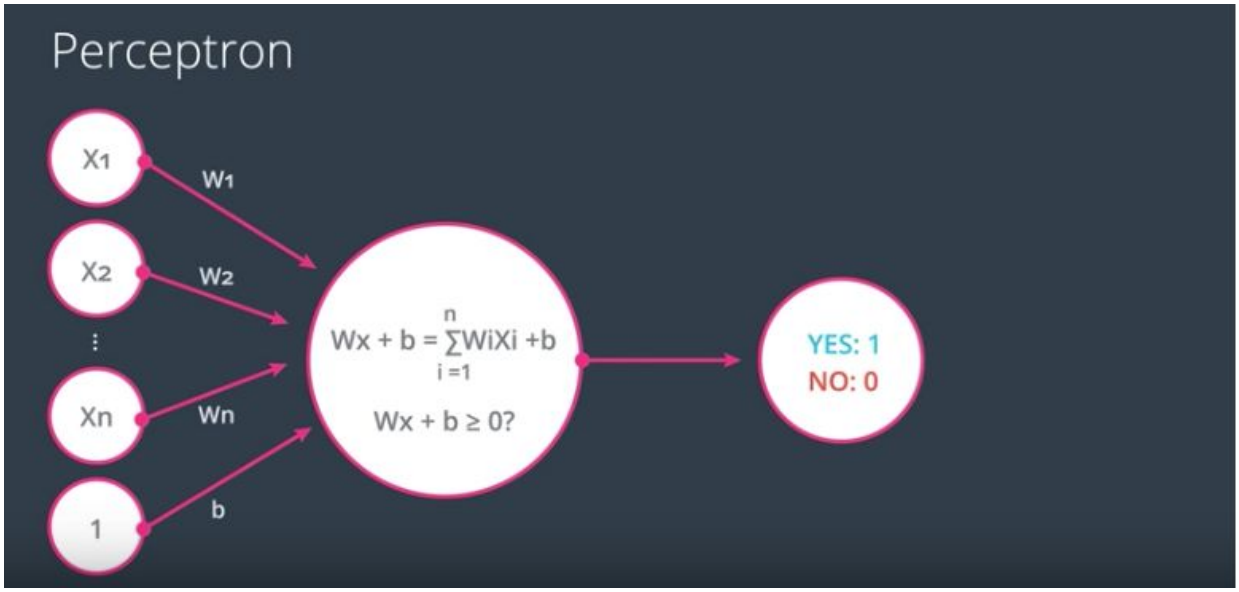
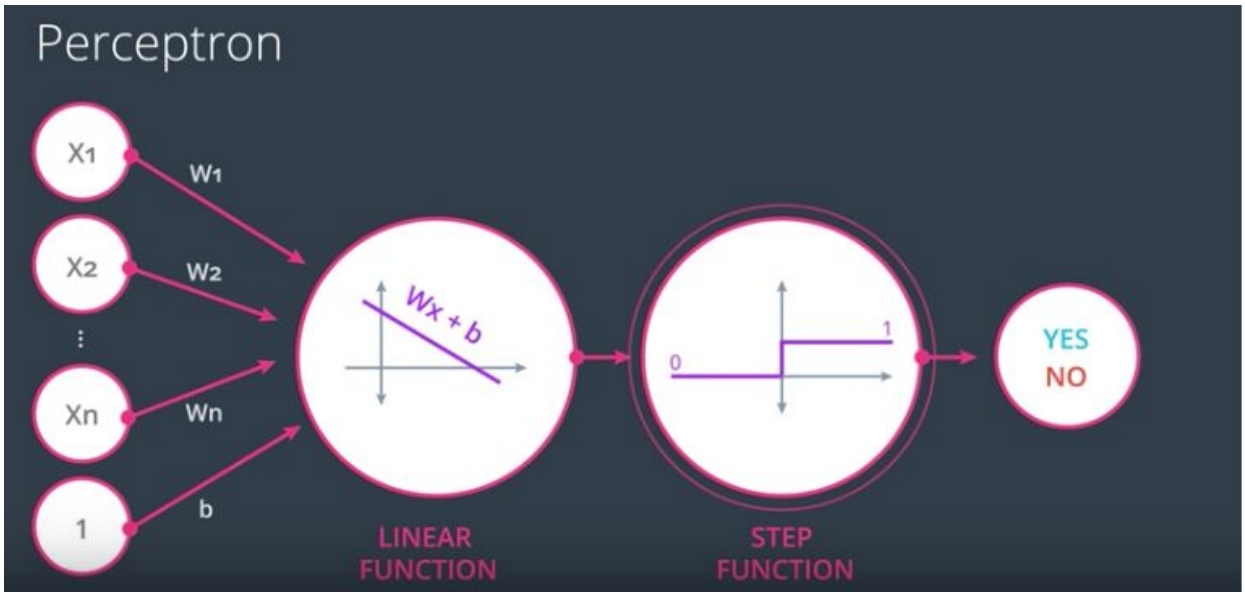
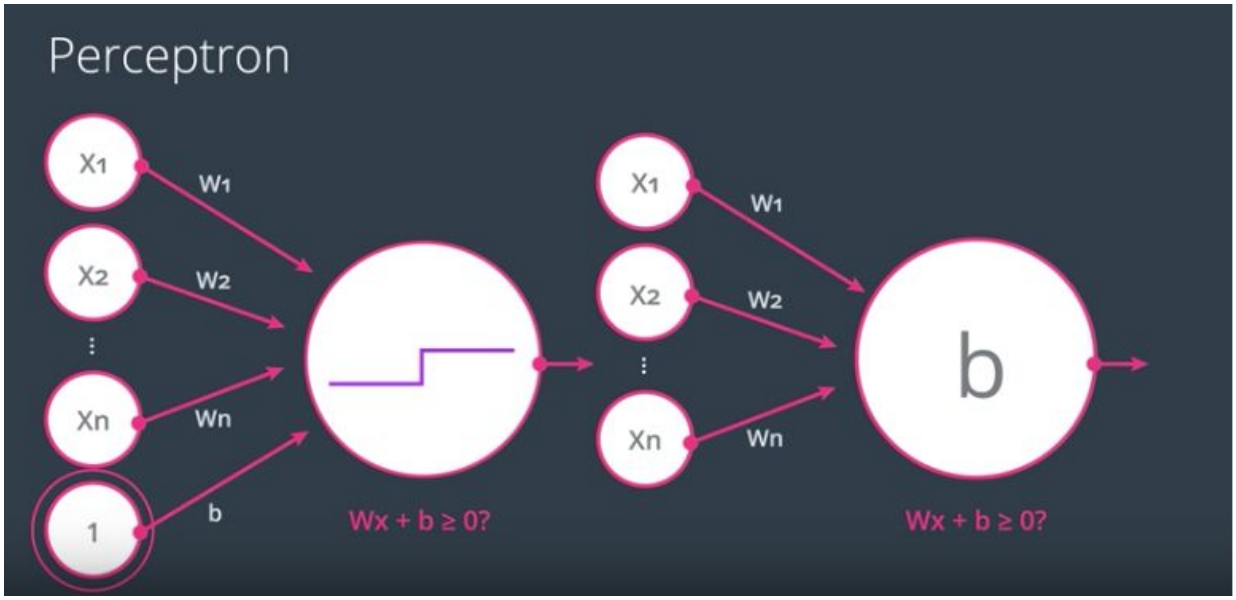
Neural Networks (NNs)

analogous to linear regression: (video)

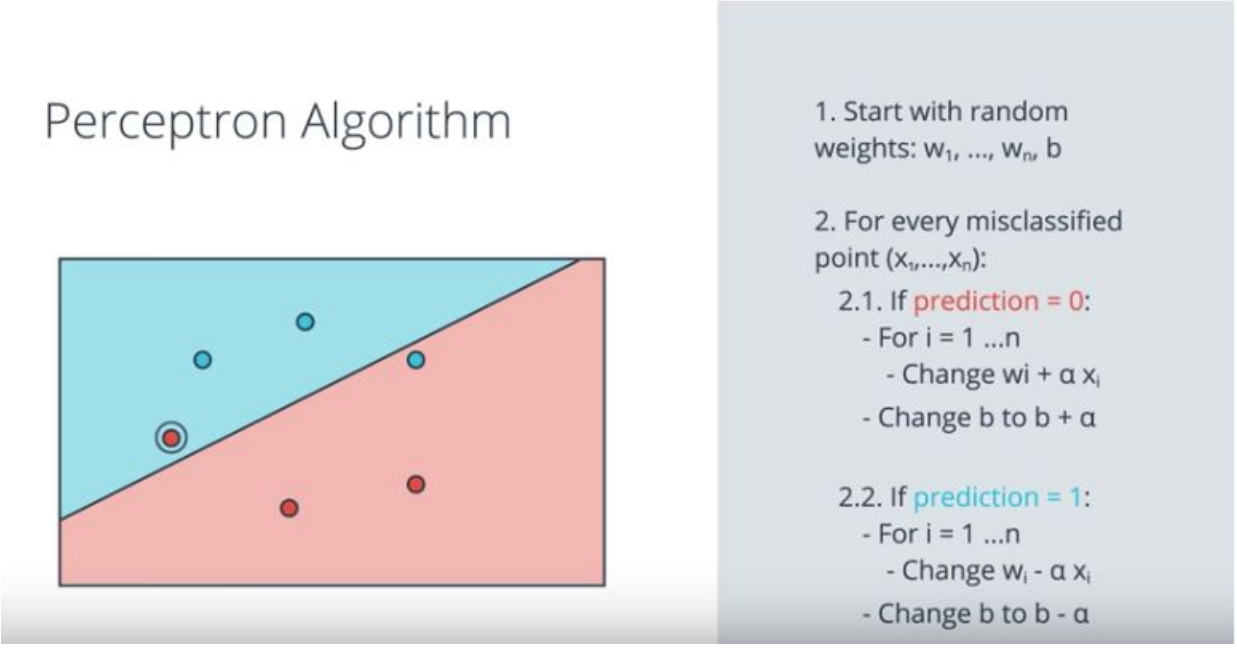
as NN:




Perceptron
X1
W1
X2
W2
Xn
Wn
1
b
Wx + b
LINEAR FUNCTION
0
1
STEP FUNCTION
YES
NO

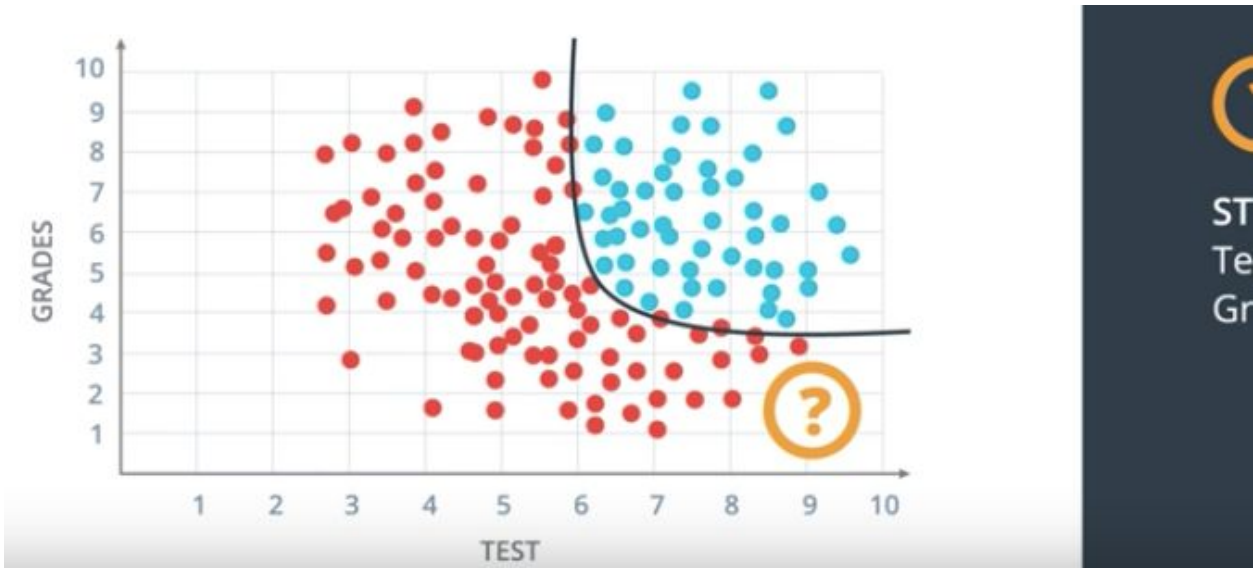

Perceptron
X1
W1
X2
W2
Xn
Wn
1
b
Wx + b ≥ 0?
X1
W1
X2
W2
Xn
Wn
b
Wx + b ≥ 0?

Perceptron algorithm



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 missclassified:

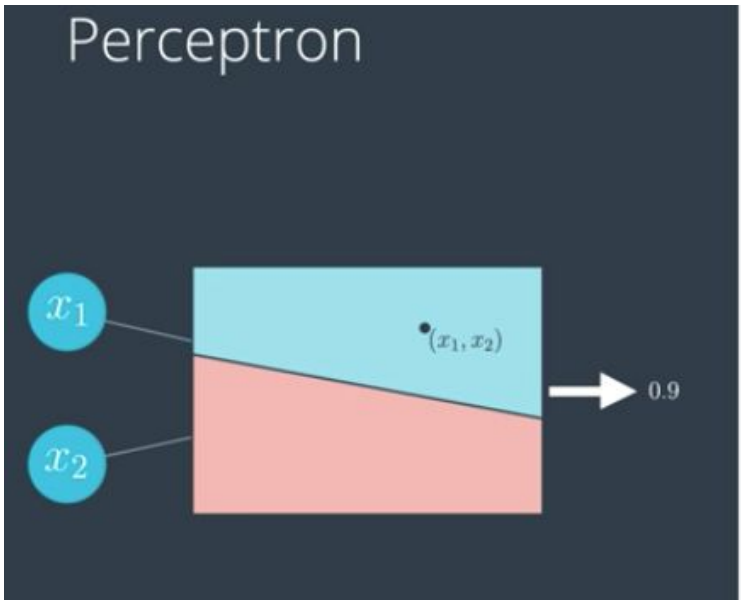
Change w_i to $\begin{cases} w_i + \alpha x_i \text{ if positive} \\ w_i - \alpha x_i \text{ if negative} \end{cases}$

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

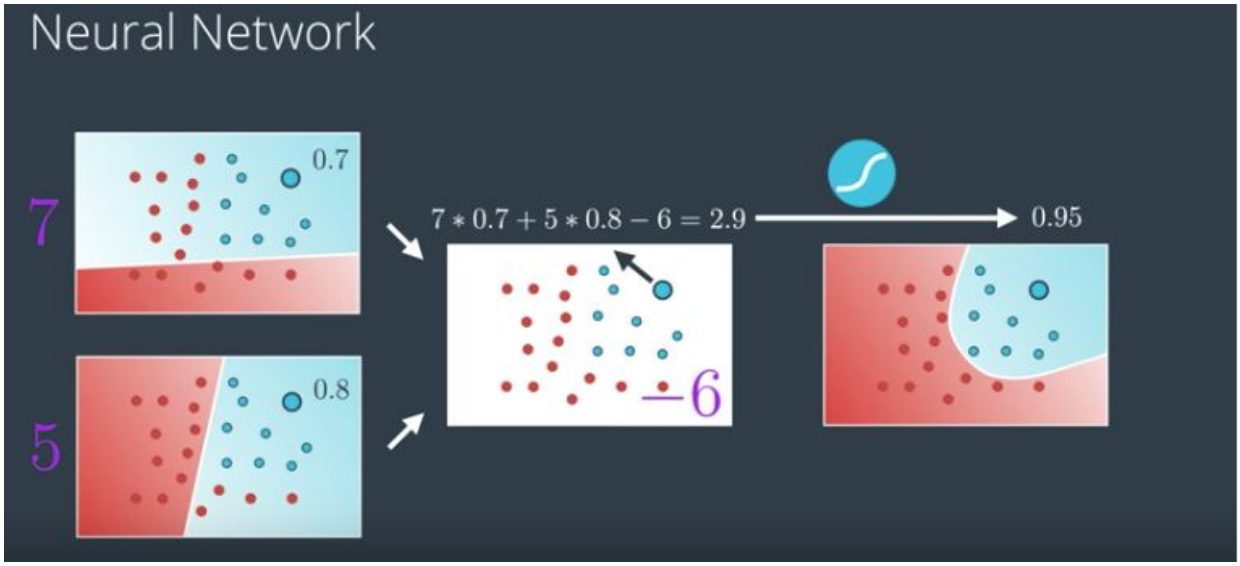
If missclassified: $\begin{cases} y-\hat{y} = 1 \text{ if positive} \\ y-\hat{y} = -1 \text{ if negative} \end{cases}$

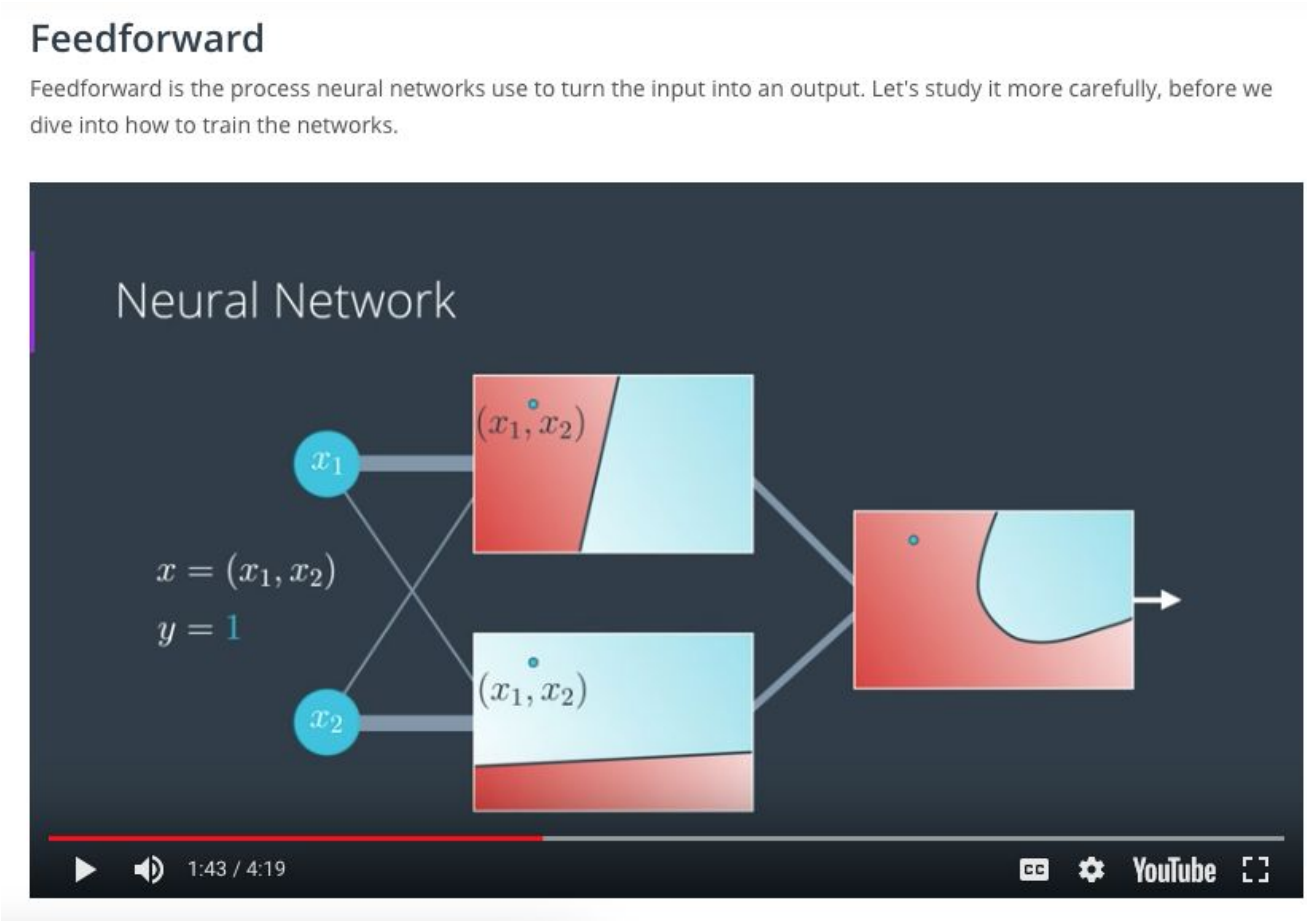
both algorithms are basically the same!

continous perceptron:



NN architecture: (video)





video

Backpropagation explanation: video