UNIVERSIDAD POLITÉCNICA DE YUCATÁN

COMPUTATIONAL ROBOTICS ENGINEERING

GROUP: 9° “B”

PROFESSOR: VICTOR ORTIZ

SUBJECT: MACHINE LEARNING

STUDENT: JOSEPH JESUS AGUILAR RODRIGUEZ

Table of Contents

Explanation of the intuition behind the Perceptron algorithm.

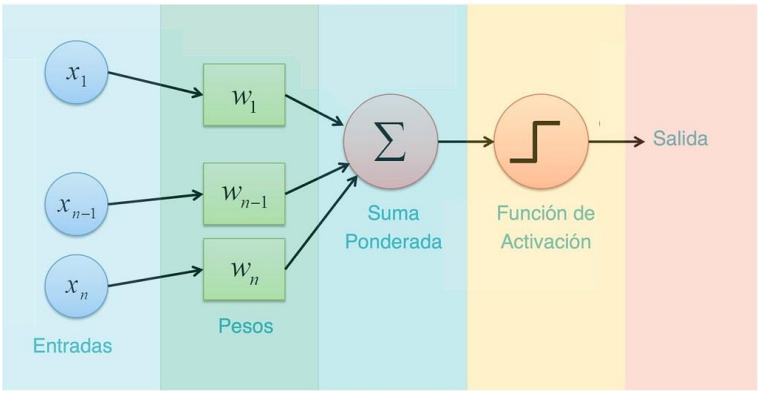
A perceptron is a mathematical model inspired by a simplified structure and function of a single biological neuron. It is a section of Machine Learning that is used to understand the concept of binary classifiers.

It is part of the neural network system. In fact, it can be said that the perceptron and neural networks are interconnected. The perceptron constitutes the basic foundation of the neural network that is part of Deep Learning. It is considered as building blocks of a single layer of the neural network.

A neural network formed by the perceptron can be defined as a complex statement with a very deep understanding of logical equations. A neural statement that follows the perceptron is either true or false, but it can never be both at the same time.

The perceptron is made up of a series of components such as:

- Entrance. The inputs in the perceptron algorithm are understood as x1, x2, x3, x4 and so on. All these inputs denote the feature perceptron values and the total occurrence of the features.
- Pesos. They are observed as values that are planned throughout the perceptron preparation session. The weights offer a preliminary value at the beginning of learning the algorithm. With the occurrence of each training inaccuracy, the weight values are updated. These are mainly represented as w1, w2, w3, w4 and so on.
- Activation function. Each activation function, or nonlinear function, takes a single number and performs a certain fixed mathematical operation on it. There are several activation functions that can be found in practice, the most common are the Sigmoid or the ReLU or rectified linear unit.
- Weighted sum. It is the proliferation of each input value or feature associated with the corresponding step value.
- Exit. The weighted sum is passed to the activation function and whatever value we get after the calculation is our predicted output.



One-layer perceptron model

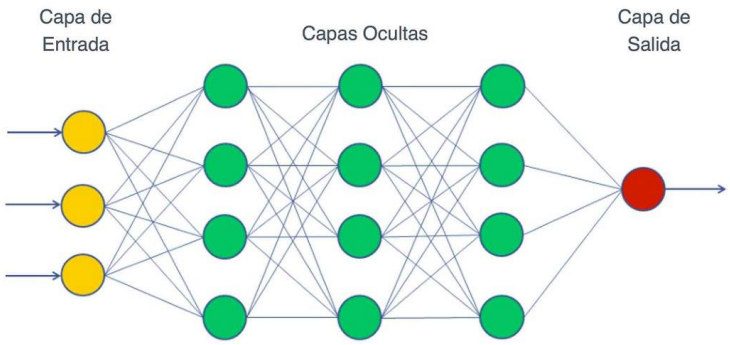
A single-layer perceptron model includes a feedforward network that depends on a threshold transfer function in its model. It is the simplest type of artificial neural network that can analyze only linearly separable objects with binary outputs, i.e. 1 and 0.

If we talk about the functioning of the one-layer perceptron model, its algorithm has no prior information, so initially, the weights are assigned inconsistently, then the algorithm adds all the weighted inputs, if the added value is more than some value default or threshold value, then a layer's perceptron is declared active and returns the output as +1.

In simple words, multiple input values feed the perceptron model, the model is run with the input values, and if the estimated value is the same as the required output, then the performance of the model is satisfied, so The weights do not require changes. In fact, if the model does not meet the required result, then some changes are made to the weights to minimize the errors.

Multilayer perceptron model

A multi-layer perceptron model has a similar structure to that of a single-layer perceptron model with more number of hidden layers. Also called backpropagation algorithm. It is executed in two stages: the forward stage and the backward stage.



In the forward stage, the activation functions originate from the input layer to the output layer, and in the backward stage, the error between the observed actual value and the demanded value originates backward in the output layer. to modify the weights.

In simple terms, the multilayer perceptron can be treated as a network of numerous artificial neurons on various layers, the activation function is no longer linear, instead non-linear activation functions such as sigmoid functions, TanH, ReLU activation functions are deployed. , among others, for its execution.

Perceptron models are the most simplistic type of neural network in that they take an input, the weight of each input, take the sum of the weighted input and apply an activation function.

They accept and construct only binary values, that is, the perceptron is only implemented for binary classification with the limitation that they are only applicable for linearly separable objects. (González, 2021)

Algorithm pseudocode

Función Perceptron:

Inicialización de pesos y bias

Para cada peso w:

w = valor aleatorio pequeño

bias = valor aleatorio pequeño

Tasa de aprendizaje

tasa_aprendizaje = un valor pequeño positivo

Número máximo de iteraciones (opcional)

max_iteraciones = un número máximo de iteraciones

Datos de entrenamiento (X: entradas, y: etiquetas)

Datos de entrenamiento (X, y)

Inicio del entrenamiento

iteracion = 0

Mientras iteracion < max_iteraciones (o hasta que no haya errores):

error_total = 0

Para cada ejemplo en los datos de entrenamiento:

```
        entrada = X[i]
        etiqueta_real = y[i]

        # Calcular el valor predicho
        valor_predicho = función_activación(entrada, pesos, bias)

        # Calcular el error
        error = etiqueta_real - valor_predicho

        # Actualizar pesos y bias
        Para cada peso w:
            w = w + tasa_aprendizaje * error * entrada

        bias = bias + tasa_aprendizaje * error

        error_total = error_total + error

    iteracion = iteracion + 1

Fin Mientras

# Devolver los pesos y bias entrenados
Devolver pesos, bias

Función función_activación(entrada, pesos, bias):
    # Sumar el producto de entradas y pesos, y agregar el bias
    valor = suma(entrada * pesos) + bias

    # Aplicar la función de activación (por ejemplo, función escalón)
```

```
    Si valor > 0:
        Devolver 1
    Sino:
        Devolver 0

Función suma(valores):
    # Sumar todos los valores en la lista
    suma = 0
    Para cada valor en valores:
        suma = suma + valor

    Devolver suma

# Uso del perceptrón
pesos, bias = Perceptron()
```

Loss function + Optimization function identification.

Perceptrons are a simple supervised learning technique that can be considered a precursor to modern neural networks, but their basic formulation does not include a loss or optimization function. Instead, it relies on simple error correction criteria to adjust weights and biases during training. However, in the context of more advanced neural networks, such as artificial neural networks (ANN), loss functions are used to quantify how well a network performs on a classification or regression task. Additionally, we apply optimization algorithms such as gradient descent to minimize the loss function and adjust the network weights and biases to reduce prediction errors.

References

González, L. (5 de october de 2021). *aprendeia*. Obtenido de aprendeia: https://aprendeia.com/que-es-el-perceptron-simple-y-multicapa/