

PERCEPTRON ALGORITHM

LAB 3

Perceptron Algorithm

- The Perceptron algorithm is a **two-class (binary) classification** machine learning algorithm.
- It is a type of neural network model, perhaps **the simplest type of neural network** model.
- It consists of a **single node or neuron that takes a row of data as input and predicts a class label**.
- This is achieved by calculating the weighted sum of the inputs and a bias (set to 1). The weighted sum of the input of the model is called the **activation**.
- **Activation** = Weights * Inputs + Bias
- If the activation is above 0.0, the model will output 1.0; otherwise, it will output 0.0.
- **Predict 1:** If Activation > 0.0
- **Predict 0:** If Activation <= 0.0

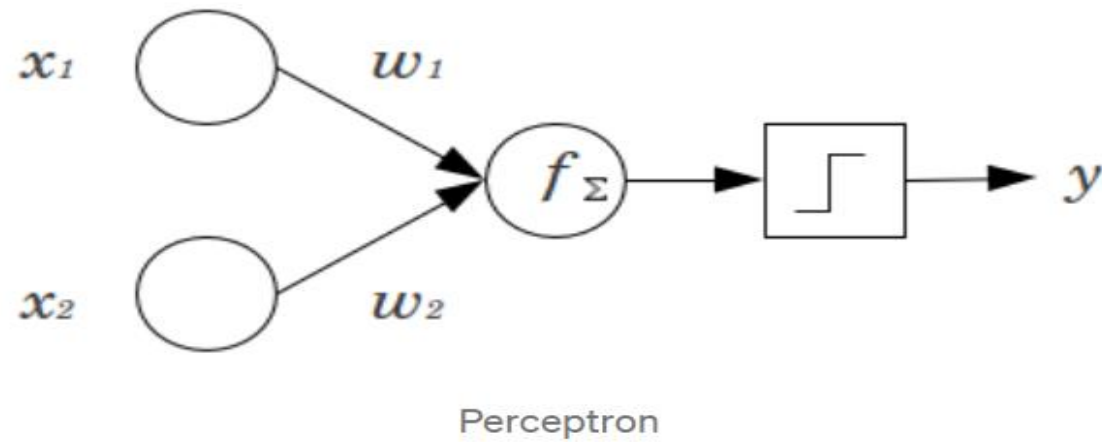
Training the Model

- The Perceptron is a **linear classification algorithm**. This means that it learns a decision boundary that separates two classes using a line (called a hyperplane) in the feature space.
- The coefficients of the model are referred to as input weights and are trained using the **stochastic gradient descent** optimization algorithm.
- Examples from the training dataset are shown to the model one at a time, the model **makes a prediction, and error is calculated**.
- The weights of the model are then updated to reduce the errors for the example. This is called **the Perceptron update rule**.
- This process is repeated for all examples in the training dataset, called an **epoch**.
- This process of updating the model using examples is then **repeated for many epochs**.

Hyperparameters

- Model weights are updated with a small proportion of the error each batch, and the proportion is controlled by a hyperparameter called the **learning rate**, typically set to a small random value .
- **Training is stopped when** the error made by the model falls to a low level or no longer improves, or a maximum number of epochs is performed.
- The other hyperparameter is the **number of training epochs**.

Perceptron Model



- x_1, x_2 - inputs
- w_1, w_2 - weights
- f - activation function
- Y -output

Demo

<https://mlweb.loria.fr/book/en/perceptron.html>