Making Networks

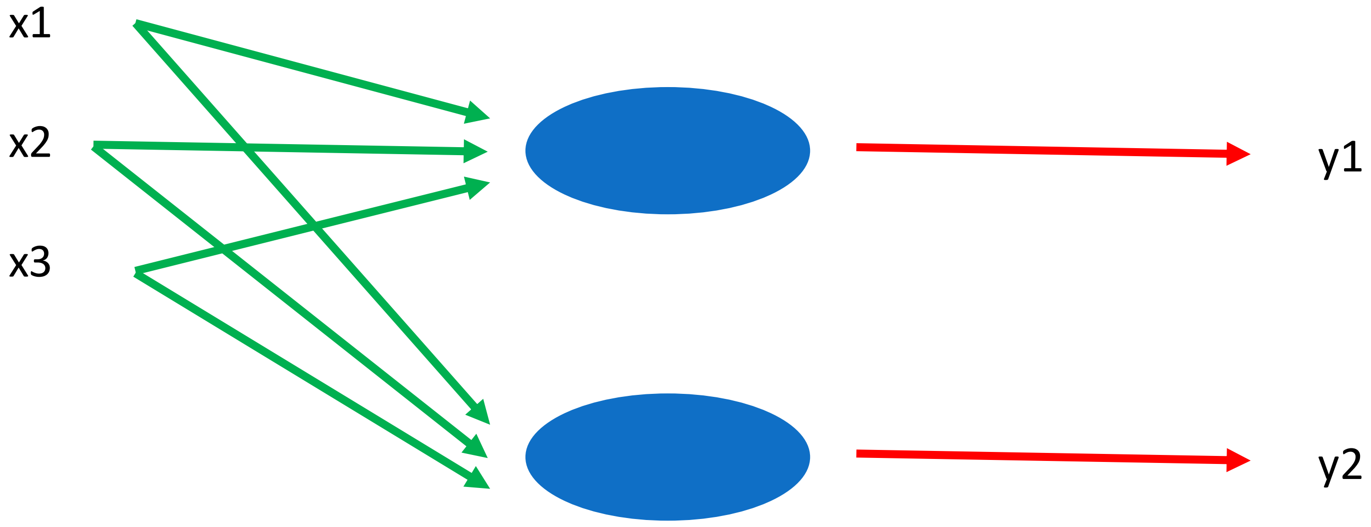
Two types of Stacking

Parallel

Sequential

Making Networks

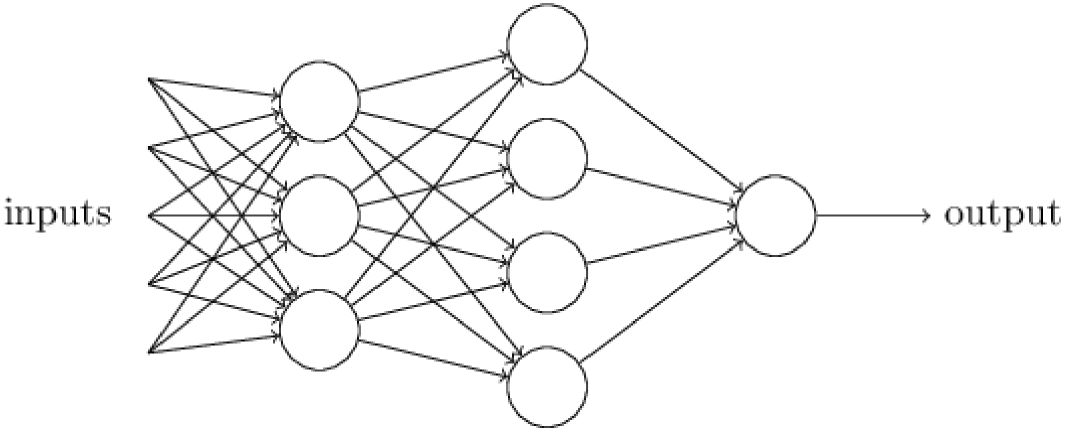
Parallel Stacking



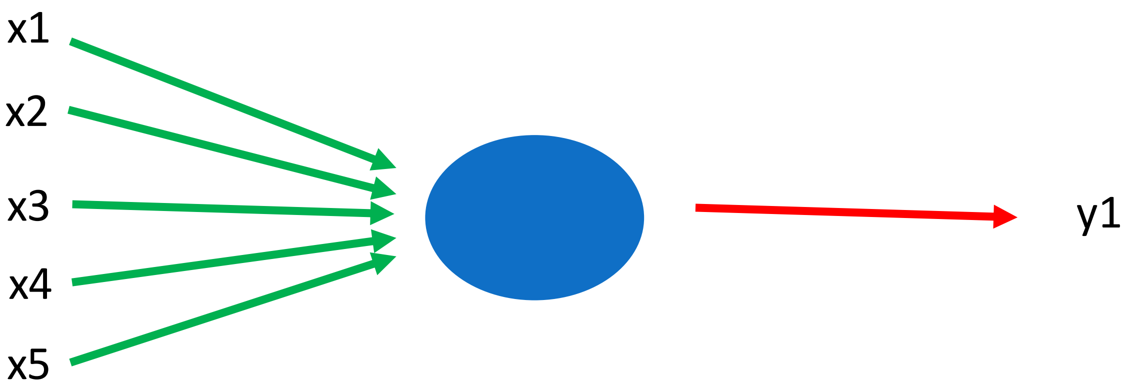
With parallel stacking we can get multiple outputs with the same input

Making Networks

Sequential Stacking

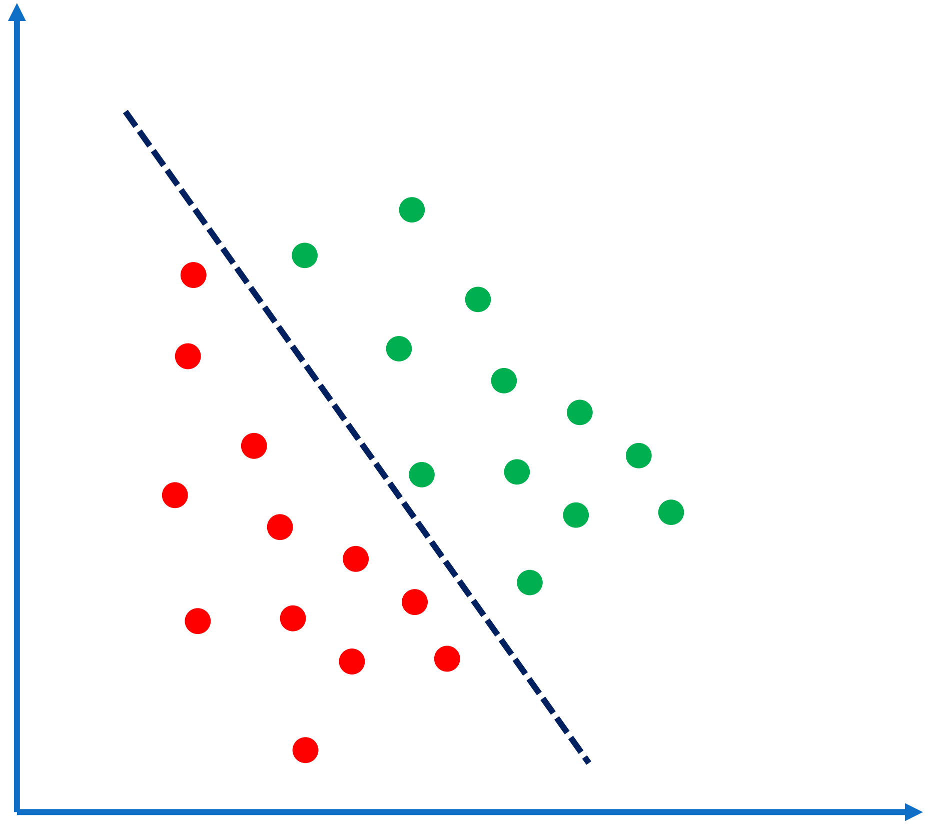


Why not use a single neuron



Making Networks

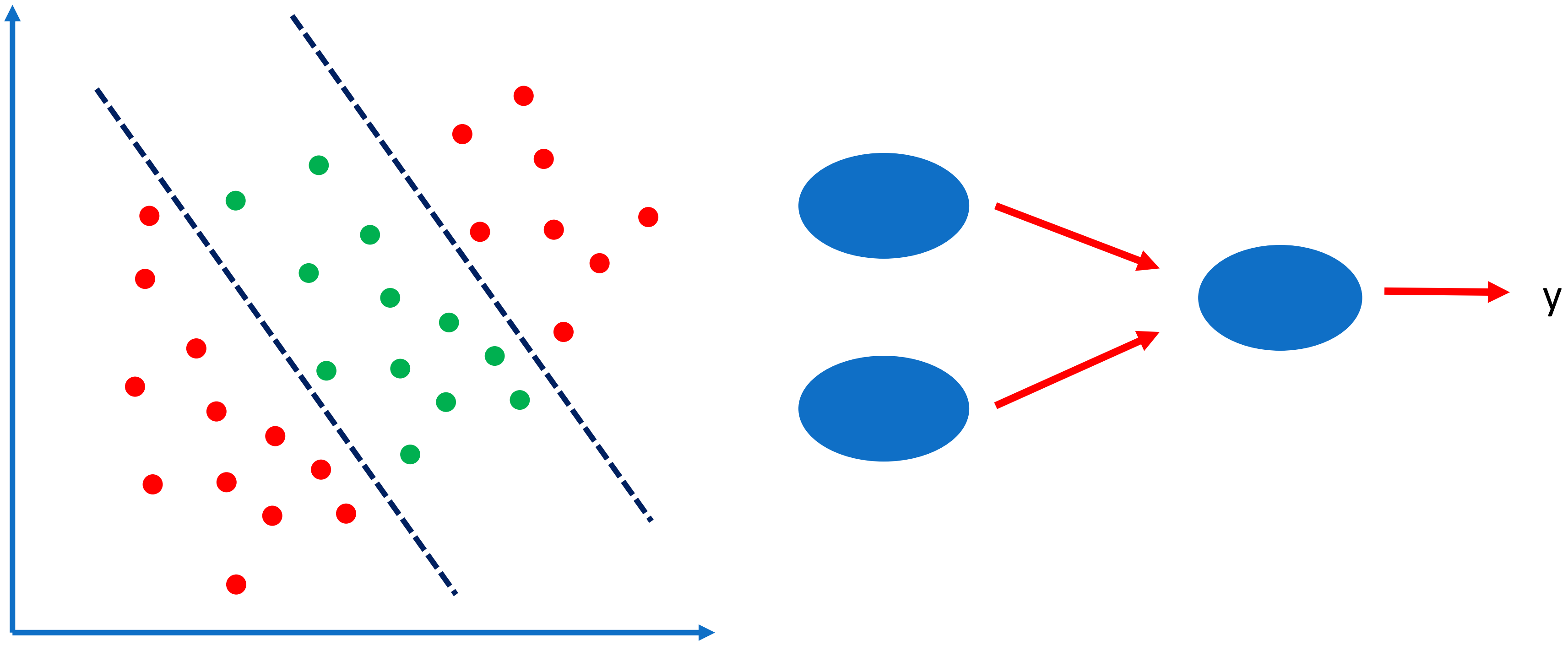
Sequential Stacking



Single neuron can handle such linear classification problem

Making Networks

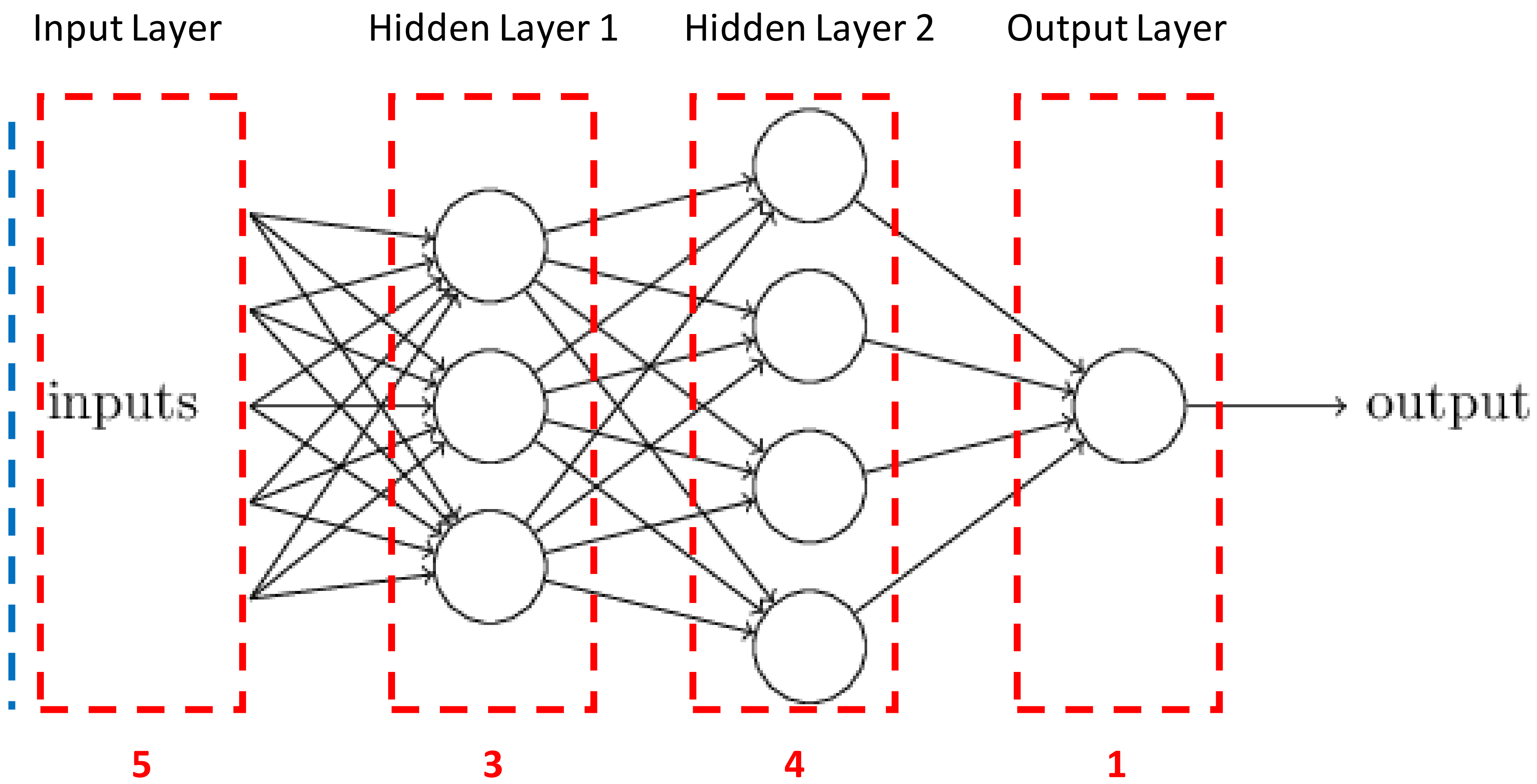
Sequential Stacking



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

Making Networks

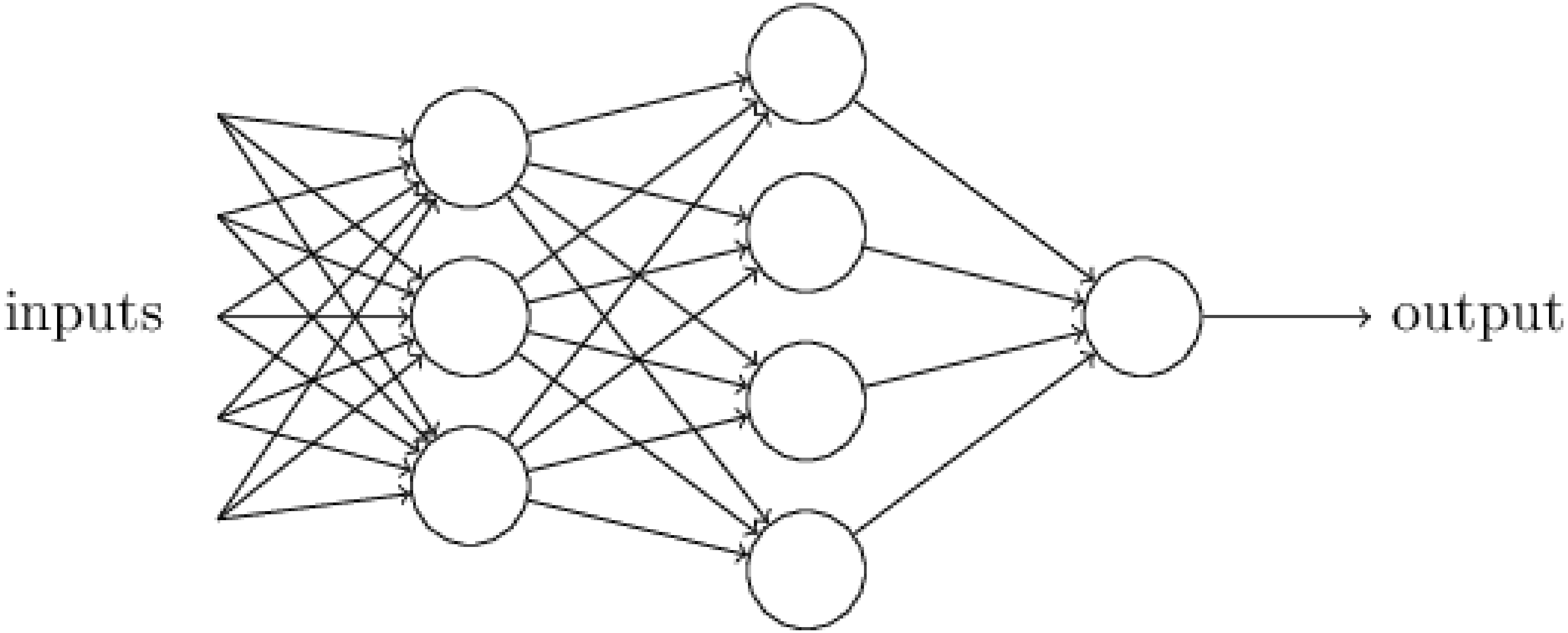
Nomenclature



5-3-4-1 Network

Making Networks

Nomenclature



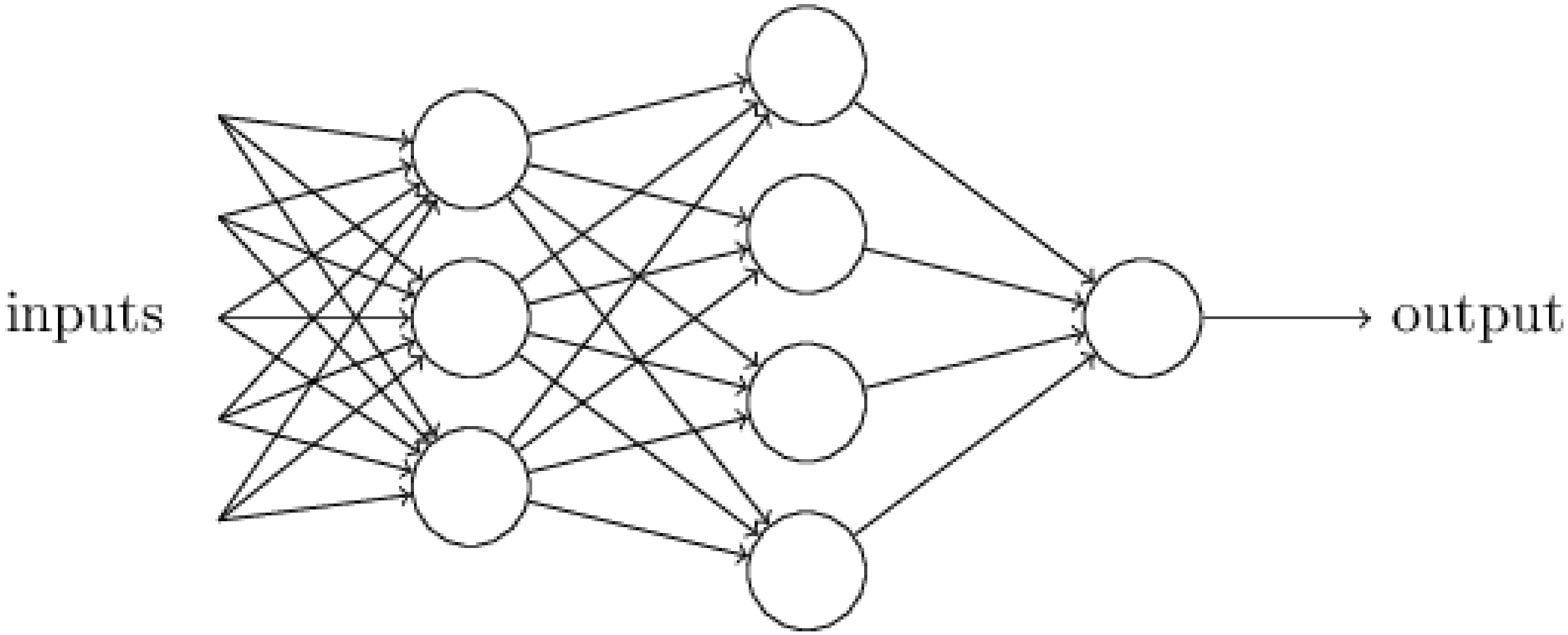
Feed Forward Network – One directional processing

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

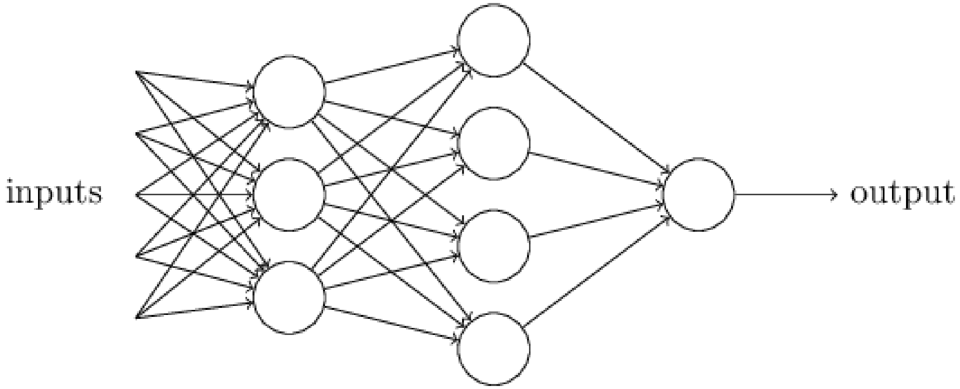
- What is a neural network

Now we are going to learn

- How does a neural network works

Problem Statement

Quick Recap



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

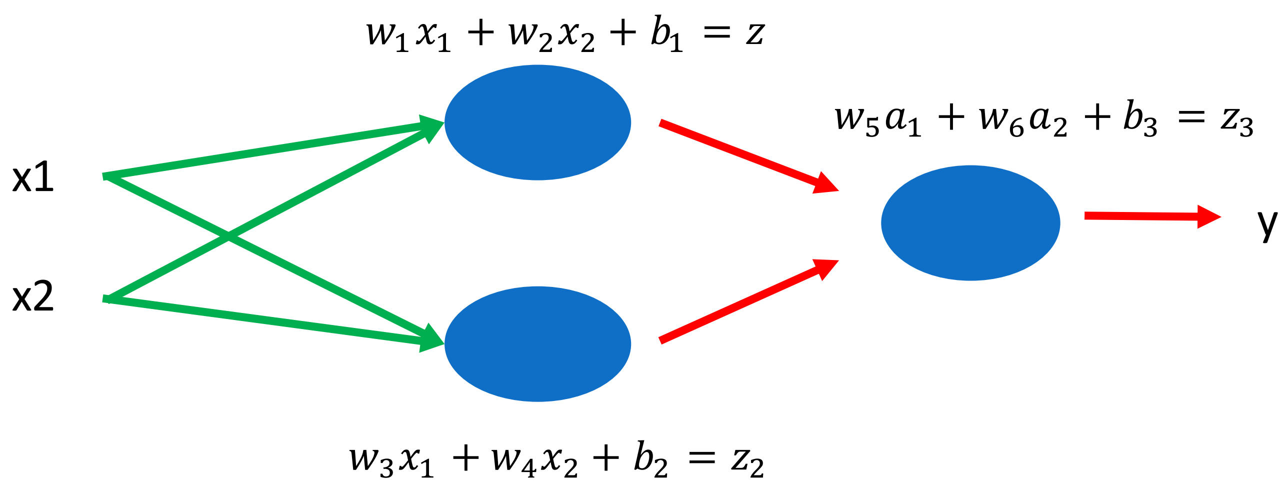
$$Output = \frac{1}{1+\exp(-\sum_j w_j x_j - b)}.$$

Problem Statement

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

Problem Statement

Example



$w_1 x_1 + w_2 x_2 + b_1 = z$

$w_5 a_1 + w_6 a_2 + b_3 = z_3$

$w_3 x_1 + w_4 x_2 + b_2 = z_2$

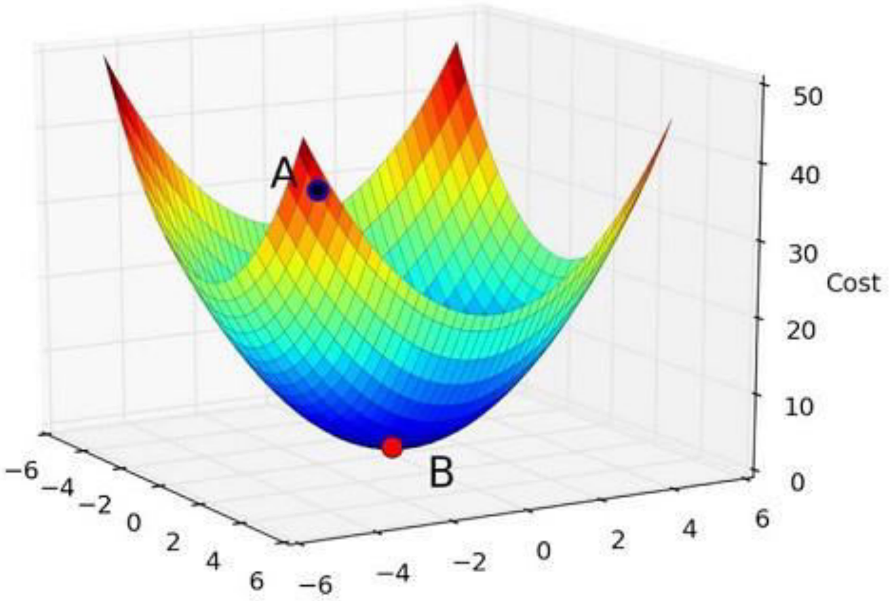
Variables to be established in this neural network

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

Total - 9 variables

Neural Network

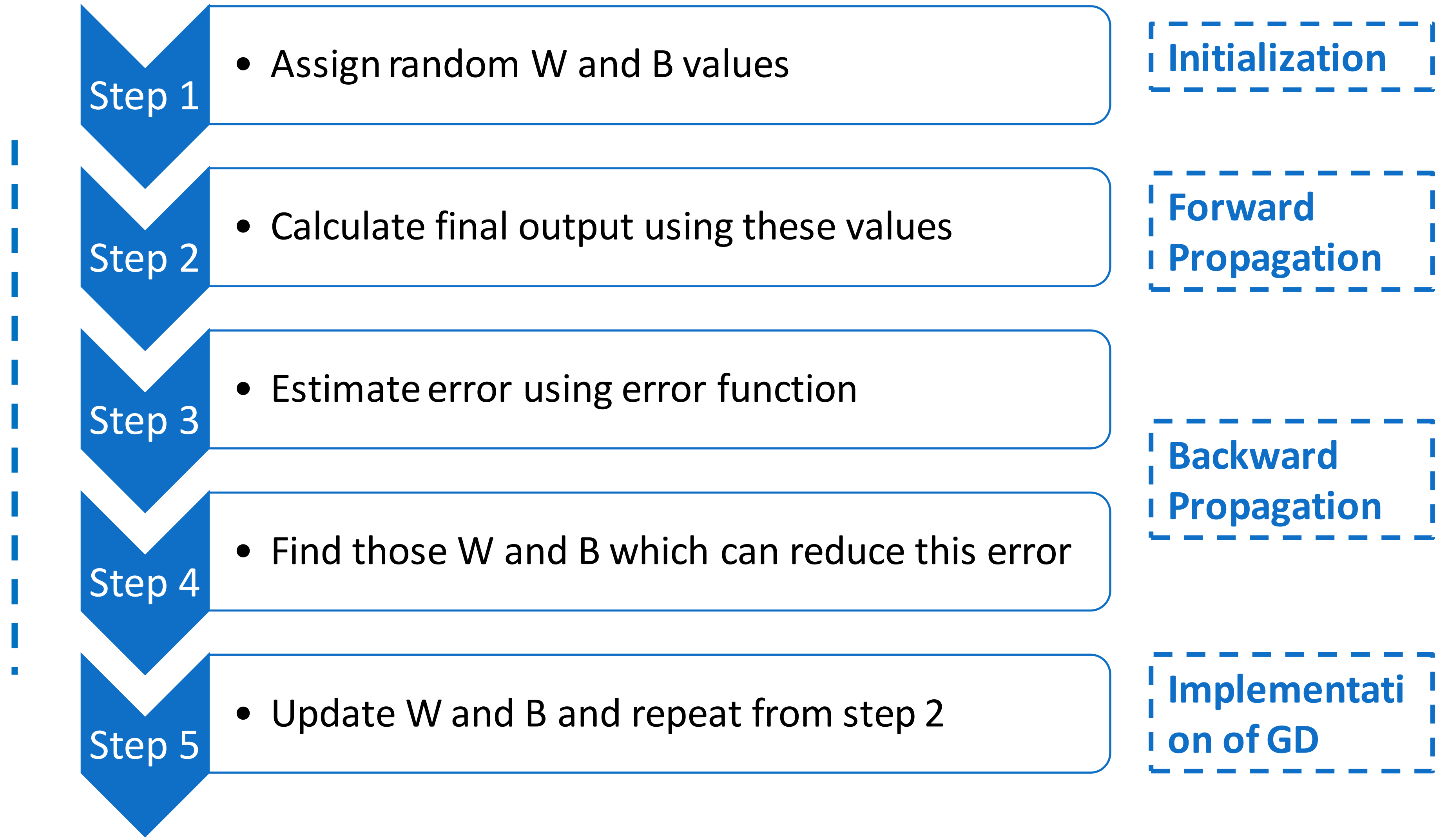
Gradient Descent



- 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

Process



Step 1
- Assign random W and B values

Initialization

Step 2
- Calculate final output using these values

Forward Propagation

Step 3
- Estimate error using error function

Step 4
- Find those W and B which can reduce this error

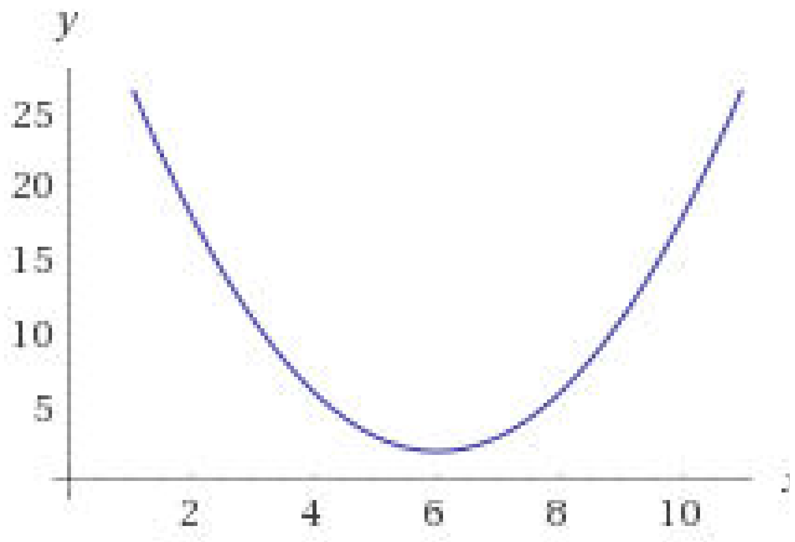
Backward Propagation

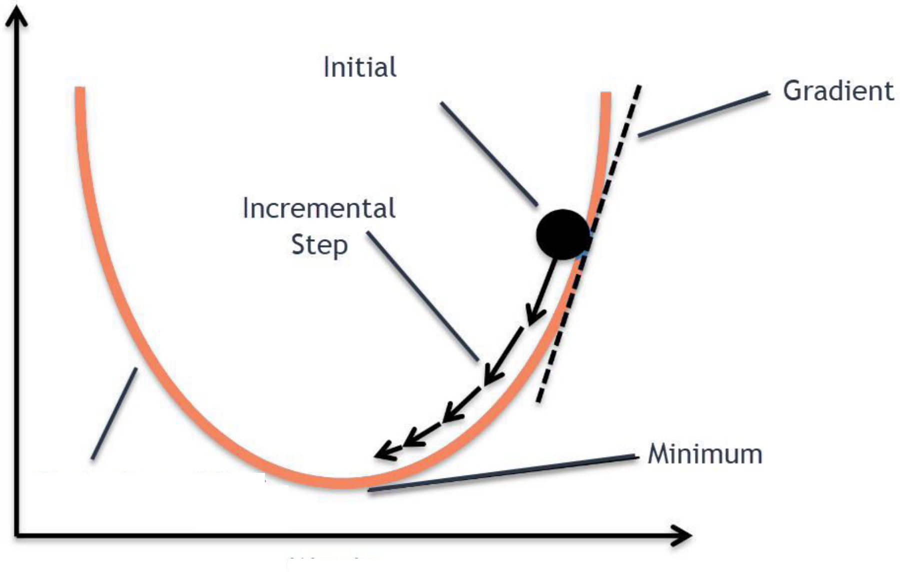
Step 5
- Update W and B and repeat from step 2

Implementati on of GD

Neural Network

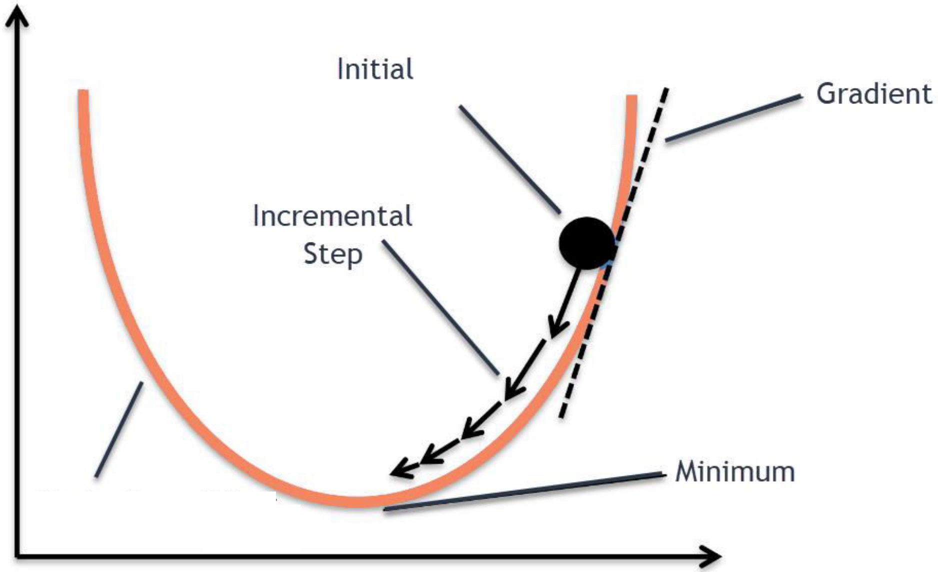
Gradient Descent





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

Gradient

Descent

Gradient Descent