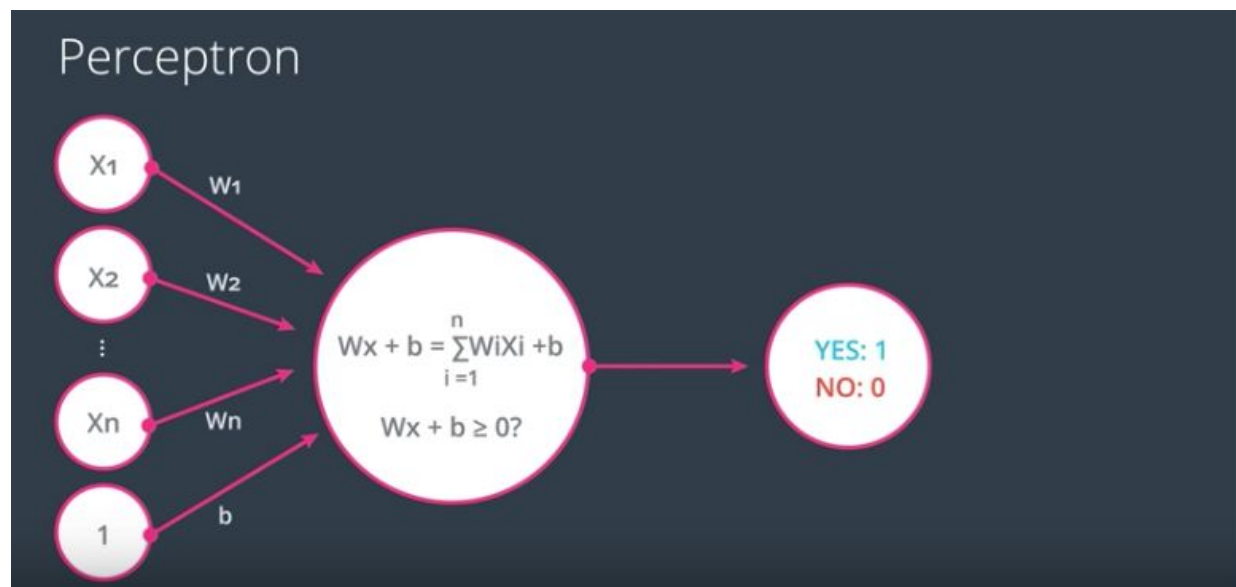


## Neural Networks (NNs)

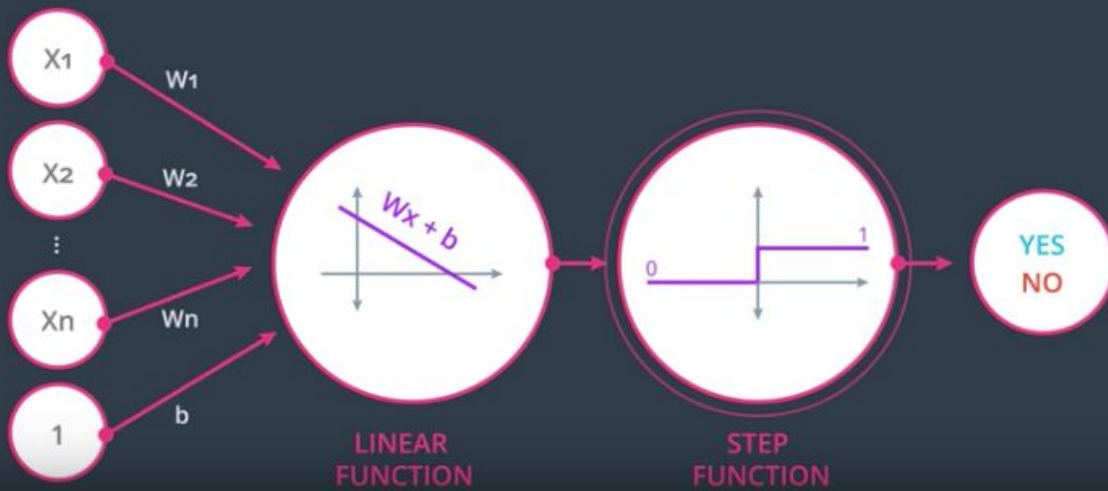
analogous to linear regression: ([video](#))



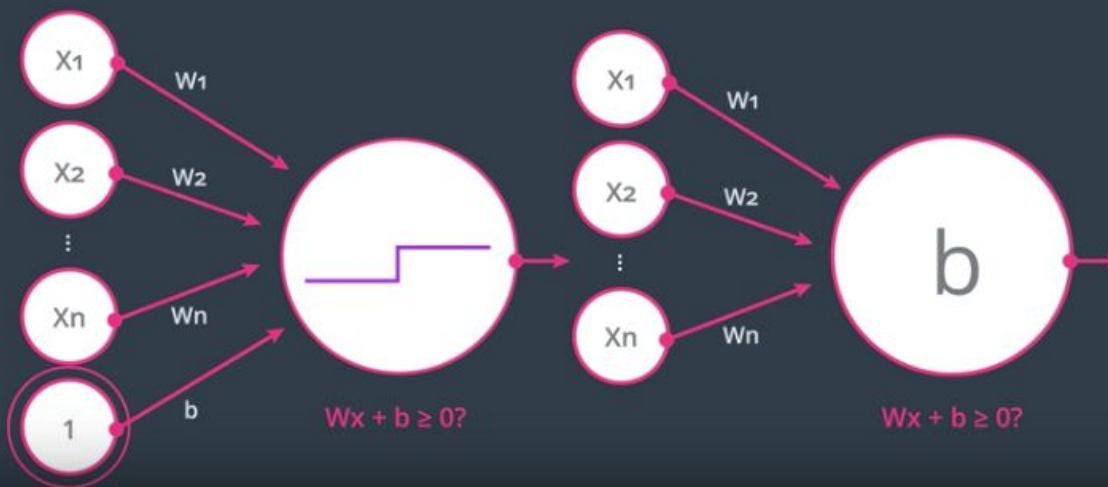
as NN:



# Perceptron

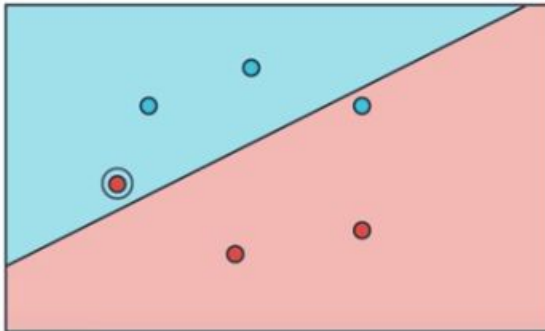


# Perceptron



## Perceptron algorithm

### Perceptron Algorithm



1. Start with random weights:  $w_1, \dots, w_n, b$

2. For every misclassified point  $(x_1, \dots, x_n)$ :

2.1. If **prediction = 0**:

- For  $i = 1 \dots n$
- Change  $w_i + \alpha x_i$
- Change  $b$  to  $b + \alpha$

2.2. If **prediction = 1**:

- For  $i = 1 \dots n$
- Change  $w_i - \alpha x_i$
- Change  $b$  to  $b - \alpha$

[video](#)

nonlinear models ([video](#))





## Perceptron vs Gradient Descent

### GRADIENT DESCENT ALGORITHM:

Change  $w_i$  to  $w_i + \alpha(y - \hat{y})x_i$

### PERCEPTRON ALGORITHM:

If  $x$  is misclassified:

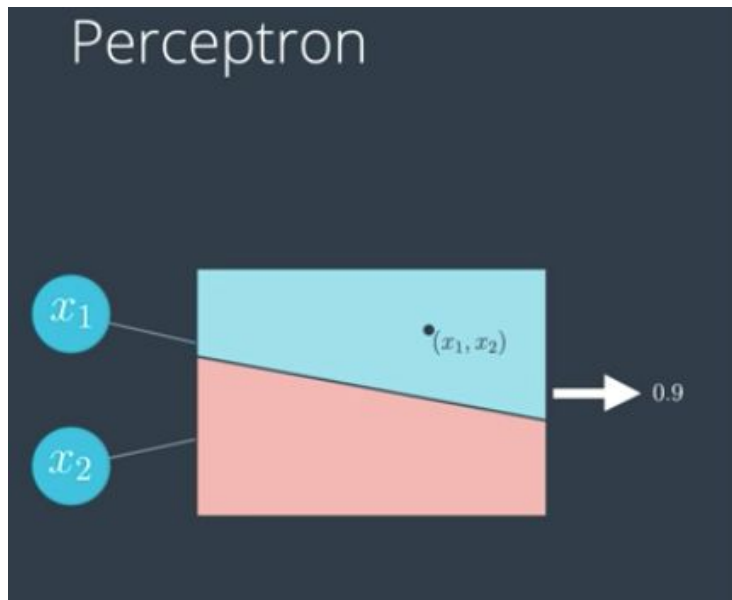
Change  $w_i$  to  $\begin{cases} w_i + \alpha x_i & \text{if positive} \\ w_i - \alpha x_i & \text{if negative} \end{cases}$

If correctly classified:  $y - \hat{y} = 0$

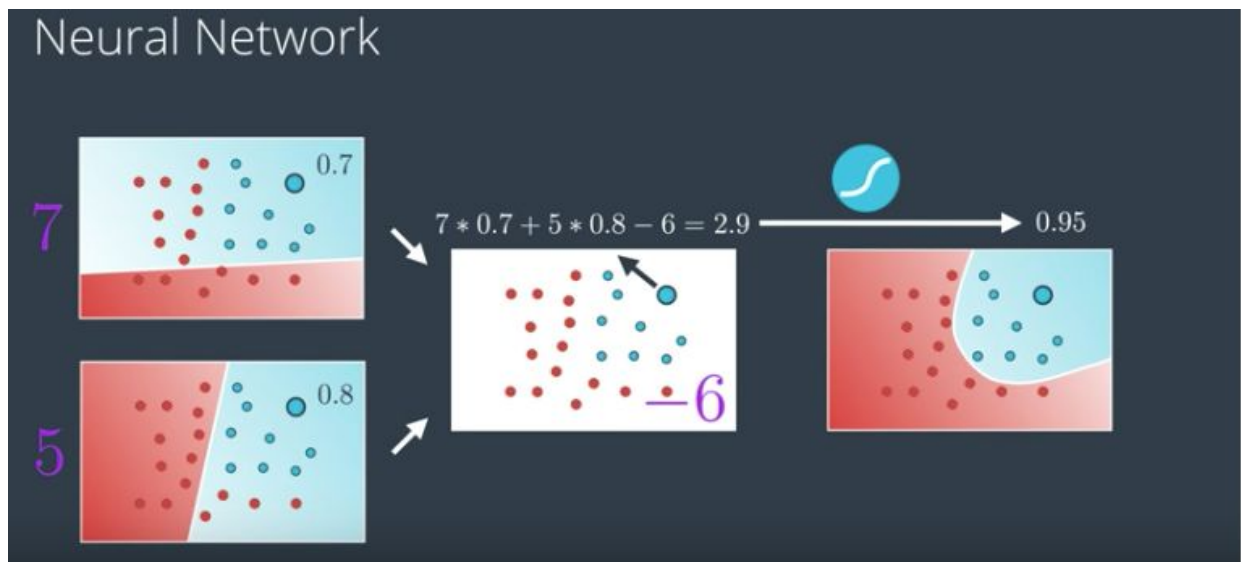
If misclassified:  $\begin{cases} y - \hat{y} = 1 & \text{if positive} \\ y - \hat{y} = -1 & \text{if negative} \end{cases}$

both algorithms are basically the same!

continuous perceptron:

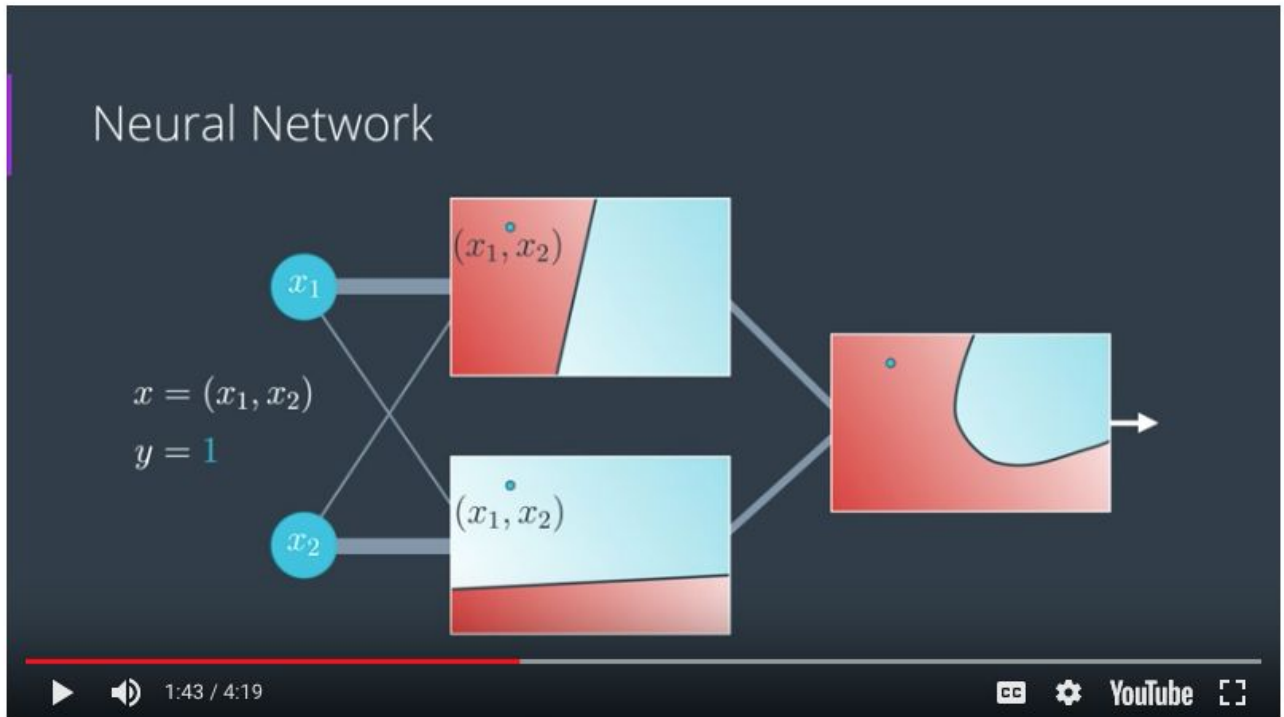


NN architecture: ([video](#))



## Feedforward

Feedforward is the process neural networks use to turn the input into an output. Let's study it more carefully, before we dive into how to train the networks.



[video](#)

Backpropagation explanation: [video](#)