Note: This is the summary note from Udacity Introduction to Deep Learning with PyTorch

Non-Linear Region
- Data can no longer be separated by a line !!!
- Original Perceptron won't work --> so we need to update this algorithm for line so that it generalized to other type of curve
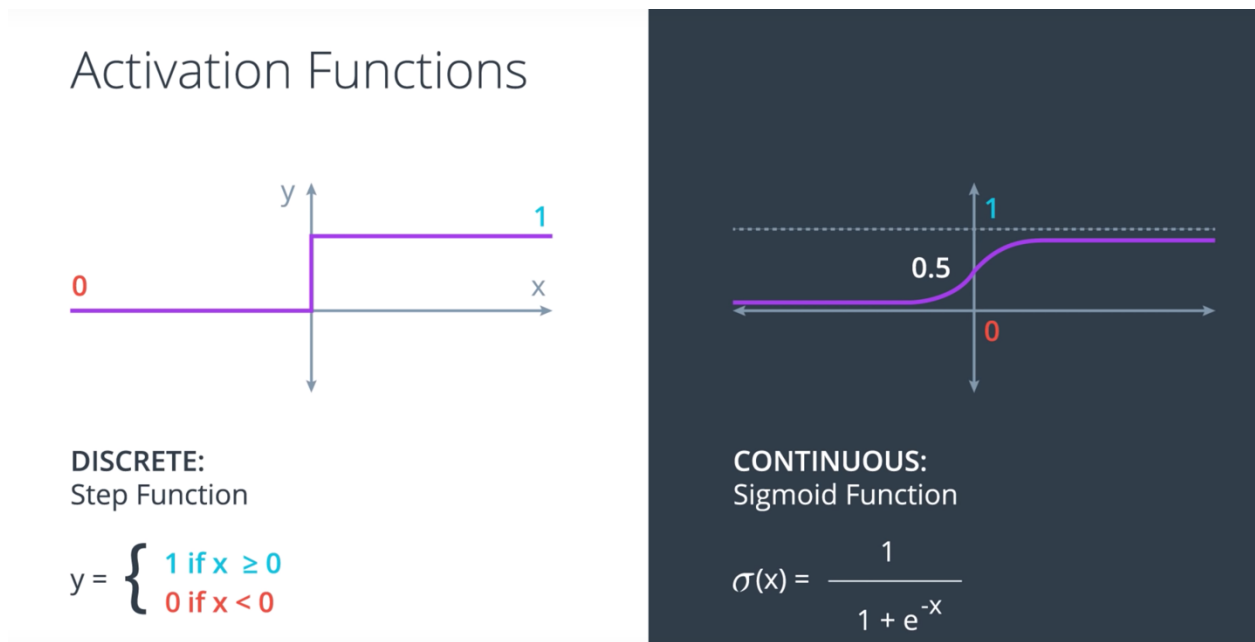
Error Function
- Help to resolve the non-linear region issue above
- Tell us how far we are from our goal / solution
- The function should return **high values** for **bad predictions** and **low values** for **good predictions**.

Log-Loss Error Function
- Logarithmic Loss, a classification loss function
- **Gradient descent can be used only if error function is continuous / differentiable**
  - Continuous = when small variation of the change translate small variation of the error function
- Log Loss quantifies the accuracy of a classifier by penalizing false classifications. Minimizing the Log Loss is basically equivalent to maximizing the accuracy of the classifier
- https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
- https://datawookie.netlify.com/blog/2015/12/making-sense-of-logarithmic-loss/
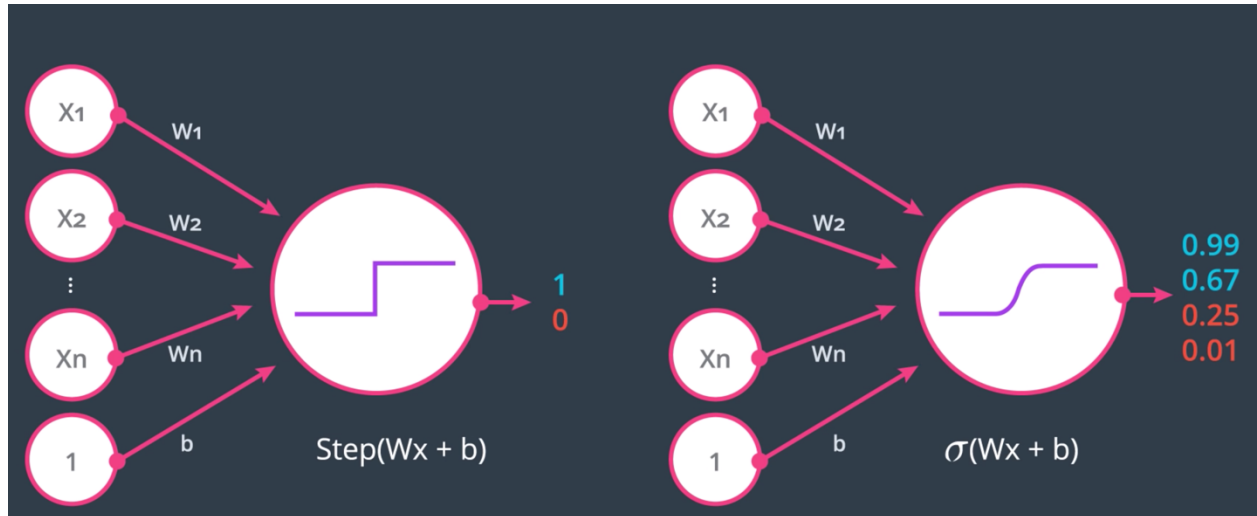
Discrete vs Continuous
- The goal is to move from discrete prediction to continuous prediction so that we can have a continuous error function that allows us to use gradient descent
- How?: we can resolve by changing activation function

Activation Functions

DISCRETE:
Step Function

$$y = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

CONTINUOUS:
Sigmoid Function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- How to obtain probability in space ?
  - Combine linear function $WX + b$ with sigmoid function

Wx + b

$\hat{y} = \sigma(Wx + b)$
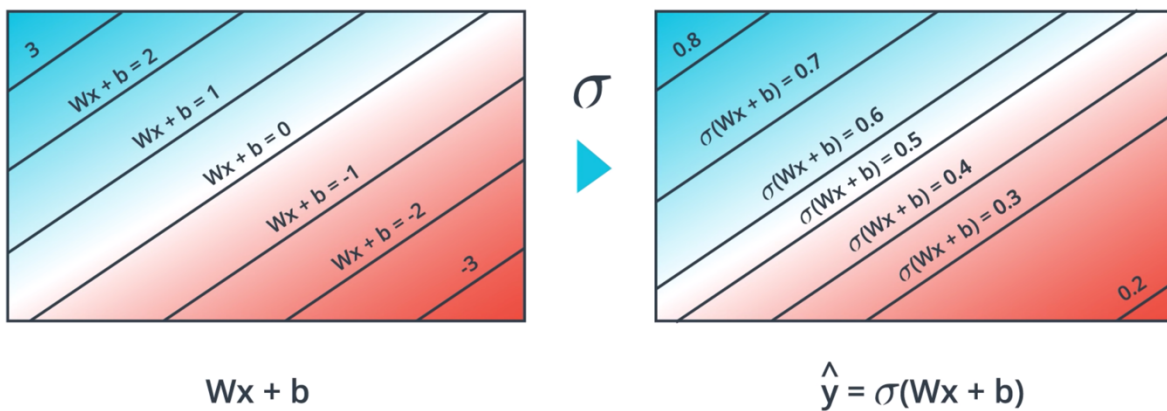
- Activation function in perceptron in this case change from step function to sigmoid function

- Tool: use this online calculator: https://www.mathway.com/Algebra

===

Mini summary:
- Problem: Data can no longer be separated by a line (Non-
- Linear Region)
- Solution: Error function
- How: move from discrete prediction to continuous prediction so that we can have a continuous error function that allows us to use gradient descent
- Sigmoid function

===