

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

Perceptron vs Gradient Descent

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

- In Gradient Descent, everyone points change the weight; While perceptron algorithm only change the weight for misclassified
- Perceptron algorithm: label is 1 or 0; prediction \hat{y} also 1 or 0
- Perceptron algorithm: \hat{y} value is either 1 or 0 while Gradient Descent is between 0 and 1

Continuous Perceptrons

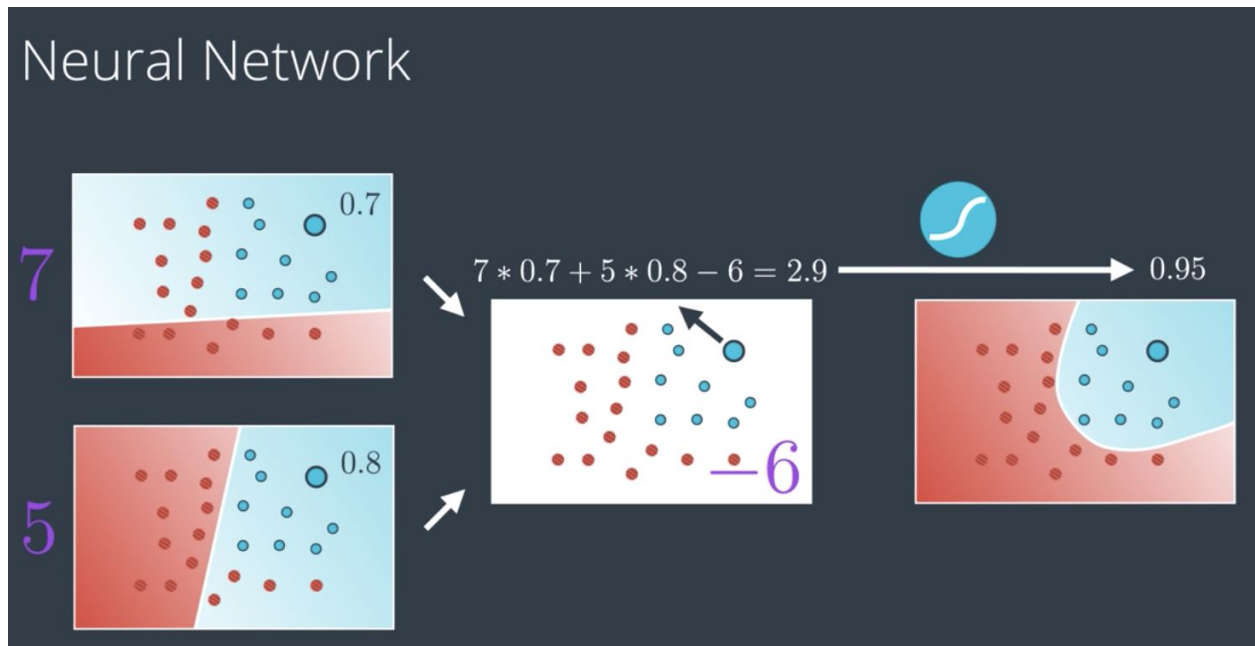
- Just recap of what perceptrons does

Non-Linear Data and Non-Linear Model

- Problem: how to deal with more complicated data set that require non-linear boundary
- Define the non-linear boundary with the help of neural network, that define non-linear equation

Neural Network Architecture

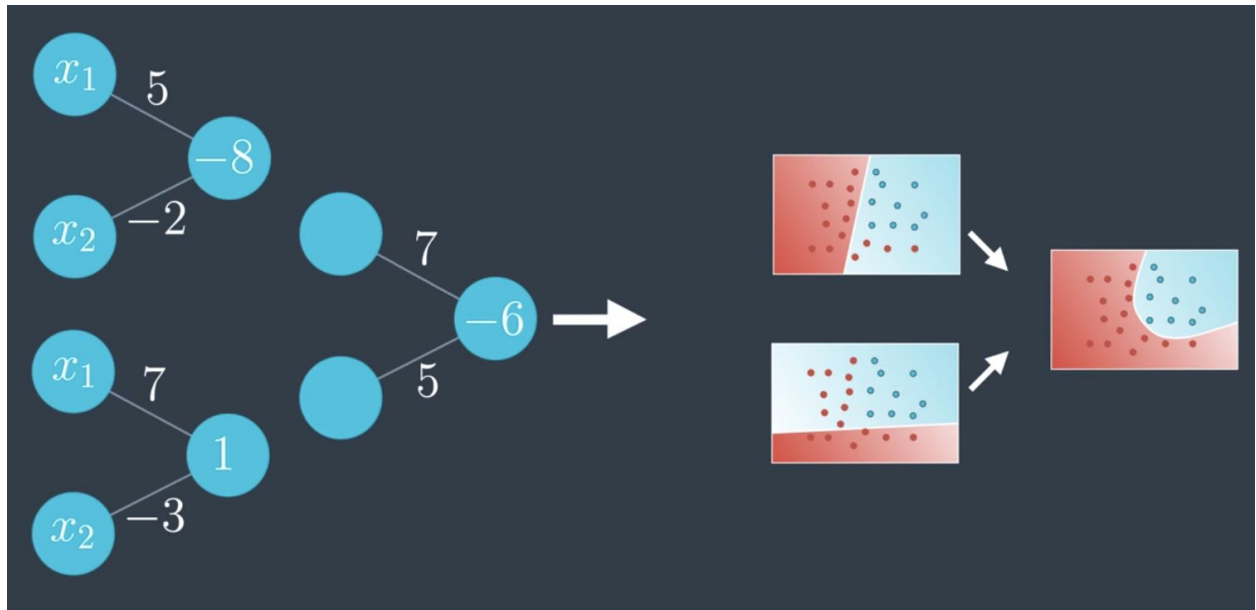
- Building block of Neural Network = Combine two linear models into non-linear models
 - Create probability function of each point
 - Sigmoid function used in the image above to convert the number to range between 0 and 1
- **Example:**
 - **Represented by graphic**



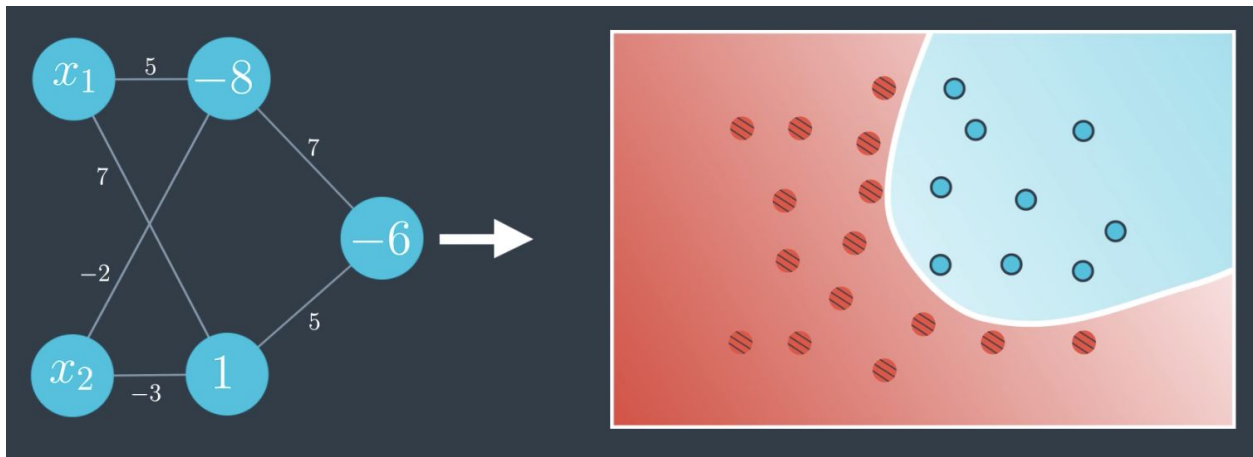
Note:

- 7, 5 are the weight we added (can be any number)
- -6 is the bias we added (can be any number)
- Curved model (Non linear model) in the right is a linear combination of linear-models on the left

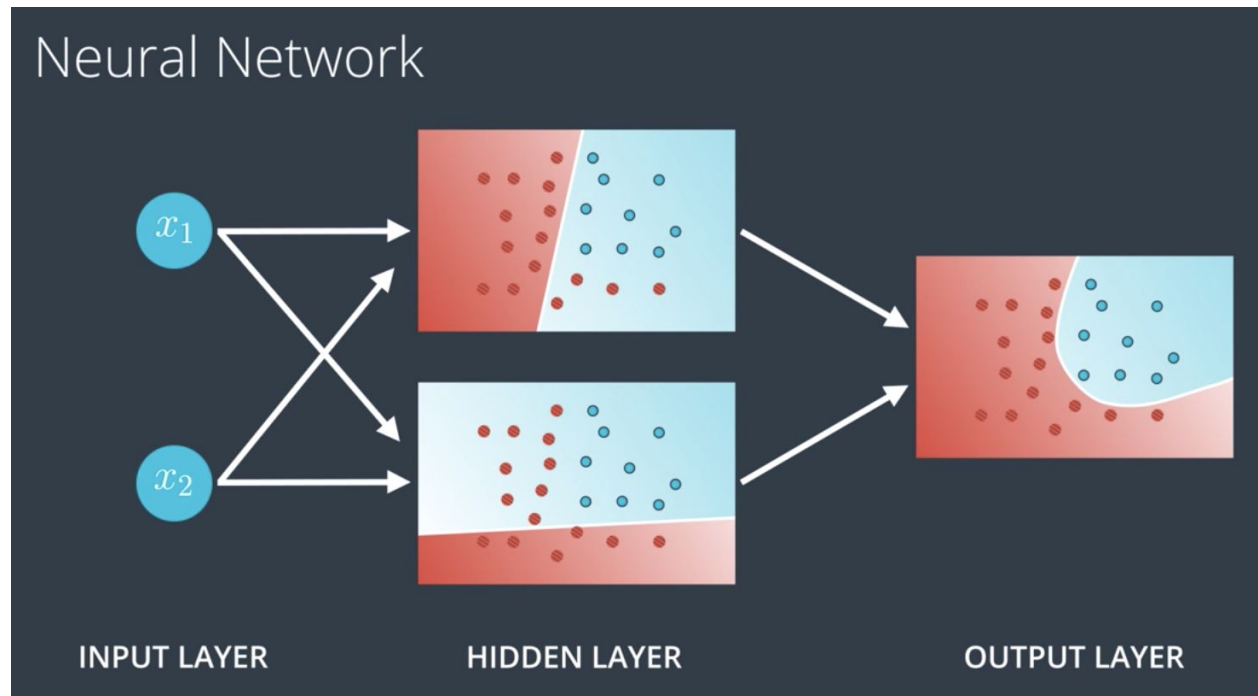
- Now represented by perceptron:



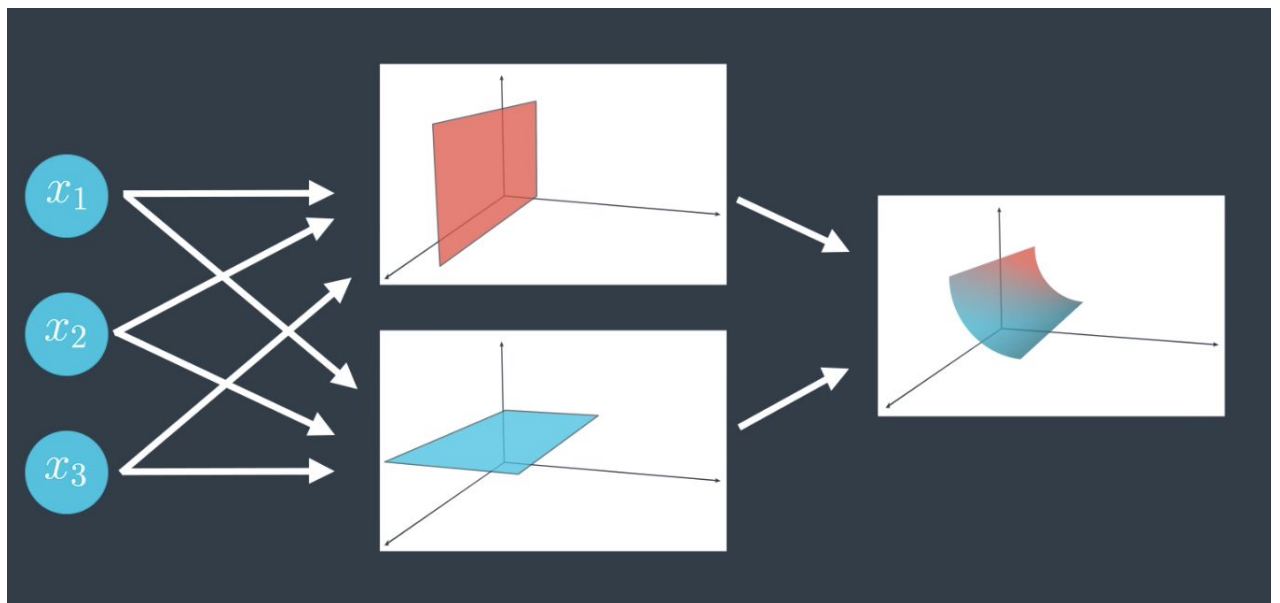
- 1st perceptron: $5x_1 - 2x_2 - 2 = 0$
- 2nd perceptron: $7x_1 - 3x_2 + 1 = 0$
- 3rd perceptron: $7(1st\ perceptron) + 5(2nd\ perceptron) - 6 = 0$



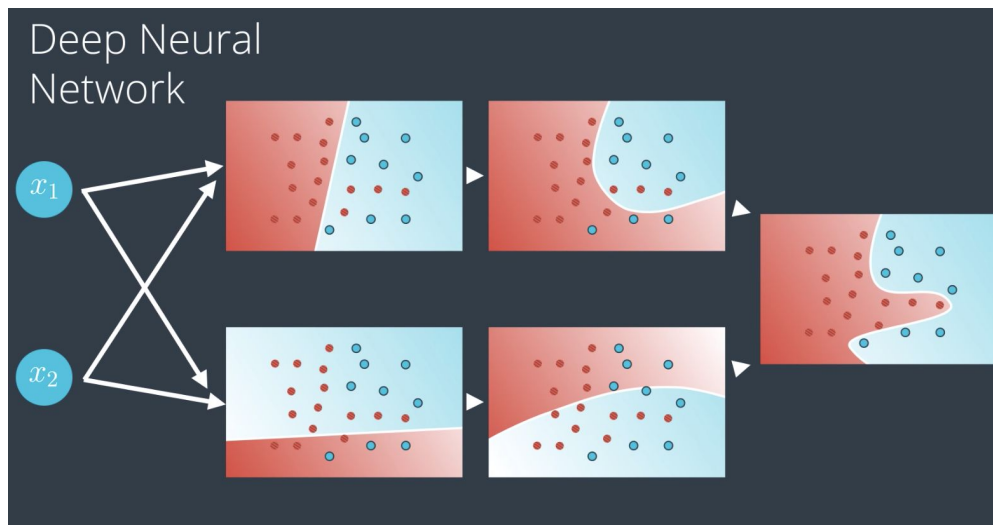
- **Neural Network Architecture**



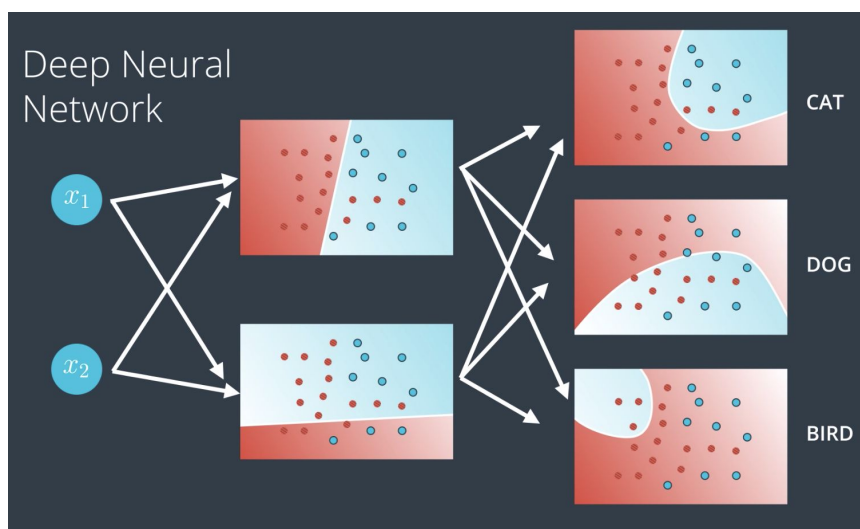
- The input layer, hidden layer and output layer can be varies based on the requirement and complexities of the model
- **Example :**
 - **More input layer**



- **More Hidden layer (Deep Neural Network)**



- **More output layer**



=====

Mini Summary

- Differences between Gradient Descent and Perceptron
 - In Gradient Descent, everyone points change the weight; While perceptron algorithm only change the weight for misclassified
- Building block of Neural Network = Combine two linear models into non-linear models; this help resolve the problem on non-linear data and non-linear model
- Neural Network Architecture (for now) consist of Input layer, hidden layer, output layer, and they are all varies based on requirements and complexities