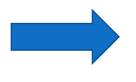
Human Brain VS Computer

Motivation







- Human mind Computer
- Good at image recognition, pattern recognition etc
- Good at arithmetic calculations



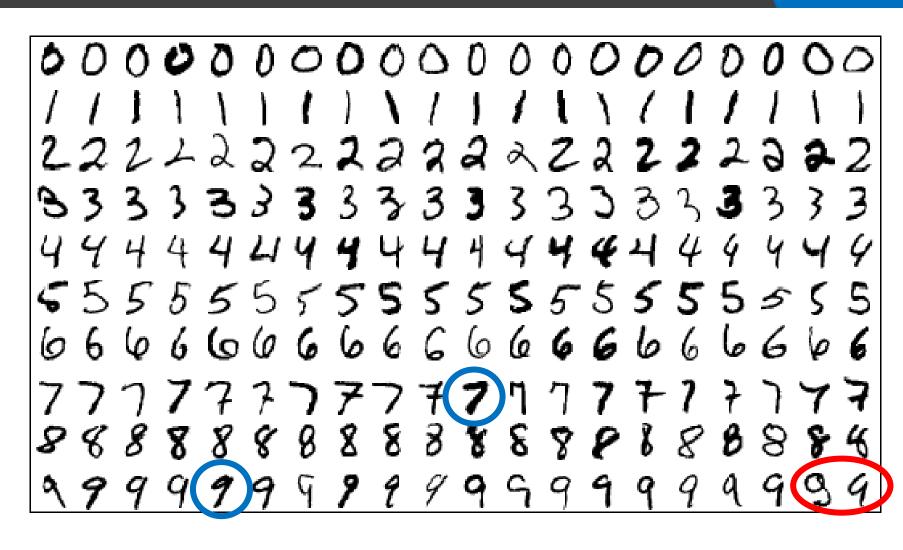


 $2574304 \times e^{354} \div \tan 5.1\pi$



Handwriting recognition

Making precise rules is difficult

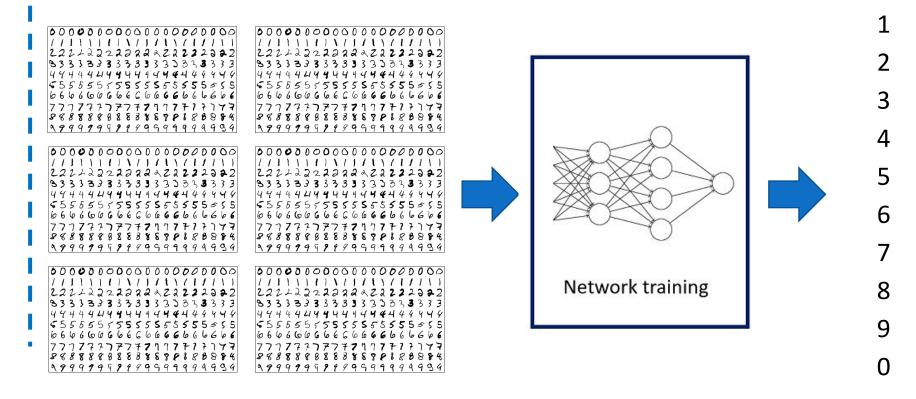




Neural Networks

Neural Networks creates own complex pattern recognition rules

Pattern recognition





Training data

Future Prediction

Dataset

Fashion MNIST

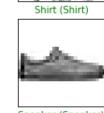
We will classify images into 10 fashion items



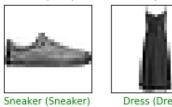


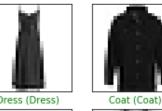
Pullover (Pullover)

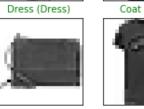
Sandal (Sandal)



Trouser (Trouser)



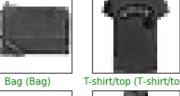


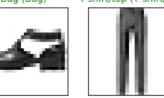


Sandal (Sandal)



Sneaker (Sneaker)





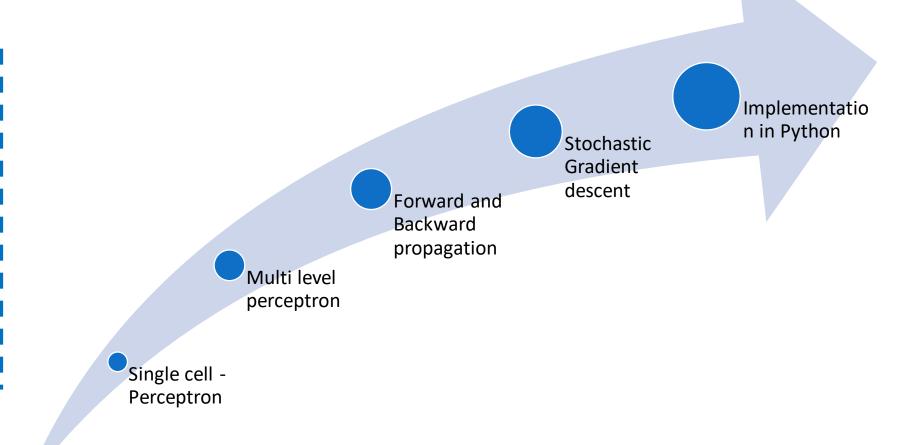






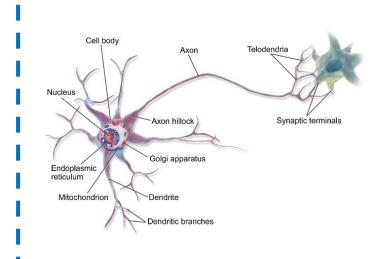
Course Flow



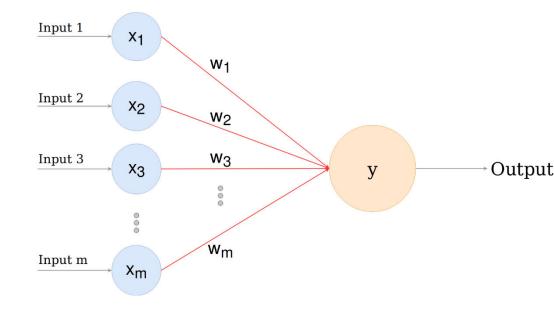




Artificial Neuron



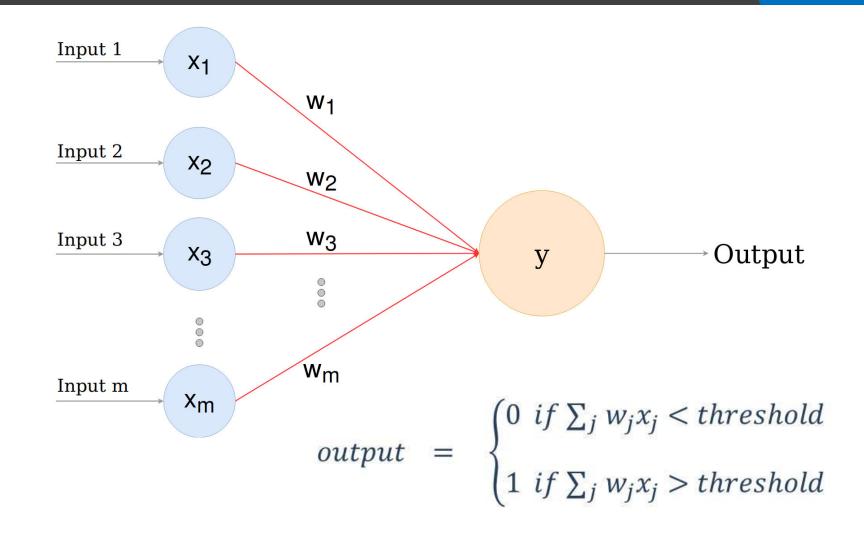
Biological Neuron



Artificial Neuron



Artificial Neuron





Purchasing a Shirt

Color

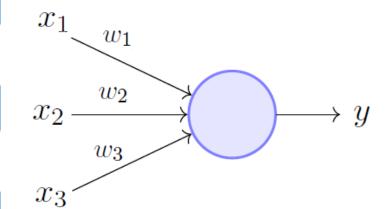
• Blue or Not

Sleeves

• Full or half

Fabric

• Cotton or not





Purchasing a Shirt

Color

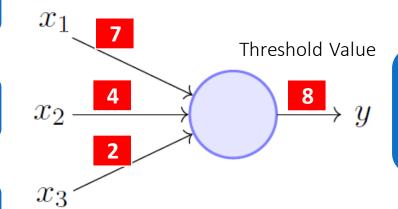
• Blue or Not

Sleeves

• Full or half

Fabric

• Cotton or not





Purchasing a Shirt

Color

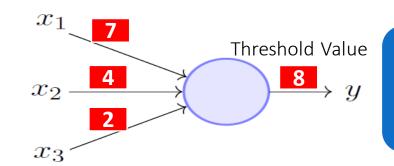
• Blue or Not

Sleeves

• Full or half

Fabric

• Cotton or not



Color	Sleeves	Fabric	Calculated Sum	Threshold	Buy / Not Buy
Blue	Half	Non Cotton	7*1 + 4*0 + 2*0 = 7	8	Not buy
Blue	Full	Non Cotton	11	8	Buy
Not Blue	Full	Cotton	6	8	Not Buy



Purchasing a Shirt

Color

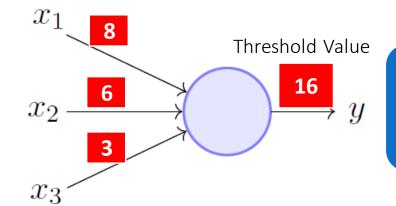
• Blue or Not

Sleeves

• Full or half

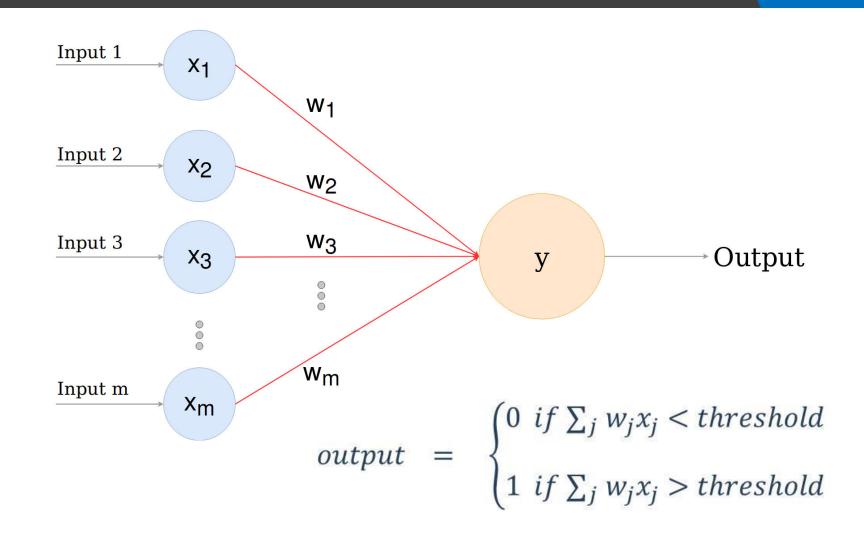
Fabric

• Cotton or not



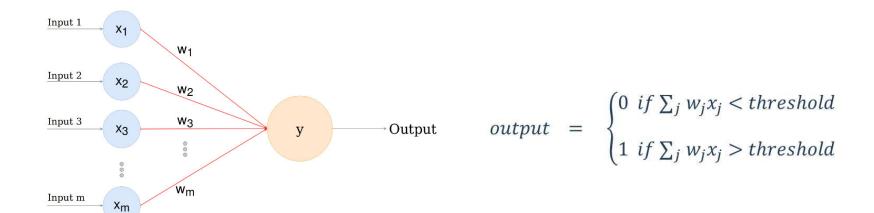


Removing Binary Restriction





Standard Equation

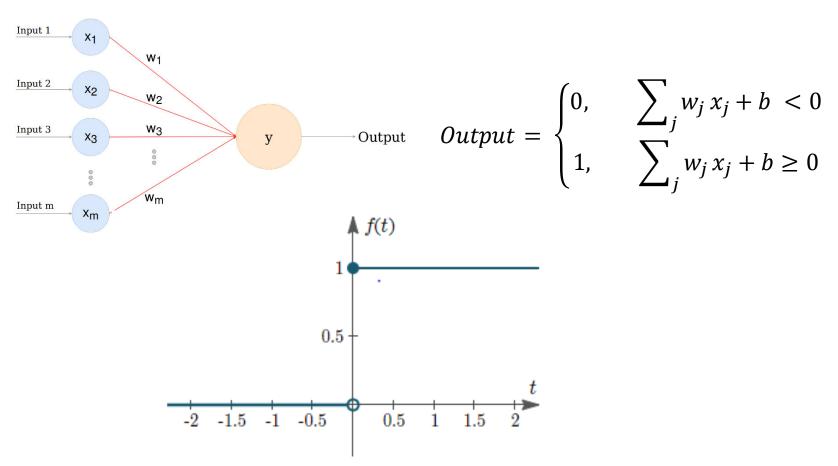


$$Output = \begin{cases} 0, & \sum_{j} w_{j} x_{j} + b < 0 \\ 1, & \sum_{j} w_{j} x_{j} + b \ge 0 \end{cases}$$

b is called Bias



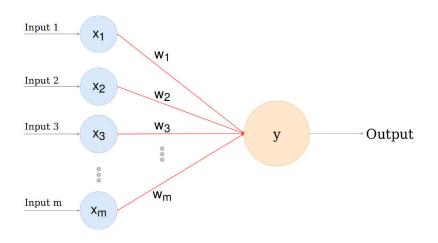
Graphical Representation

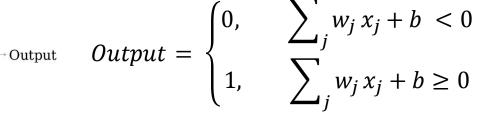


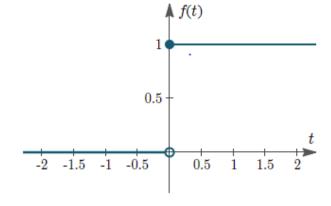
Step Activation function



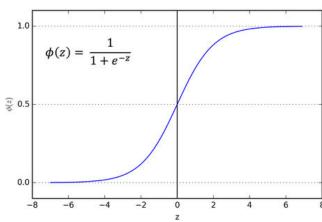
Sigmoid Activation







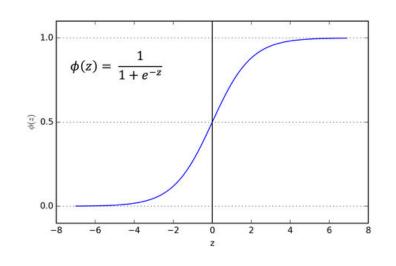




Sigmoid Activation function



Sigmoid Activation



Sigmoid Activation function

- Sigmoid is better because it is less sensitive to individual observation
- Artificial neuron with sigmoid activation is called sigmoid or logistic neuron

$$\sigma(z) \equiv rac{1}{1+e^{-z}}, \hspace{1cm} \mathit{Output} = \hspace{1cm} rac{1}{1+\exp(-\sum_{j}w_{j}x_{j}-b)}.$$



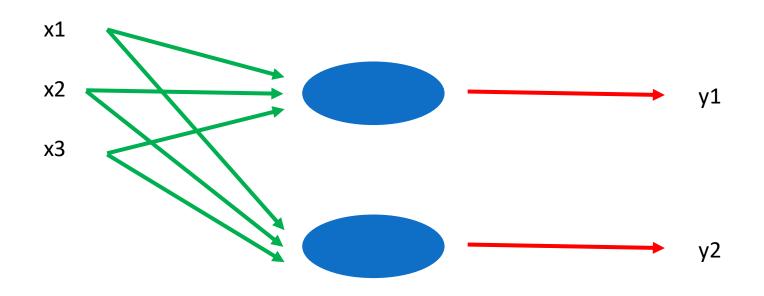
Two types of Stacking

Parallel

Sequential



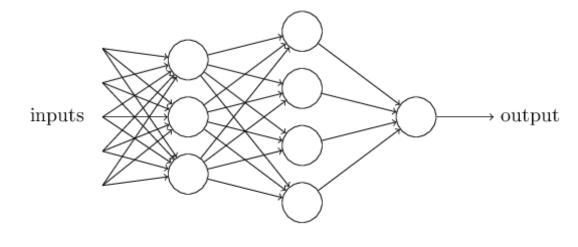
Parallel Stacking



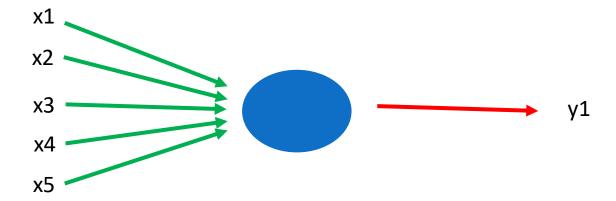
With parallel stacking we can get multiple outputs with the same input



Sequential Stacking

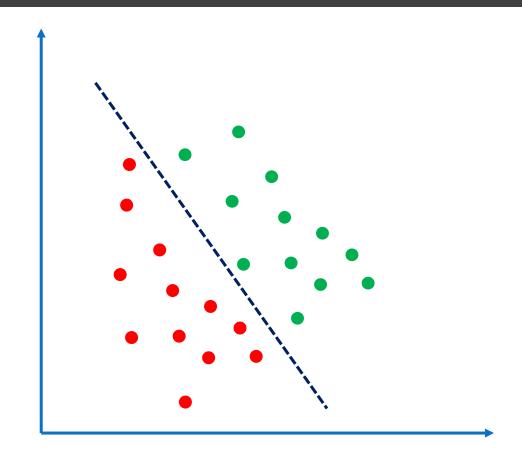


Why not use a single neuron





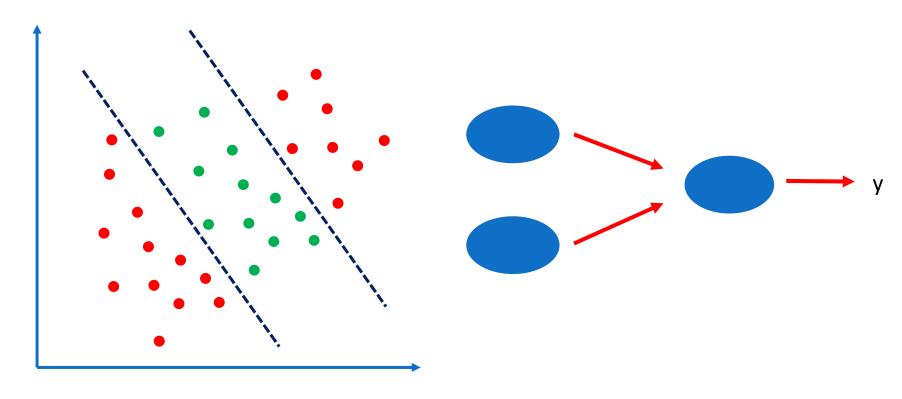
Sequential Stacking



Single neuron can handle such linear classification problem

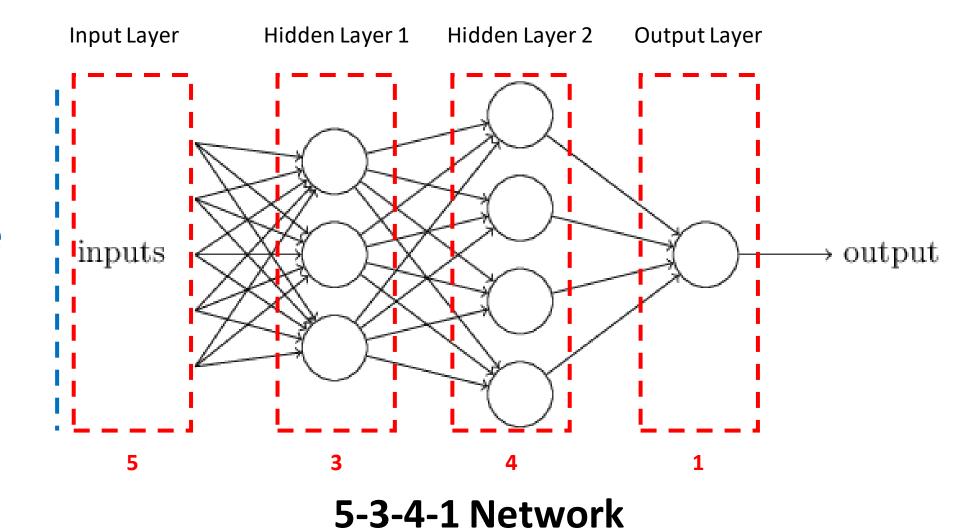


Sequential Stacking



Each neuron can focus on the particular features of the object instead of the final outcome

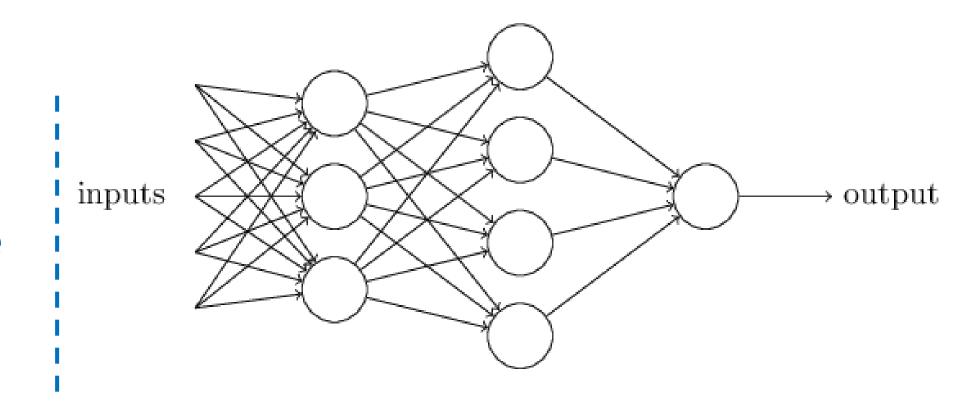




Nomenclature



Nomenclature



Feed Forward Network

One directional processing

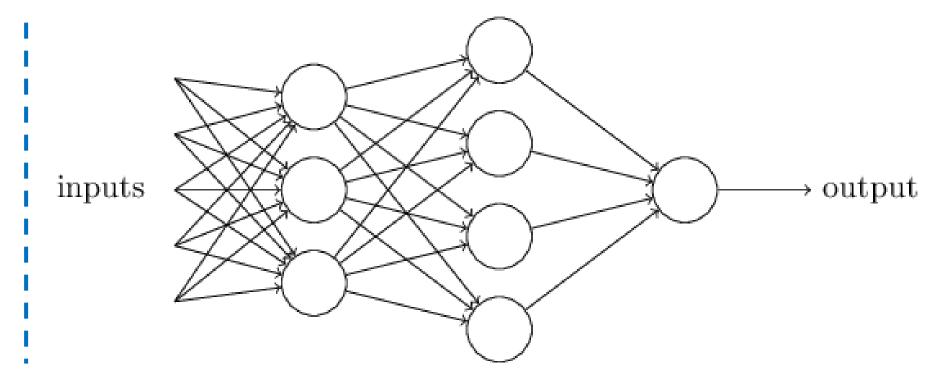
Fully connected network — Output from a neuron goes to all neurons of next layer



Deep Learning

Such artificial neural networks primarily constitutes deep learning

Deep Learning





More number of layers => Deeper network => More complex relationships

Neural Network

How it works

Covered till Now

What is a neural network

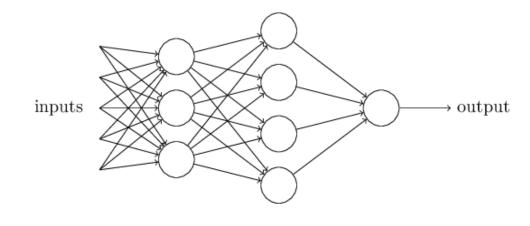
Now we are going to learn

How does a neural network works



Problem Statement

Quick Recap



$$\sigma(z) \equiv rac{1}{1 + e^{-z}}$$

$$Output = \frac{1}{1 + \exp(-\sum_{j} w_{j} x_{j} - b)}.$$

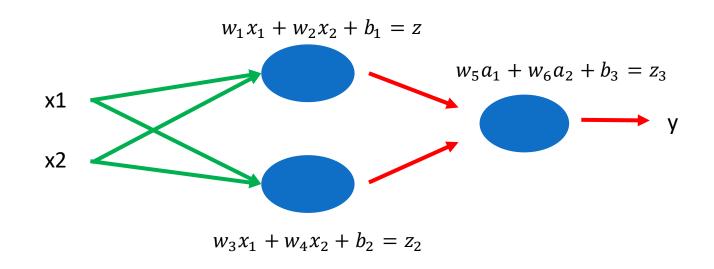
Problem Statement

• Establish the values of weights and biases so that predicted output is as close to actual output as possible



Problem Statement

Example



Variables to be established in this neural network

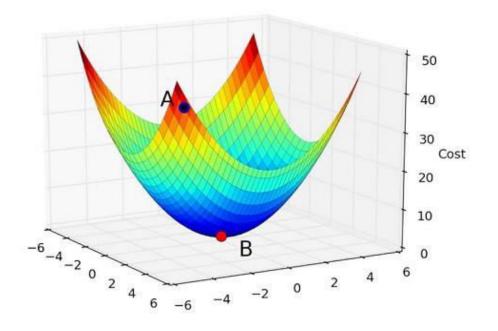
- Weights W1, W2.....W6
- Biases B1, B2, B3

Total - 9 variables



Neural Network

Gradient Descent



- GD is an optimization technique to find minimum of a function
- Better than other technique such as OLS when we have large number of features and complex relationships



Gradient Descent

 Assign random W and B values Step 1 Calculate final output using these values Step 2 • Estimate error using error function Step 3 • Find those W and B which can reduce this error Step 4 Update W and B and repeat from step 2 Step 5

Initialization

Forward

Propagation

Backward

Propagation

Implementati

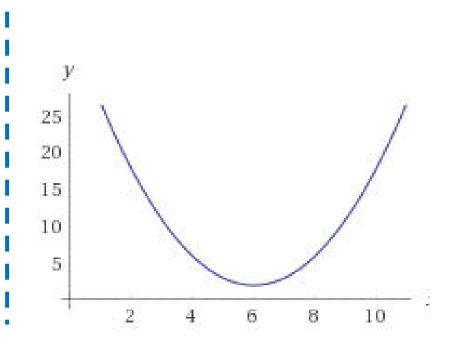
on of GD

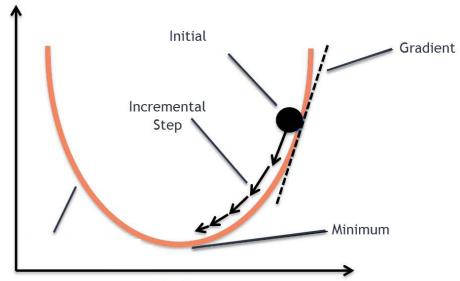


Process

Neural Network

Gradient Descent

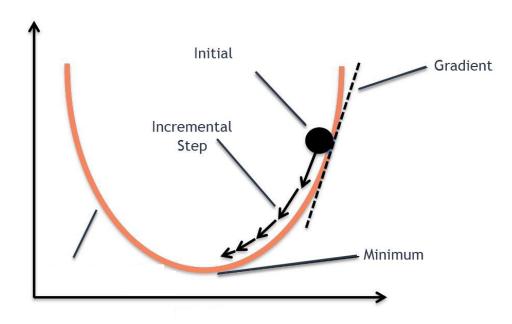






Neural Network

Gradient Descent



- 1. Start at a random point
- 2. Find out the **instantaneous slope** at that point
- 3. Slightly move in the direction of steepest slope
- 4. Reiterate





radiant Daggert

