

Meteorological patterns recognition using Artificial Neural Networks programmed with the Swish activation function

Reconocimiento de patrones meteorológicos utilizando Redes Neuronales Artificiales programadas con la función de activación Swish

AGUILERA-MENDEZ, José María†, JUÁREZ-TOLEDO, Carlos*, MARTÍNEZ-CARRILLO, Irma and FLORES VÁZQUEZ, Ana Lilia

Universidad Autónoma del Estado de México

ID 1st Author: *José María, Aguilera-Méndez* / ORC ID: 0000-0002-9826-421X, CVU CONACYT ID: 66670

ID 1st Co-author: *Carlos, Juárez-Toledo* / ORC ID: 0000-0002-7440-3246, Researcher ID Thomson: C-1368-2016, CVU CONACYT ID: 39912

ID 2nd Co-author: *Irma, Martínez-Carrillo* / ORC ID: 0000-0002-7952-4418, Researcher ID Thomson: B-9264-2016, CVU CONACYT ID: 39914

ID 3rd Co-author: *Ana Lilia, Flores-Vázquez* / ORC ID: 0000-0003-0524-9764, Researcher ID Thomson: C-1325-2016, CVU CONACYT ID: 92382

DOI: 10.35429/JOCT.2021.15.5.21.28

Received March 15, 2021; Accepted June 30, 2021

Abstract

Artificial neural networks are a set of tools that are widely used for the information classification. Its expansion within artificial intelligence has been due to its use in the Machine Learning area. A fundamental part of the artificial neural networks algorithm is the so-called activation function, the above because it is the part that triggers the process as a whole and due to its result the neuron/perceptron sends its outputs. Back-propagation activation function of an artificial neural network is also described; this is artificial neural network with a simpler functioning whose adaptation has made it especially attractive to pattern recognition; also, a different algorithm such as Swish is introduced. As part of the pattern recognition study, three wind classifications present on the Mexican Republic Atlantic coast are formed, each group is made up of graphic files referring to meteorological maps with wind indicators in order to feed the network and as new maps are generated, the Artificial Neural Network will be an aid in the meteorological patterns detection.

Artificial Neural Network, Swish, Pattern recognition

Resumen

Las redes neuronales artificiales son un conjunto de herramientas que son ampliamente utilizadas para la clasificación de información. Su expansión dentro de la Inteligencia Artificial se ha debido a su utilización en el área de Aprendizaje Automático. Una parte fundamental del algoritmo es la llamada función de activación que tiene la función de disparar el proceso en su conjunto la cual debido a su resultado envía sus salidas. También se describe a la red neuronal artificial de retro-propagación o hacia atrás, tiene el funcionamiento más simple y cuya adaptación la ha hecho especialmente atractiva al reconocimiento de patrones y se da entrada a los diferentes algoritmos como el Swish. Como parte del estudio de reconocimiento de patrones se forman tres clasificaciones de los diferentes tipos de viento presentes en la costa Atlántica de la república mexicana, cada grupo se encuentra conformado por archivos gráficos referentes a mapas meteorológicos con indicadores de viento con el objetivo de alimentar la red y según se generen nuevos mapas, la red neuronal artificial resultante es un auxiliar en la detección de patrones meteorológicos.

Redes neuronales artificiales, Swish, Reconocimiento de patrones

Citation: AGUILERA-MENDEZ, José María, JUÁREZ-TOLEDO, Carlos, MARTÍNEZ-CARRILLO, Irma and FLORES VÁZQUEZ, Ana Lilia. Meteorological patterns recognition using Artificial Neural Networks programmed with the Swish activation function. Journal of Computational Technologies. 2021. 5-15: 21-28

* Author Correspondence: (e-mail: jaguileram001@alumno.uaemex.mx)

† Researcher contributing as first author.

Introduction

In the last twenty-five years, the use and advance of Artificial Neural Networks (ANN) within a great variety of fields of science has been well known; this document will first give an introduction to the theory of ANNs and their different types. Later, it will be explained why they are used to a great extent in the field of pattern recognition with the different algorithms that can build them in addition to the formulas that compose them; finally, it will be discussed its adequacy for the patterns recognition in images and its use in the detection of two meteorological phenomena types using the Swish activation function, which has shown, on other types of functions, a good performance in the recognition tasks as well as the feedback type and quality obtained on ANN learning.

Neural Networks and Deep Learning

Deep Learning (DL) is considered as a subset of Machine Learning (ML) in which multilayer ANN “learn” from the data that is entered. Within each ANN layer, the DL algorithms perform a series of calculations that allow them to progressively “learn” and, as far as possible, improve the results precision over time (Kim, 2019); this is a great advantage since learning does not depend on the model, but on the data quality and quantity that are entered into the algorithms.

In the field of Artificial Intelligence (AI), ANN is considered a ML tool widely used in DL as one of the supervised learning techniques, with the ability to solve large and increasingly complex pattern classification problems and regressions (BBVA, 2019; Mathworks, 2020) that now include voice recognition and moving patterns.

Method

As part of the study of Artificial Intelligence (AI), Artificial Neural Networks (ANN) have, within their neurons, the evaluation mechanisms that allow their learning. These mechanisms are known as activation functions, which are mathematical equations that are responsible to return a result from input values. It is sought that the range of the value delivered by an activation function is normalized and they are in the range between [-1, 1] or [0, 1].

The activation function must be designed, configured and programmed in an efficient way since it could become a point of failure due to the fact that it has control of the ANN complete operation; this is due to the large number of calculations that they carry out. So we have that an output function of a simple node would be:

$$Y = \sum (weight * input + bias) \quad (1)$$

With an output range of $(-\infty, \infty)$; so it is advisable to apply a function that limits the output values and sets them in the desired range.

$$Y = activation\ function(\sum (weight * input + bias)) \quad (2)$$

Then the activation function decides whether a neuron should be activated or not and it is a non-linear transformation that is done on the input before propagating it to the next layer of neurons or to finalize the output. Not all proposed functions can be activation functions as they must meet the following monotonous properties (Li & Yuan, 2017):

- Continuous
- Discriminating
- Polynomial (Non-Linear)
- Monotonously Increases

Currently three types of activation functions are recognized, which are described below.

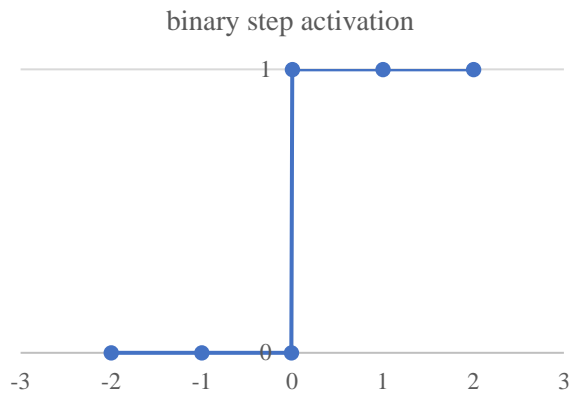
A. Binary step or unit step activation function: The function is mainly used in problems where the result can only have any of the values [0, 1] for any input value and is generally used in linear classification perceptrons; depending on the established threshold, the neuron is activated and sends the same signal to the next layer and so on; one of its limitations is that it does not allow the sending of multiple values. Its general formula is:

$$a_j^i = f(z_j^i) = \begin{cases} 0, & \text{if } z_j^i < 0 \\ 1, & \text{if } z_j^i > 0 \end{cases} \quad (3)$$

As an example, when the binary activation function is defined as

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x > 0 \end{cases}$$

Its representation is shown in Graphic 1.

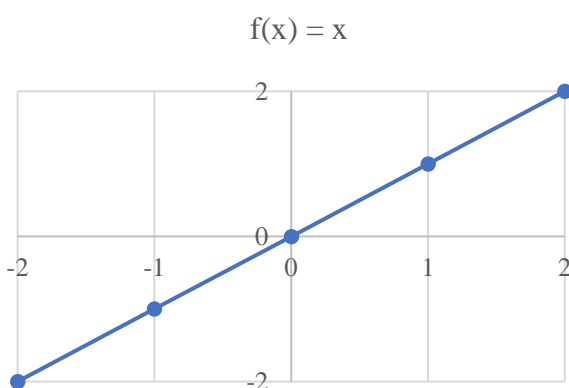


Graphic 1 Presentation of the binary step function.
Source: Own Elaboration

B. Linear activation function: Also known as the identity activation function, it allows the input to be equal to the output; so if you have a multilayer neural network and apply the linear function, it is said to be a linear regression. The linear activation function general form is:

$$f(x) = ax \tag{4}$$

Where the characteristics of (4) are: its domain and range take values $(-\infty, \infty)$; when calculating $\frac{df(x)}{dx} = a$ you get a constant; therefore it cannot be optimized with gradient descent. Although its application could be interesting, this function has a couple of limitations that discourage its use: since the derivative is constant, the gradient has no relation to the input and, backward propagation is constant since the change is Δx as indicated in Graphic 2.



Graphic 2 Graph of linear or identity function
Source: Own Elaboration

C. Non-linear activation function: Modern ANNs use non-linear activation functions. They are highly complex since the inputs and outputs of the network and their assignments can process data such as images, video, audio and data sets that, as their definition indicates, are not linear, also called multidimensional; they can be of the following three types:

1. Sigmoid or logistic activation functions: Transform the entered values to a scale (0, 1), where high values tend asymptotically to 1 and very low values asymptotically tend to 0. Its general formula is:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

It is a function of slow convergence, bounded between 0 and 1, it is not centered at 0 and it has shown good performance in the ANN last layers. As part of the sigmoid functions, there is the so-called hyperbolic tangent, whose peculiarity consists in that it transforms the entered values to a scale (-1, 1), where high values tend asymptotically to 1 and low values tend asymptotically to -1. Its general formula is:

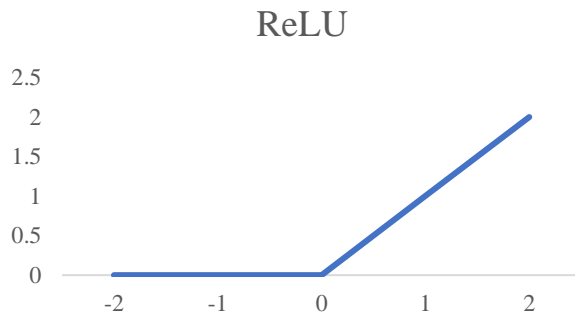
$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{6}$$

It is a function of slow convergence, bounded between -1 and 1, centered at 0 and has shown good performance in recurrent ANN. Sigmoid-based models have been developed, including the inverse hyperbolic tangent, whose peculiarity is that the output varies between $[-\pi/2, \pi/2]$. The softmax function, also known as the soft argmax function or multiple class logistic regression, in which case its use is recommended when the classes are mutually exclusive. The Gudermannian and Gaussian Error Linear Units (GELU) functions to mention a few (A.I. Wiki, 2021).

2. Rectified Lineal Unit (ReLU): Transform the inputs by converting negative values to 0, leaving positive values as they are entered.

$$ReLU = f(x) = \max(0, x) = \begin{cases} 0, & \text{when } x < 0 \\ x, & \text{when } x \geq 0 \end{cases} \tag{7}$$

Among some of its characteristics are that they are only activated if the input values are positive as shown in Graphic 3; They are not limited and are recommended for use in imaging and convolutional ANNs. It might be the case that it looks like a linear function, but it is not and allows back propagation.



Graphic 3 Graph of a ReLU function

Source: Own Elaboration

In the event that ReLU contains as inputs a greater number of negative terms or zeroes, then ReLU would produce the output as zero and the ANN could not perform back propagation. This case is known as Dying ReLU. There are other variants that are ReLU based, such as Leaky ReLU, Parametric ReLU, Exponential Linear Unit (ELU), among others.

3. Complex non-linear activation functions: Although most of the functions' adaptations and modifications look for improvements in the processing and the way in which they can be activated, the so-called last generation of ANN (Kim, 2019) seeks to make more extensive use of functions such as the Scaled Exponential Linear Unit (SELU), Softplus or SmoothReLU, Swish, plus a couple of variants of the same Swish represented by the equation:

$$y = x * \text{sigmoid}(x), \text{ replacing} \quad (8)$$

$$y = x * \left(\frac{1}{1 + e^{-x}} \right) = \left(\frac{x}{1 + e^{-x}} \right)$$

The research phase in conjunction with the reports on the suitability and use of Swish (Google, 2021; Ramachandran et al., 2017) indicate better performance with more effective results compared to any other ReLU-based method.

Backward propagation Artificial Neural Networks

Backward propagating ANN is a gradient calculation method for learning; owe their recent success to the fact that they are considered computationally efficient and their fields of application lean towards pattern recognition (Wechsler, 1992). Its learning is supervised type with networks and multilayer perceptrons and with sets of instructions that define all the outputs and the values that are expected. In general, we seek to adjust the weights of each neuron in such a way that the error in the output is minimized because the algorithm indicates the intervention or weighting of each of the neurons within the total calculations; This is done by calculating the cost function partial derivatives with respect to each of the variables.

The method employs a two-phase cycle or propagation. When a value is inserted into the network, it is distributed or propagated from the first layer to the subsequent or hidden layers of the network, until it returns a result. The delivered result is compared with the expected result and an error signal is calculated for each of the outputs. The error outputs propagate backwards, starting from the output layer, towards the neurons in the hidden layer that contributed information to the output; in this way, the neurons in the hidden layers only receive a part of the total error signal, based on the approximate contribution that each neuron has made to the output. This process is cyclical, layer by layer, until all neurons in the network have received an error signal that relatively describes their contribution to Total Error. As the network is trained, the neurons in the hidden layers organize themselves in such a way that the neurons or perceptrons learn to recognize different characteristics of the total input space. After their training, by presenting them with a different input pattern, the neurons in the network hidden layer will respond with an active output if the new input contains a pattern that resembles that characteristic that individual neurons have learned to recognize.

Propagation Rule

The propagation rule determines the weighting resulting from the interaction of neuron i with the N neighboring neurons. The simplest and most widely used propagation rule consists of making a sum of the weighted inputs with their corresponding weights:

$$net_i(t) = \sum_{j=1}^N weight_{ij} * x_j(t) \quad (9)$$

In summary, a backward propagation algorithm has the general equation:

$$w_{ij}(t+1) = w_{ij}(t) + [\alpha * \delta_j^p * x_i^p(t) + \beta \Delta w_{ij}(t)] \quad (10)$$

Being,

$$\delta_j^p = \begin{cases} \delta_j * f'(net_j^p(t)), & \text{if } j \text{ is an output neuron} \\ f'(net_j^p(t)) \sum_{k=1}^M \delta_k^p v_{kj}, & \text{if } j \text{ is a hidden neuron} \end{cases} \quad (11)$$

In its simplest form, the algorithm operation is explained, using Figure 1 as follows:

1. Tickets X, arrive through a pre-connected route
2. The input is modeled using actual weights W. Weights are generally randomly selected.
3. Calculate the output for each neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error of the outputs according to the relation:

$$Error_{propagation} = output_{expected} - output_{obtained}$$

5. Go back from the output layer to the hidden layer to adjust the weights to reduce the error.
6. Keep repeating the process until achieve the desired result.

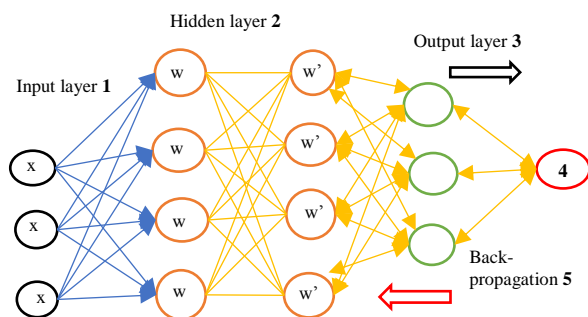


Figure 1 Backward propagation Artificial Neural Network

Source: Own Elaboration

Types of Backward propagation ANNs

There are two types of back-propagation networks:

1. Static Back-propagation: It is a type of back-propagation network that produces a mapping for both a static input and output. It is useful for solving classification problems of the static type such as optical character recognition.

2. Recurrent back-propagation: advances until a fixed value is reached; after that the error is calculated and propagated backwards. When the gradient is negative, the increase in weight decreases the error; if the gradient is positive, if the weight is decreased, the error decreases. Therefore, the algorithms must consider the existence of negative recurrence.

Meteorological phenomenon and numerical models

The Mexican Republic is geographically located in a convergence of climates that give rise to an exchange of warm, humid and polar winds in different magnitudes throughout the year; this air masses collision generates, in a large part of the territory especially on the coasts, gusts of wind with speeds exceeding 20 kilometers per hour with an average duration of 3 hours, which sometimes exceed 72 hours.

In recent decades, numerical models have been developed that have served to explain meteorological phenomena. Among some are the model Fifth-Generation Penn State/NCAR Mesoscale (MM5) and Weather Research and Forecasting (WRF), they are the most used in academic fields, and recently became open source. The models are intended to generate forecasts on various climate variables, such as rain, humidity, rainfall, winds, etc. (Coiffier, 2011; Fierro, 2020; Lorenc, 1986); whose data outputs are in tabular format, but through third-party software or utilities it is possible to generate the corresponding graphics as shown in Figure 2.

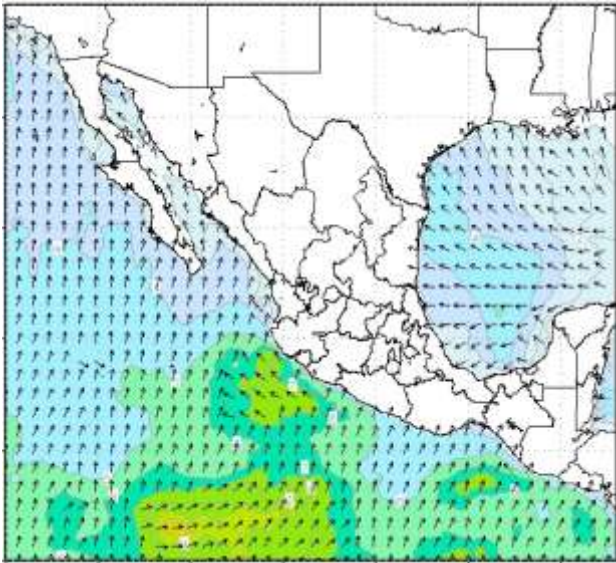


Figure 2 Wind forecast. Image generated by the MM5 model

Source: Own Elaboration

In recent years and thanks to the development of technologies that facilitate computational animation, the information visualization has become an added value for the end user, as shown in Figure 3, computer-generated images have gained attractiveness, but they have become complicated for automated sorting systems.

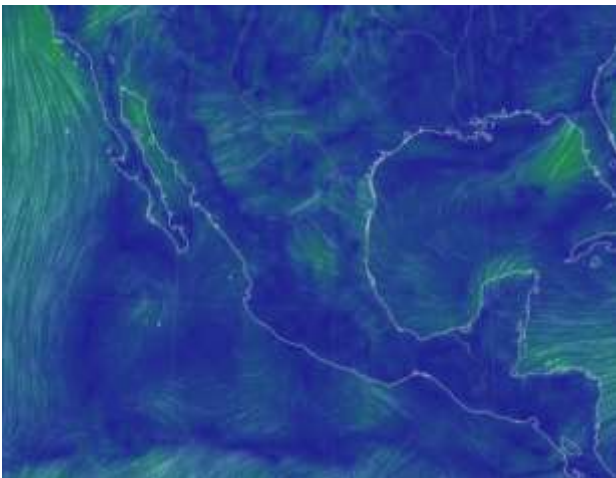


Figure 3 Map screen-shot with real-time movement of the wind variable according to the GFS model

Source: Own Elaboration

Development

The use of Mexican Republic images is proposed, with the same dimension of 640x480 pixels, they are processed by image recognition and it is indicated, depending on the area, if it could be a north event, deep sea, windy or polar event. Once learned, the subsequent forecasts analysis is proposed so that the researchers continue with the ANN training until it becomes automatic.

There is a reserve of 40 images per day from the month of December 2020. That is, it would be expected to feed the ANN as learning with approximately 3,600 images similar to Figure 4 and based on this, that its classification and performance for the new predictions has a greater effectiveness.

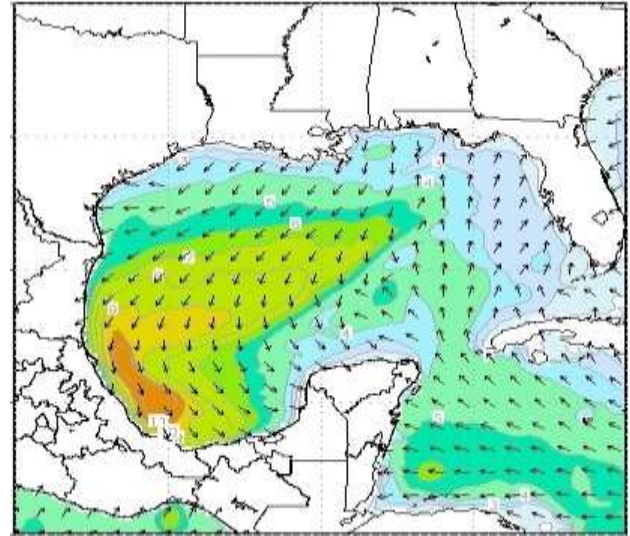


Figure 4 Wind forecast image showing the north wind event generated by the WIII model. This images serve as input for ANN learning

Source: Own Elaboration

Data and groups

In this document only the ingestion of data to the ANN in graphic file format is contemplated since it is one of the typical outputs of the models that generate meteorological variables. It is intended to use the wind speed and direction variable maps to generate a learning that indicates the file (related to time) in which the wind speed and direction exceeds a danger threshold.

Wind references are sought in the Mexican Republic to identify three groups:

- North wind
- South wind
- Atlantic wind

Each group made up of at least 1,200 files with the images corresponding to the wind type that is to be evaluated within the national territory.

Process

The graphic files generated by the WWIII model are required for the execution of the experiment. We worked on the Atlantic Ocean coast due to the fact that there are three types of phenomena with different conditions: a north wind phenomenon; another one of south wind and finally one under normal conditions of Atlantic strong winds with the most significant result for the realization of the modeling.

Once the maps were downloaded, the TeachableMachine tool was used, which is enabled with the Swish activation function and is part of the Tensorflow libraries for JavaScript. Learning models base their performance on the time it takes to show a response and the quality of it, not on the time it takes to generate their learning.

Three classes were created: North, South and Normal; corresponding to each of the meteorological phenomena that occur in the Atlantic area. Once the images were loaded, the training parameters of the model were established as shown in Table 1.

Wind type	Number of images loaded
North	1,738
South	1,094
Normal	953

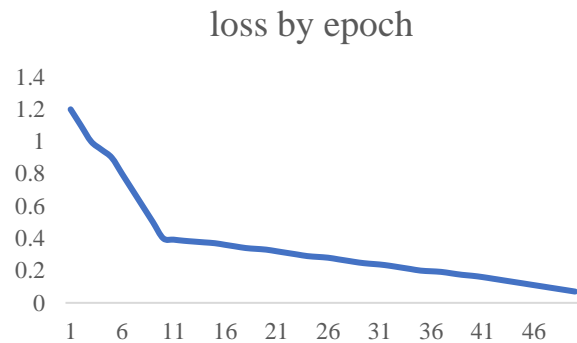
Table 1 Number of images loaded into ANN

Source: Own Elaboration

As part of the parameters, the size of the epoch or the number of times the data set would be processed was defined, it was established at 50. Batch, which corresponds to the number of elements that belong to a batch of elements that may or may not belong to the same classification, and whose proposed value is 16 and, finally, the learning rate at 0.001, which is the value proposed by developers. Teachablemachine divided the sample into two segments:

1. Preparation samples, which are 85% of the total samples and are used to train the model; epoch take the data from this sample.
2. Test samples, which correspond to the remaining 15% and are used to check the model performance.

Once the images were inserted, the ANN learning was carried out; to assess the learning degree, a confusion matrix is used showing the class of the samples on the y-axis and the x-axis the prediction or the class to which the model assigns the samples after learning. Loss measures the level of learning of a model when predicting the correct classification of a given set of samples. A good sampling should aim for zero, as shown in Graphic 4.



Graphic 4 Learning assess

Source: Own Elaboration

Another quality value is the precision by epoch, which is the percentage of classifications that a model hits during preparation. If the predictions are correct, the graph tends to one; otherwise it will be lower.

Results

Once the model was trained, the recognition performed by the ANN was tested with a batch of 300 images from different days and with different phenomena; with adequate results greater than 80%; which are indicated as shown in Figure 5.

A higher error rate was observed in situations where the maps of the phenomena converge in areas of the southern basin of Veracruz, that is, towards the coordinates with latitude 20°N and longitude 96°W. When detailing the images we find that the phenomena differ by the wind direction that originates them in previous days; resulting in classification errors. Hurricanes are another meteorological phenomenon that should be considered because they are frequent in the summer and autumn seasons; these generate gusts of wind in both directions that could increase the size of the expected error according to the confusion matrix.

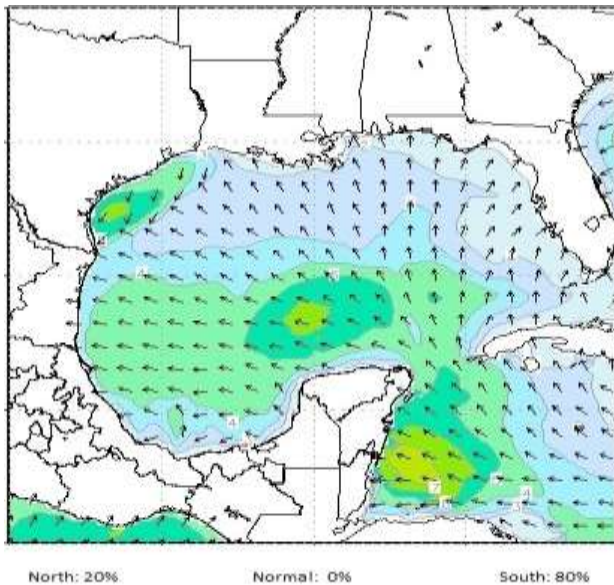


Figure 5 Result of the evaluation of an image by the trained ANN

Source: Own Elaboration

Conclusions

For an image that presented both the south and the north phenomena, the recognition was favored towards the north phenomenon, which was probably due to the fact that the sample of the north wind phenomenon is greater than that of the south wind phenomenon and, on the other hand, was a predicted error in the confusion matrix; for this reason, it is necessary to create historical records of the models executions until achieving a number of training images in similar quantities and achieving a balanced learning.

Also, it is necessary to use at least the wind direction variable to eliminate ambiguities in the decision, so the use of control vectors to aid recognition will be considered. The foregoing because the vectors available do not provide enough information to improve the ranking. As well as providing an additional value to the network that gives it a decisions memory that were made about the events in previous time (t-n) as one more element of ANN learning. Within the memory that should be considered is that of large meteorological phenomena that affect the forecast or recognition of the patterns already established, due to their similarity, but that have different origins and therefore different effects.

Acknowledgments

To Consejo Nacional de Ciencia y Tecnología (CONACYT) México.

References

- A.I. Wiki. (2021). Pathmind, Inc. <https://wiki.pathmind.com/neural-network>
- BBVA. (2019). *Machine learning: What is it and how does it work?* BBVA Artificial Intelligence. <https://www.bbva.com/en/science-technology/technologies/artificial-intelligence/>
- Coiffier, J. (2011). *Fundamentals of numerical weather prediction*. Cambridge University Press. www.cambridge.org/9781107001039
- Fierro, A. A. (2020). *Predicción de Series Temporales con Redes Neuronales*. Universidad Nacional de La Plata.
- Google, I. (2021). *Teachable Machine (2.4.1)*. <https://teachablemachine.withgoogle.com/>
- Kim, K. G. (2019). Deep learning book review. *Nature*, 29(7553), 1–73.
- Li, Y., & Yuan, Y. (2017). Convergence Analysis of Two-layer Neural Networks with ReLU Activation. *Advances in Neural Information Processing Systems, 2017-Decem*, 598–608. <http://arxiv.org/abs/1705.09886>
- Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474), 1177–1194.
- Mathworks. (2020). Machine Learning with MATLAB. In *MathWorks*. <https://uk.mathworks.com/solutions/machine-learning.html>
- Ramachandran, P., Zoph, B., & Le, Q. V. (2017). *Searching for Activation Functions*. <http://arxiv.org/abs/1710.05941>
- Wechsler, H. (Ed.). (1992). *Neural Networks for Perception*. Elsevier. <https://doi.org/10.1016/C2013-0-11676-5>