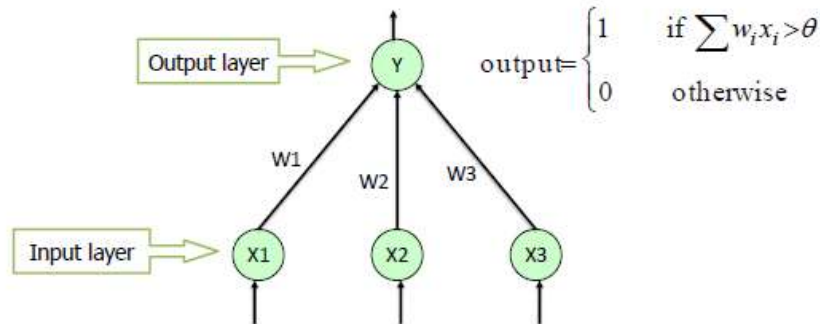


[Map](#) > [Data Science](#) > [Predicting the Future](#) > [Modeling](#) > [Classification/Regression](#) > [Artificial Neural Network](#) > Perceptron

## Artificial Neural Network - Perceptron

A single layer perceptron (SLP) is a feed-forward network based on a threshold transfer function. SLP is the simplest type of artificial neural networks and can only classify linearly separable cases with a binary target (1, 0).

### Single Layer Perceptron



### Algorithm

The single layer perceptron does not have a priori knowledge, so the initial weights are assigned randomly. SLP sums all the weighted inputs and if the sum is above the threshold (some predetermined value), SLP is said to be activated (output=1).

$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n > \theta \quad \text{Output} \quad \rightarrow \quad 1$$

$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n \leq \theta \quad \rightarrow \quad 0$$

The input values are presented to the perceptron, and if the predicted output is the same as the desired output, then the performance is considered satisfactory and no changes to the weights are made. However, if the output does not match the desired output, then the weights need to be changed to reduce the error.

### Perceptron Weights Adjustment

$$\Delta w = \eta \times d \times x$$

$d \rightarrow$  Predicted output - Desired output

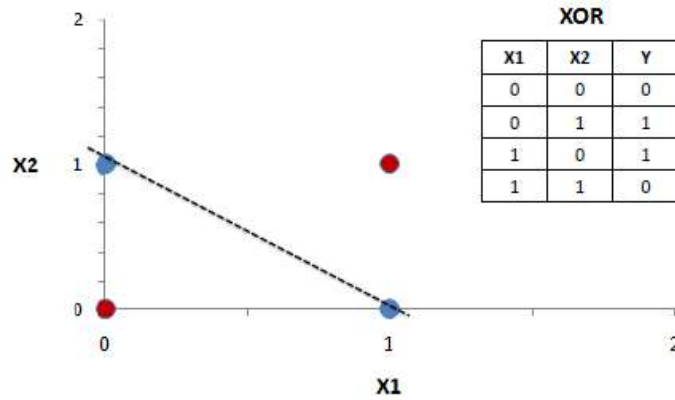
$\eta \rightarrow$  Learning rate, usually less than 1

$x \rightarrow$  Input data

Because SLP is a linear classifier and if the cases are not linearly separable the learning process will never reach a point where all the cases are classified properly. The most famous example of the inability of perceptron to solve problems with linearly non-separable cases is the XOR problem.

ads vi

Use m  
Intellig  
improve  
and boo



ads via

Use m  
Intellig  
improve  
and boo

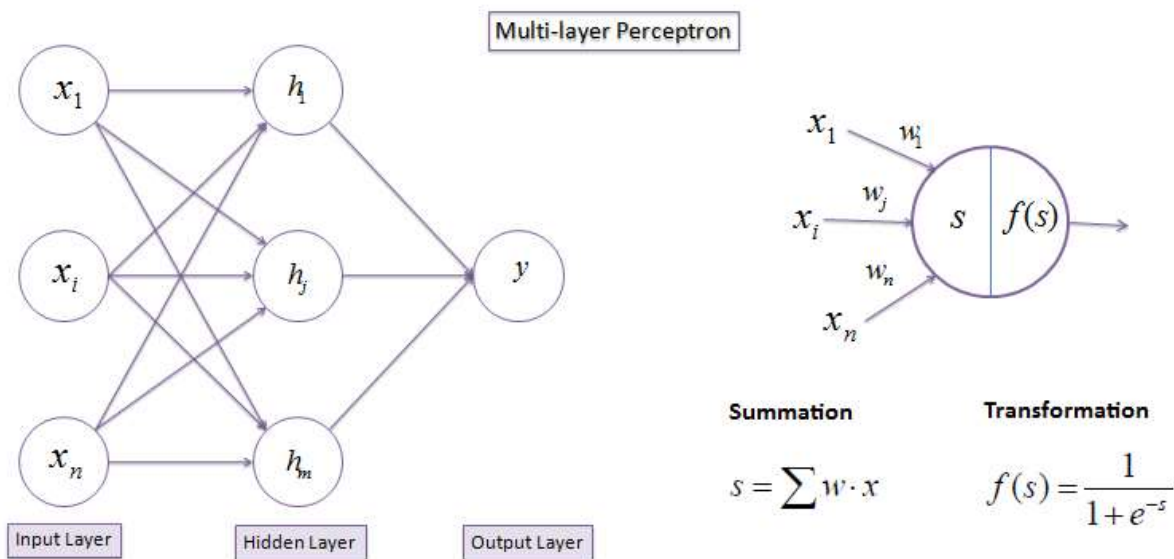
However, a multi-layer perceptron using the backpropagation algorithm can successfully classify the XOR data.

### Multi-layer Perceptron - Backpropagation algorithm

A multi-layer perceptron (MLP) has the same structure of a single layer perceptron with one or more hidden layers. The backpropagation algorithm consists of two phases: the forward phase where the activations are propagated from the input to the output layer, and the backward phase, where the error between the observed actual and the requested nominal value in the output layer is propagated backwards in order to modify the weights and bias values.

#### Forward propagation:

Propagate inputs by adding all the weighted inputs and then computing outputs using sigmoid threshold.



#### Backward propagation:

Propagates the errors backward by apportioning them to each unit according to the amount of this error the unit is responsible for.

1. Error in any **output** neuron

$$d_o = y \times (1 - y) \times (t - y)$$

2. Error in any **hidden** neuron

$$d_i = y_i \times (1 - y_i) \times (w_i \times d_o)$$

3. Change the **weights**

$$\Delta w = \eta \times d \times x$$


Example:

[A Neural Network in 11 lines of Python](https://www.saedsayad.com/artificial_neural_network_bkp.htm)

[ANN Interactive](#) (only Windows)

[Exercise](#)



 ads vi

Use m  
Intellig  
improve c  
and boo