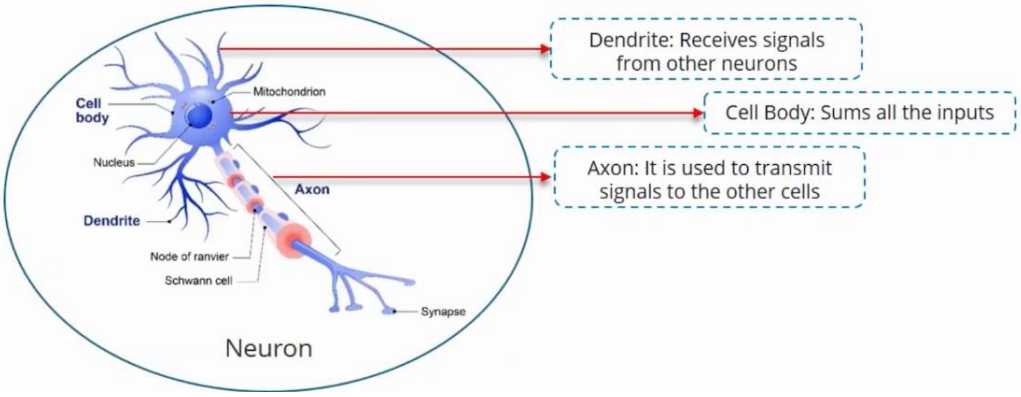
Lecture 2

Agenda

- What Is Deep Learning?
- How Deep Learning Works?
- Single Layer Perceptron (Early Deep Learning Models)
- Single Layer Perceptron Examples
- Limitations Of Single Layer Perceptron
- Multi Layer Perceptron
- Multi Layer Perceptron Examples

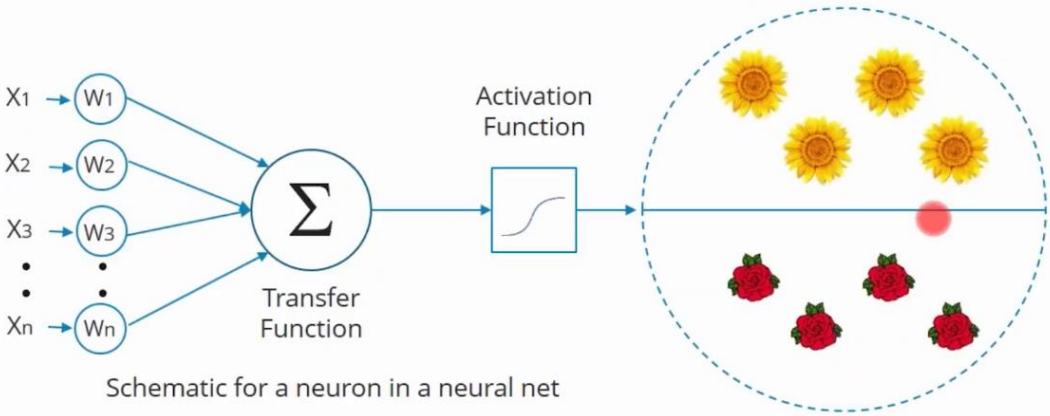
How Deep Learning Works?

Deep learning is a form of machine learning that uses a model of computing that's very much inspired by the structure of the brain, so lets understand that first.



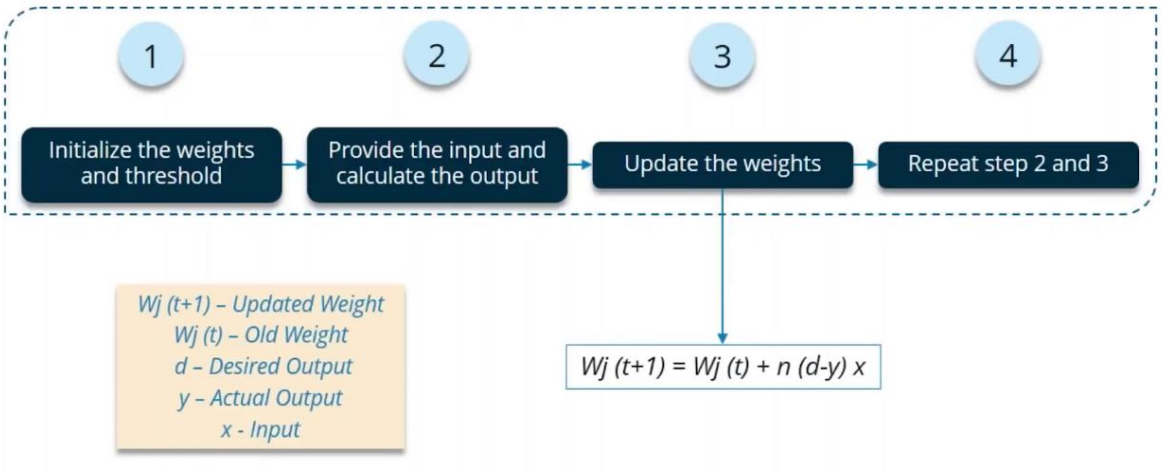
Perceptron Learning Algorithm

To get started, I'll explain a type of artificial neuron called a *perceptron*.



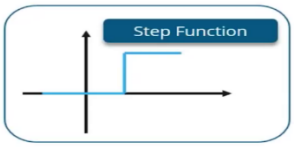
Schematic for a neuron in a neural net

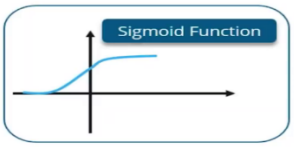
Perceptron Learning Algorithm

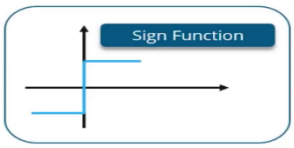


Note: Perceptron can be used to classify the data which is linearly separable.

Activation Function







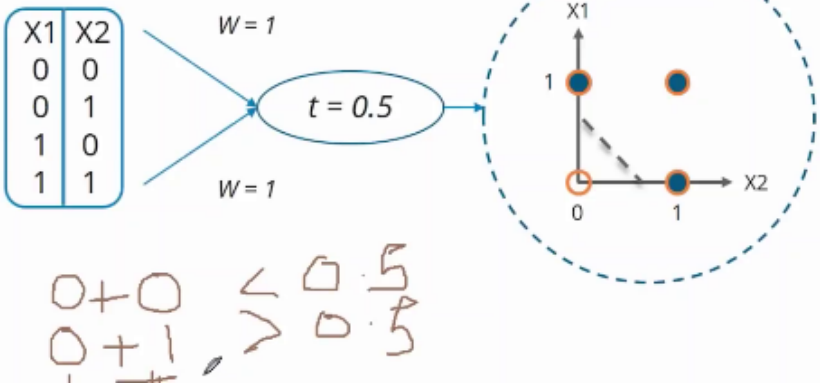
Applications

It can be used to implement Logic Gates.

OR

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

X1	X2
0	0
0	1
1	0
1	1



0+0 < 0.5
0+1 > 0.5

What Is Tensorflow?

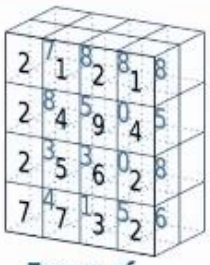
- Tensors are the standard way of representing data in deep learning.
- Tensors are just multidimensional arrays, an extension of two-dimensional tables (matrices) to data with higher dimension.

't'
'e'
'n'
's'
'o'
'r'

Tensor of dimensions[6]

3	1	4	1
5	9	2	6
5	3	5	8
9	7	9	3
2	3	8	4
6	2	6	4

Tensor of dimensions[6,4]



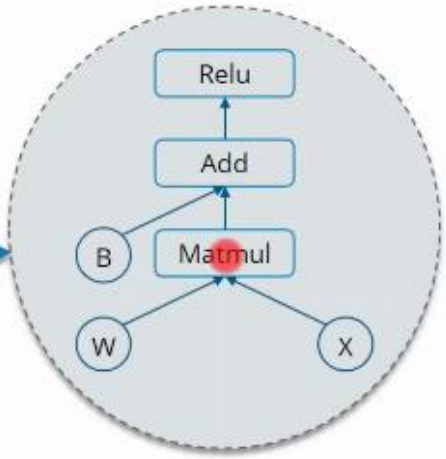
Tensor of dimensions[6,4,2]

In Tensorflow, computation is approached as a dataflow graph

Tensor

Flow

3.2	-1.4	5.1	...
-1.0	-2	2.4	...
...	...	...	...
...	...	...	...



TensorFlow Code Basics

A **Computational graph** is a series of Tensorflow operations arranged into a graph of nodes.

To actually evaluate the nodes, we must run the computational graph within a session.

A **session** encapsulates the control and state of the Tensorflow runtime.

Constant: It takes no input and it outputs values it stores internally.

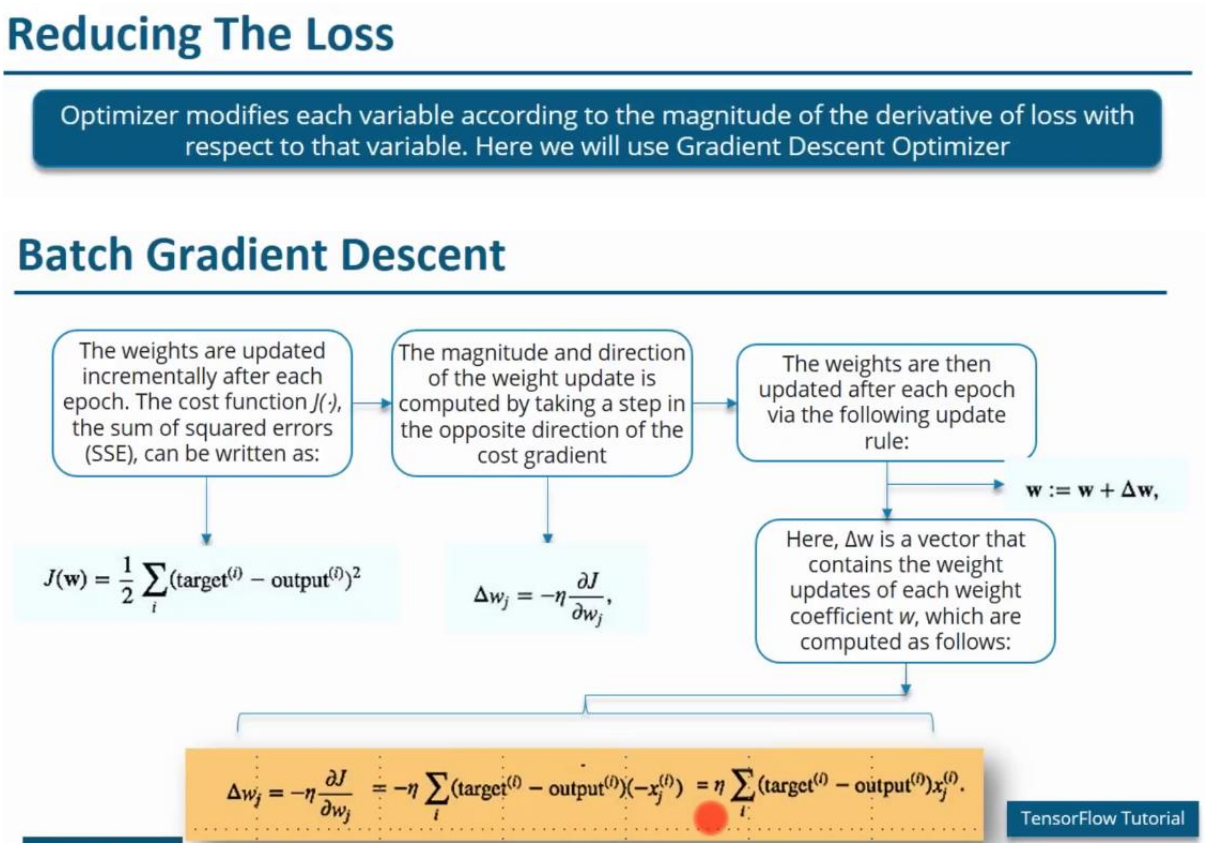
A graph can be parameterized to accept external inputs, known as **placeholders.**

Placeholder is a promise to provide a value later.

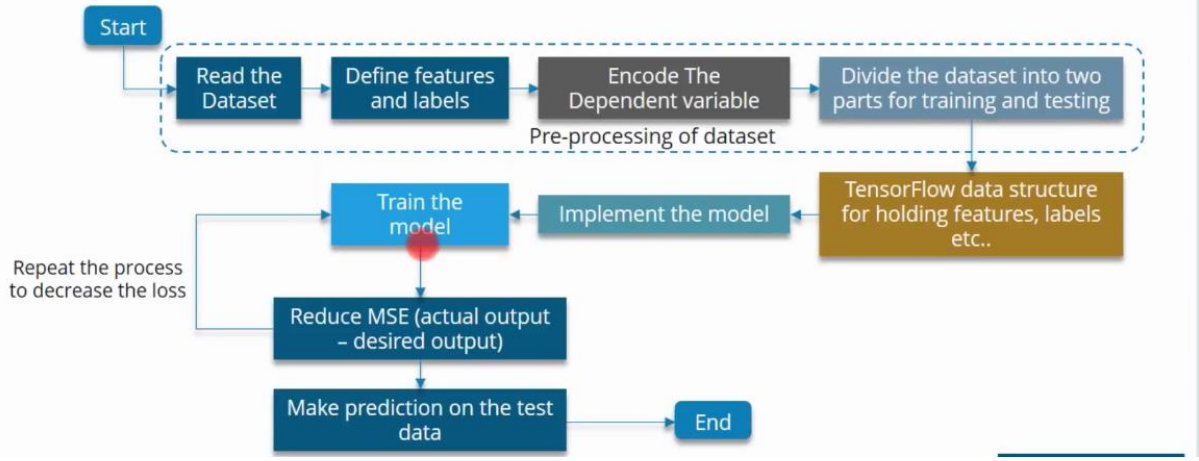
To make the model trainable, we need to able to modify the graph to get new outputs with the same input. **Variables** allow us to add trainable parameters to a graph.

A loss function measures how apart the current model is from the provided data.

Tensorflow provides optimizers that slowly change each variable in order to minimize the loss function.

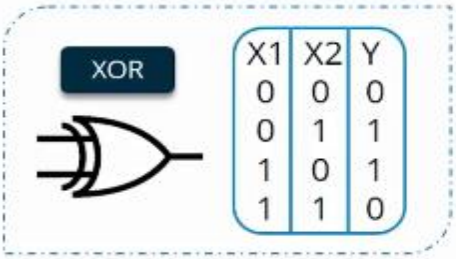


Implementation Of The Use-Case

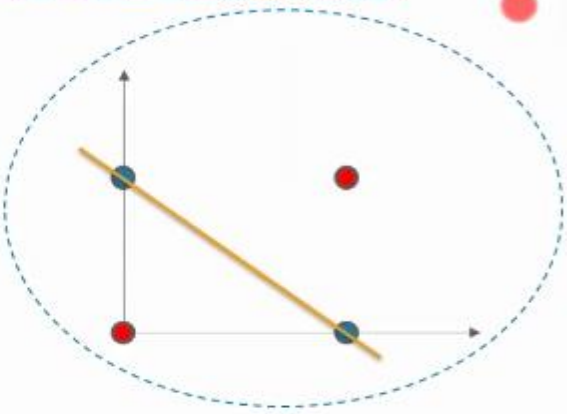


Limitations Of Single Layer Perceptron

Let us understand this with an example:
How can I implement an XOR gate using Single Layer Perceptron?

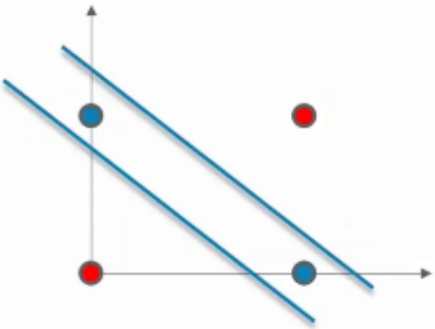


X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0



Limitations Of Single Layer Perceptron

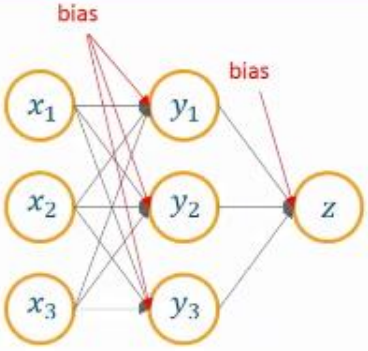
Now, how will I separate the two set of outputs?

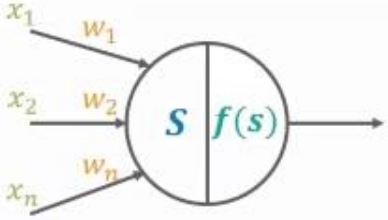


For Solving this problem, a Multilayer Perceptron with backpropagation can be used

Multilayer Perceptron

A Multi-layer Perceptron has the same structure of a single layer perceptron but with one or more hidden layers and is thus considered a deep neural network.





Summation:

$$S = \sum_{i=1}^{n} w_i * x_i$$

Transformation:

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

Note:Backpropogation skipped