Neural Networks

Borja Jiménez, Marcos García

Jaume I University, Spain

al361929@uji.es
al375909@uji.es

***Abstract*— We have analysed the basics of a neural network starting with a comparison of a biological neuron and how it has been adapted to a computing environment and becoming one of the most powerful tools in fields like classification or prediction nowadays.**

***Keywords*— artificial intelligence, convergence, neural network, perceptron, training.**

I. Introduction

Artificial intelligence has become a very popular topic nowadays although at the same time it does not seem a concept related to daily life, but it is present in plenty of areas of our society. For instance optimisation in software for cameras in smartphones, automatization of process replacing jobs that were hitherto limited to humans and so on.

There is no agreed definition of artificial intelligence by science, there is not even a definition of what it calls a system as intelligent. For this reason, the field of study of artificial intelligence is characterised by being an experimental and highly changeable science.

In its simplest form, artificial intelligence is the attempt to mimic human intelligence using a robot, or a software. But it is a very vague concept because there are many ramifications. The experts in artificial intelligence Stuart Russell and Peter Norvig have differentiated it into four types: Systems that act like humans, like robots; systems that use rational logic, such as expert systems; systems that act rationally, such as intelligent agents and systems that think like humans, such as artificial neural networks [1].

For example, Tesla automatic cars use this technology. These cars are composed of a lot of cameras and sensors that identify all kinds of elements from the road, adapting the car to the type of driving. The whole system behind all this is just an artificial neural network trained to perform that task.

Precisely, this type of technology is used to perform the optimization of existing algorithms and to develop new algorithms impossible to program in a conventional way such as the classification of images or virtual assistants due to the necessity of programm each individual situation where a neural network is capable of generalize and predict a proper solution.

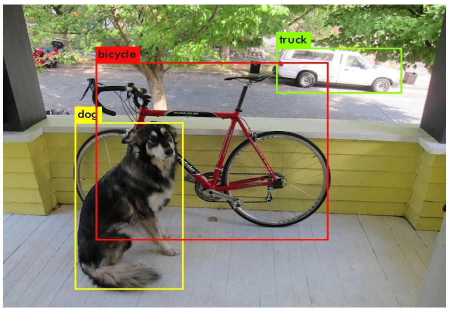

Figure. 1 Example of a recognition system

The remaining sections of this paper are as follows:

Section 2, 3, 4 and 5 introduce basic concepts of the components of a neural network to understand the functioning.

Section 6 shows how the experiments have been performed and what methods and materials were used.

Section 7 describes the experiments made with two models of neural networks.

Section 8 ends the article with some conclusions about the study.

II. Neuron

In order to understand how an artificial neuron works and how it can form more complex structures such as neural networks, it is necessary to understand the basis of this technology: biological neurons.

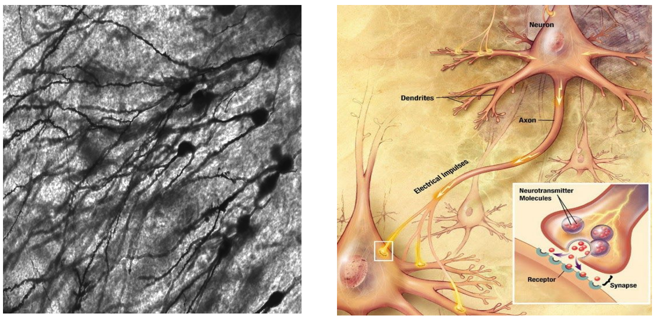

Figure. 2 Illustration of real neuron components

A neuron, or nerve cell, is an electrically excitable cell that communicates with other cells via specialized connections called synapses. The typical neuron consists of a cell body, denoted as soma, dendrites, and a single axon.

The axon and dendrites are the neuron endings, those are responsible for the transmission of information between neurons, codified in electrical pulses.

A single neuron can be connected to different neurons via multiple dendrites which receive the electrical pulses. Once the pulses are received in the soma, the neuron manipulates the signal through both electrical and chemical processes.

Due the neurons are electrically excitable, if the voltage of the signal received changes suffers a large variation in a short interval of time, the neuron generates a pulse called an action potential which is sent to the next neurons connected to its axon [2].

III. Artificial Neuron

To approach the functioning of a biological neuron in a computational environment, most of the main components of a neuron have been copied or adapted, resulting in a mathematical model denoted as perceptron. A perceptron is a basic unit of inference in the form of a linear discriminator, from which an algorithm is capable of generating a criterion for clustering capable of solving classification problems.

IV. Perceptron

Following the architecture of a biological neuron in this approach, each neuron is connected to others through weighted links working as inputs. Through those likes the output of the previous neurons are sent, and the values of the outputs are multiplied by the weights of the links. These weights can increment or cancel the activation state of the next neurons. They will be modified by the neuron representing the importance of those links in the training part as we will see later. The weights would have a higher value if it helps to determine an accurate output [3]-[5].

Once the information has arrived to the body of the neuron, the weights are sum together with a determined constant denoted as bias. The bias value, allows the activation function to be shifted to the left or right, to better fit the data and it influences the output values [5]. If the sum is higher than some determined threshold, the step function is triggered. The output of the function will be sent through the axon, the link to the next neuron, to the dendrites of the next neurons and the process will begin again in the next neuron until it arrives at the last one.

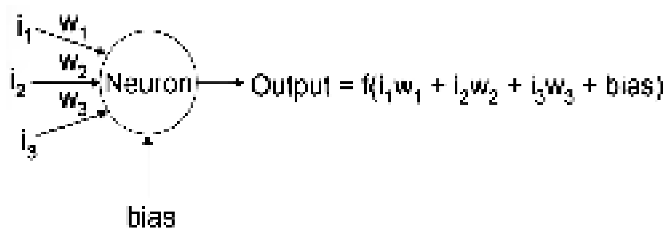

Figure. 3 Diagram of a perceptron being i_n the previous neurons, w_x the values of the i_x weigh, and the bias constant.

V. Artificial Neural Network (ANN's)

The artificial neural networks are a set of units called artificial neurons, which are linked between them to send signals. The input data goes through the neural network where it suffers various operations, producing a specific output data.

Once we know what the neural networks are, we have to differentiate two types: feedforward and recurrent.

The main difference between these two types, is how the data flows through the network. In the feedforward network, the data flows in only one direction from the input to output. Whereas in the recurrent neural network, the data can flow in multiple directions.

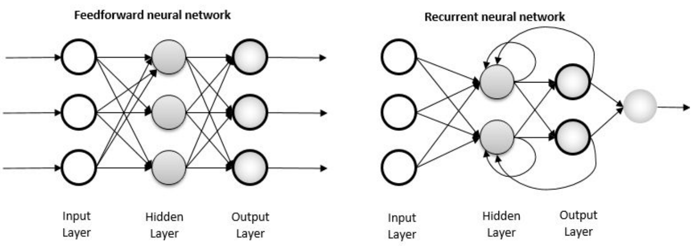

Figure. 4 Feedforward representation on the left and Recurrent neural networks representations on the right

But, how do neural networks learn? We are going to see two different types of learning, supervised and unsupervised.

A. Supervised Learning

In this learning, we have to teach the network. For that, some example data (inputs and desired outputs) is available for the network and it tries to adjust the weights with the objective to obtain a good prediction. In this paper, we will use this training for our network.

The supervised learning is classified into two categories of algorithms: classification and regression.

“A classification problem is when the output variable is a category, such as “Red” or “Blue” or “disease” and “no disease”.”

“A regression problem is when the output variable is a real value, such as “dollars” or “weight”.” [7].

B. Unsupervised Learning

For this training we only supply the network with the desired outputs and like in the supervised training, the network adjusts the weights to obtain a good prediction.

The unsupervised learning is classified into two categories of algorithms: clustering and association.

“A clustering problem is when you want to discover the different groups in your data, such as grouping customers by purchasing behavior.”

“An association problem is when you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.” [7].

So, one important term in the neural network performance is the overfitting, and appears in the training part of the model. The error on the training will have a very low value, but when new data is shown to the neural network, the error in the testing will be very high. This effect is produced when the model does not have the ability to generalize because it has only memorized the data provided in the training stage [8].

VI. Material and methods

We carry on experiments with artificial neural networks to understand the functioning and what kind of problems they are capable of solving. In the first experiment, we use a dataset to represent two different types of dots and the objective is to test if the neural network is able to distinguish the different kinds of dots and separate them. In the second experiment, we used a more advanced artificial neural network to try to make a classification of handwritten numbers or traffic signals.

In the first experiment a Perceptron algorithm was used to solve independent linear problems. In order to solve independent but non-linear problems, a more advanced Perceptron model called Multilayer Perceptron (MPL) was used.

To develop these experiments we use a software called Anaconda with the Jupyter Notebook technologies. Jupyter Notebook is an app that allows the edition and execution of code from different languages through the web. The app allows the representation of graphs and tables to facilitate the development of the experiments.

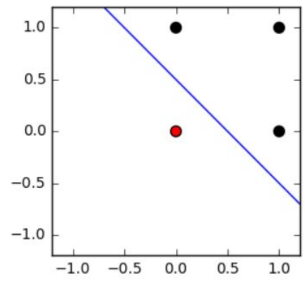

Figure. 5 Perceptron convergences splitting the data into red and black dots

VII. Experimental Exercises

In this section, we will expose the exercises we were asked to solve in the practical section of the course. Following the flow of the theoretical sessions, the first part consisted of the use of a single perceptron as the only tool to solve classification problems. Afterwards, we had to extrapolate this knowledge to a neural network made up of several perceptrons denoted as MLP.

A. Perceptron

In order to start working with a perceptron, it is necessary to develop a process denoted as training. In the practical classes, we were asked to develop a model of a single perceptron able to classify linear separable sets of data. This data was given as vectors of integers, the first one is a tuple representing coordinates of a point in the space of a plot, and the second vector represents the values of labels of those points.

Once the artificial neurons have the labelled data, it is used to modify the weights. In each iteration, the perceptron grabs a pair of data, a point and his value. The neuron produces a value and compares it with the true label. If the prediction is correct the link's weight will be changed to a higher value, otherwise if the prediction fails the link will be penalized resulting in a lower value of the weight. For this reason, the process of training can be summed up in the overall modification of the weights.

At the end of the iterations, the perceptron algorithm is guaranteed to converge on some solution while using linearly separable training data, drawing in the plot a line that represents the border of each label.

B. Multi Layer Perceptron (MLP)

A model implemented with only one perceptron can not classify the data in more than 2 classes. For this reason using more than one neuron allows us to produce a higher number of classification labels and it is capable of solving linearly non-separable problems. A basic example of the new feature is when we tried to represent a XOR function. With only one perceptron in this problem, it would have never converged on a proper solution due to the necessity of two decision boundaries.

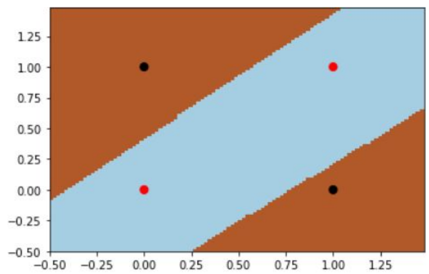

Figure. 6 XOR function with MLP

Using this premise, new problems can be solved now. A common example of the power of this tool is the classification and recognition of pictures. The first exercise related to this problem that we had to solve was the recognition of manuscript numbers.

The use of a more complex model, brings the possibility to tweak some of the given parameters of the neural network in order to perform better. There is not a universal set of parameters that will guarantee the best score, for this reason, some tests are needed. The number of neurons and the layers are the most important parameters. Below, in Figure 7 and Figure 8, we will see their configuration and how they affect the overall performance of the model.

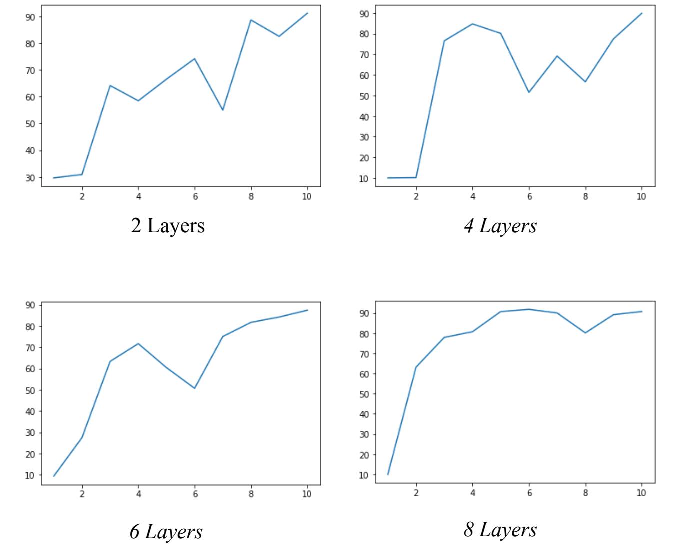

Figure. 7 Number of neurons (X axis) vs score (Y axis) with different number of layers

To obtain the best parameters we developed a double nested loop. The first one iterates over the number of layers and the second one over the numbers of neurons. Fig. 7 shows the evolution of the iterations.and we observed that the best score with a lower number of a neuron is achieved with 8 layers and 5 neurons, after this number, it is not worth keeping increasing it due to the lower variation of the score. As we have explained in section Section V, to train a neural network capable of developing a classification algorithm, it is necessary to use supervised training with labeled data. For this purpose, we used a dataset consisting of pictures of the manuscript numbers and the value that the picture represents. Once we have the data, it is necessary to split it in order to obtain the training set and the testing set, in a process denoted as cross-validation.

For the last exercise we generated a model capable of classifying a huge dataset of traffic signs giving as output the label of the class related to the sign. Following the same steps as in the previous exercise, we develop a loop that iterates over the number of neurons in order to find the optimal parameters to train the model. Fig. 8 shows that the optimal value for the number of neurons is five.

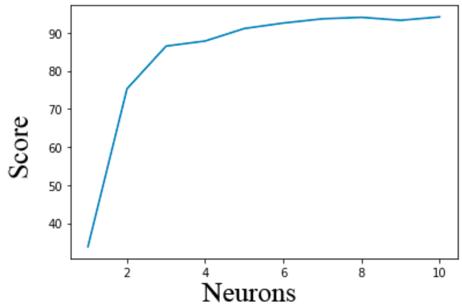

Figure. 8 Evolution of the score over the number of neurons with one layer

VIII. Conclusion

Analyzing the output given by the perceptron in section VII-A , we can conclude that the use of a single perceptron carries certain limitations. First using only one neuron the output's range can only have a maximum of two different values (0 or 1). Second, in order to achieve the perceptron to converge the use of linear data is needed. For this reason, in order to work with classification problems, where more than two groups are needed, it is necessary to create neural networks where more than one perceptron is involved. This kind of layout, considering a network of two perceptrons, will produce two decision boundaries classifying the input data into four categories as we exposed in the XOR exercise. Finally, Fig. 7 and Fig. 8 show that the correct layout of a network is not always using the maximum number of resources like neurons or layers to reach the best score. For instance, in Fig. 8 increasing over 5 neurons is not worth having a higher computational cost. Once we have set the optimal parameters the neural network becomes a very functional tool capable of solving complex problems with a very low error rate.

References

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 3rd ed. 2009.*
[2] https://en.wikipedia.org/wiki/Neuron
[3] https://en.wikipedia.org/wiki/Artificial_neuron
[4] https://en.wikipedia.org/wiki/Perceptron
[5] https://es.wikipedia.org/wiki/Perceptrón
[6] https://medium.com/deeper-learning/bias
[7] https://www.geeksforgeeks.org/supervised-unsupervised-learning/
[8] https://es.mathworks.com/help/deeplearning/ug/overfiting.html
[9] http://matlab.izmiran.ru/help/toolbox/PercepLimitations&Cautions