

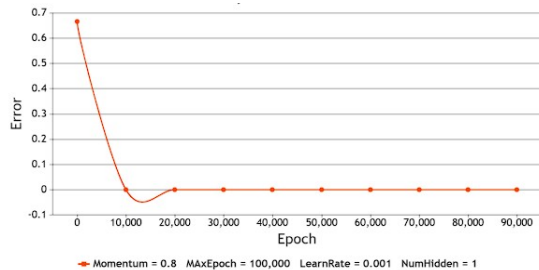


Figure 9. Network parameter tuning results.

## REFERENCES

[1] Gámez A. H., Cabrera, J., Salas, O., Bravo B. J. (2016). Aplicación de mapas de Kohonen para la priorización de zonas de mercado: una aproximación práctica. *Revista EIA*. 2016, 13, 157-169. doi: [10.24050/reia.v13i25.1024](https://doi.org/10.24050/reia.v13i25.1024).

[2] Ballesteros, A. (2001). *Neural Network Framework*. Last accessed 4, Sep, 2023, <http://www.redes-neuronales.com.es/tutorial-redesneuronales/tutorial-redes.htm>, 2001.

[3] Bobbio, N. *Derecha e Izquierda*. Madrid: Taurus, 1998, 120-132.

[4] Callejas, I., Píneros J., Del Valle J., Hernán F., Delgado F. Implementación de una red neuronal artificial tipo SOM en una FPGA para la resolución de trayectorias tipo laberinto. *II International Congress of Engineering Mechatronics and Automation (CIIMA)*. 2013, 1-6. doi: [10.1109/CIIMA.2013.6682790](https://doi.org/10.1109/CIIMA.2013.6682790).

[5] D'Adamo, O., García Beaudoux, V., Kievsky, T. Comunicación política y redes sociales. Análisis de las campañas para las elecciones legislativas de 2013 en la ciudad de Buenos Aires. *Revista Mexicana De Opinión Pública*, 2015, (19), 107-125. doi: [10.22201/fcyps.24484911e.2015.19.50206](https://doi.org/10.22201/fcyps.24484911e.2015.19.50206).

[6] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J. Flexible, High Performance Convolutional Neural Networks for Image Classification. *Proc. of the 22nd International Joint Conference on Artificial Intelligence ICAI*, 2011. doi: [10.5591/978-1-57735-516-8/IJCAI11-210](https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210).

[7] Rojas, J., Trujillo-Rasúa, R., Bello, R. A continuation approach for training Artificial Neural Networks with meta-heuristics. *Pattern Recognition Letters*, 2019. doi: [10.1016/j.patrec.2019.05.017](https://doi.org/10.1016/j.patrec.2019.05.017).

[8] Xeftaris, D. Multidimensional electoral competition between differentiated candidates, *Games and Economic Behavior*, 2017, vol. 105(C), 112-121. doi: [10.1016/j.geb.2017.07.005](https://doi.org/10.1016/j.geb.2017.07.005).

[9] Fukushima, K. *Neocognitron: A Self-organizing Neural Network Model*. Tokyo: Springer-Verla, 1980, (36), 193-202.

[10] Hilera, J., Martínez-Hernando, V. *Redes neuronales artificiales: fundamentos, modelos y aplicaciones*. Madrid: RA-MA Editorial, 1995.

[11] Johnson, Joel W. and Hoyo, Verónica, Beyond Personal Vote Incentives: Dividing the Vote in Preferential Electoral Systems, *Electoral Studies*, 31(1), 131-142. Recuperado (September 28, 2021) de <https://ssrn.com/abstract=2001725>.

[12] Péronnet, M. *Vocabulario básico de la Revolución Francesa*. Barcelona: Critica, 1985.

[13] Ramírez, F. *Historia de la IA: Frank Rosenblatt y el Mark I Perceptrón, el primer ordenador fabricado específicamente para crear redes neuronales en 1957*, Recuperado el 29, septiembre, 2021, de <https://telefonicatech.com/blog/historia-de-la-ia-frank-rosenblatt-y-e>, 2018.

[14] Rojas, R. *Neural Networks: A Systematic Introduction*. Springer, 1996.

[15] Tran, Dat, Kiranyaz, S., Gabbouj, M., Iosifidis, A. PyGOP: A Python library for Generalized Operational Perceptron algorithms. *Knowledge-Based Systems*, 2019, 182(4), 855-863. doi: [10.1016/j.knsys.2019.06.009](https://doi.org/10.1016/j.knsys.2019.06.009).

[16] Tjjskens, A., Roels, S., Janssen, H. Neural networks for metamodelling the hygrothermal behaviour of building components, *Building and Environment*, 2019, 162, 106282, doi: [10.1016/j.buildenv.2019.106282](https://doi.org/10.1016/j.buildenv.2019.106282).

[17] Caraballo, I. Diseño de redes neuronales con aprendizaje combinado de retropropagación y búsqueda aleatoria progresiva aplicado a la determinación de austenita retenida en aceros TRIP. *Revista de Metalurgia*, 2010, 46(6), 499-510. doi: [10.3989/revmetalmadrid.0924](https://doi.org/10.3989/revmetalmadrid.0924).

[18] Wang, C. H. Gender differences in the effects of personality traits on voter turnout, *Electoral Studies*, 2014, 34, 167-176. doi: [10.1016/j.electstud.2013.10.005](https://doi.org/10.1016/j.electstud.2013.10.005).

[19] Hu, Z., Bodyanskiy, Y., Tyshchenko, O. A deep cascade neuro-fuzzy system for high-dimensional

online fuzzy clustering. 2016 IEEE First Int. Conf. on Data Stream Mining & Processing (DSMP), 2016, 318-322. doi: [10.1109/DSMP.2016.7583567](https://doi.org/10.1109/DSMP.2016.7583567).

## ABOUT THE AUTHORS



Dr. Yessica Yazmin Calderon Segura. She has experience in algorithm optimization, mathematical models, processes to minimize time, Neural Networks, simulation, percolation systems, nanostructures and electromagnetic phenomena. She has published co-authored articles in international journals with a high impact factor. She as well as other knowledge on the topics of image processing, neural networks and systems. She is currently a member of the SNI, as a candidate. She is the author and co-author of 14 articles in international journals. She has participated in 24 presentations at national and international conferences. Under her direction they have graduated: 1 bachelor's thesis and 2 master's thesis at FCAeI-CIICAp-UAEM. Currently 3 FCAeI-UAEM bachelor's theses in process, under her direction.



Dr. Gennadiy Burlak. In 1975 he studied bachelor's and master's degrees at the Kyiv National University (KNU), at the Faculty of Physics and at the Department of Theoretical Physics. He also obtained the Ph. D. (candidate in Physical-Mathematical Sciences) and the D. Sc. (Doctor in Physical-Mathematical Sciences) at the KNU in 1979 and 1988, respectively. He worked as a professor in the Department of Theoretical Physics. Currently, he is Professor-Researcher C of the Centro de Investigación en Ingeniería y Ciencias Aplicadas (CIICAp) of the Universidad Autónoma del Estado de Morelos (UAEM), since 1998. Dr. Burlak is the

author and co-author of four books and 150 articles in international magazines. He has participated in 157 presentations at national and international conferences. Under his direction they have graduated: 5 doctoral theses and 8 master's and bachelor's theses. Currently 2 doctoral theses in process under his direction.



Lic. José Antonio García Pacheco. A Master's student in Engineering and Applied Sciences at the Centro de Investigación en Ingeniería y Ciencias Aplicadas (CIICAp). He obtained his degree in Computer Science from the Universidad Autónoma del Estado de Morelos (UAEM) in 2022. His passion and experience focus on artificial neural network research, software development, and algorithm optimization. His focus and dedication are exemplary, making him a promising researcher and professional in the field of applied computing.