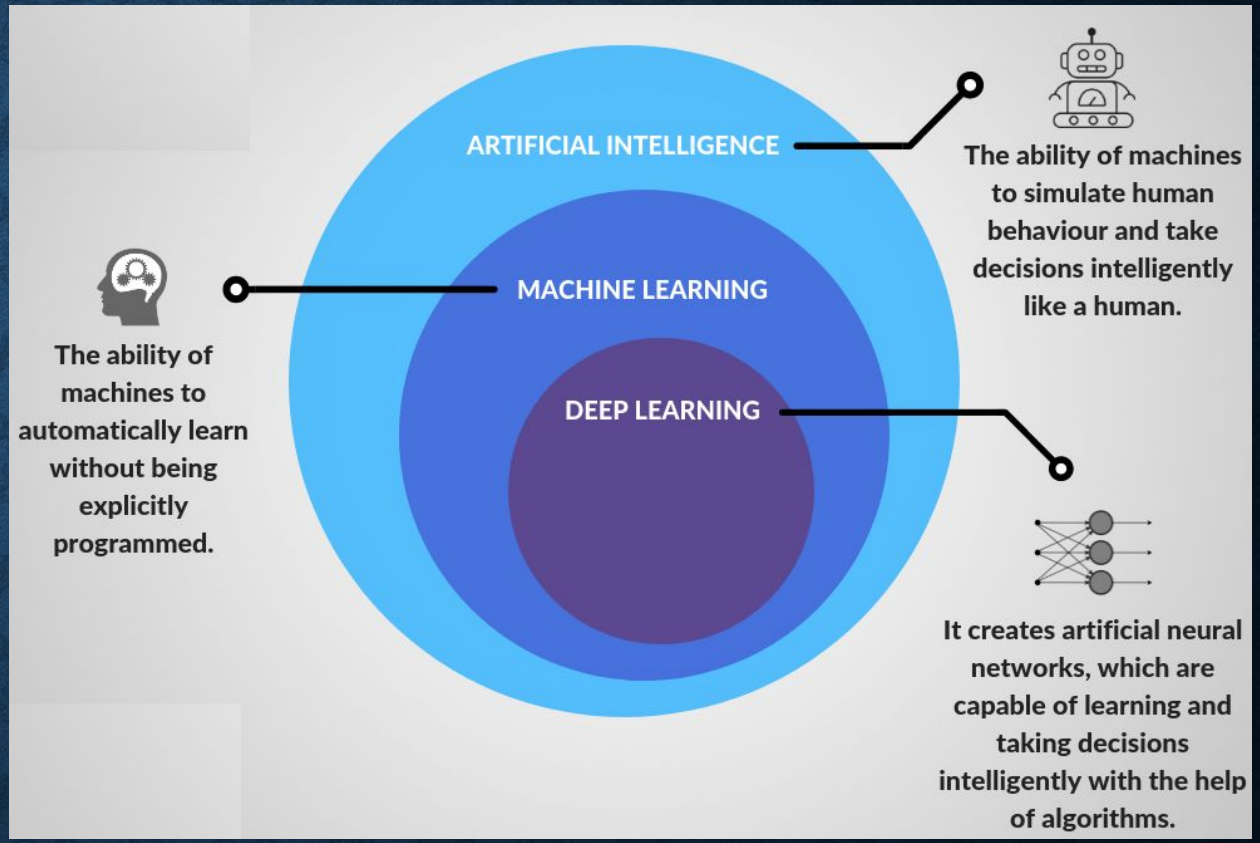
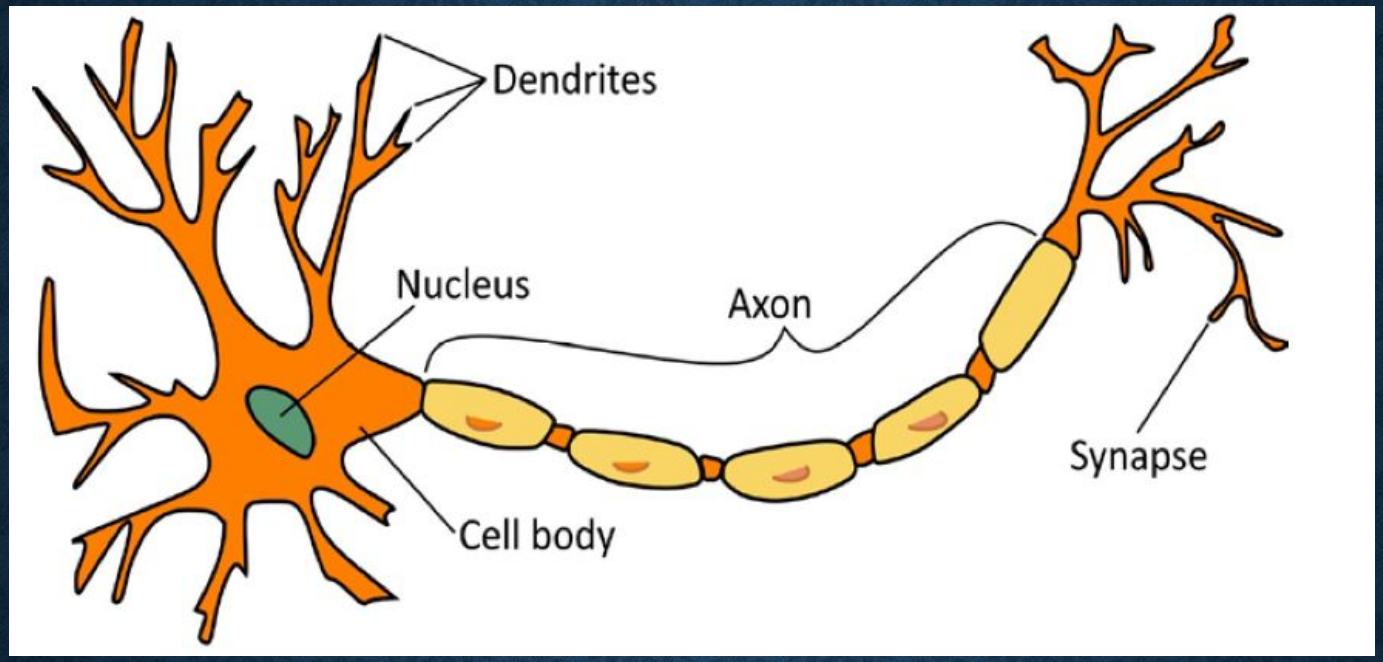
# Introduction to Deep learning

- To understand Deep Reinforcement Learning
- Deep learning is a subset of Machine Learning
- Deep Learning is computationally Advanced
- Ability to work on huge volume of data and reliability

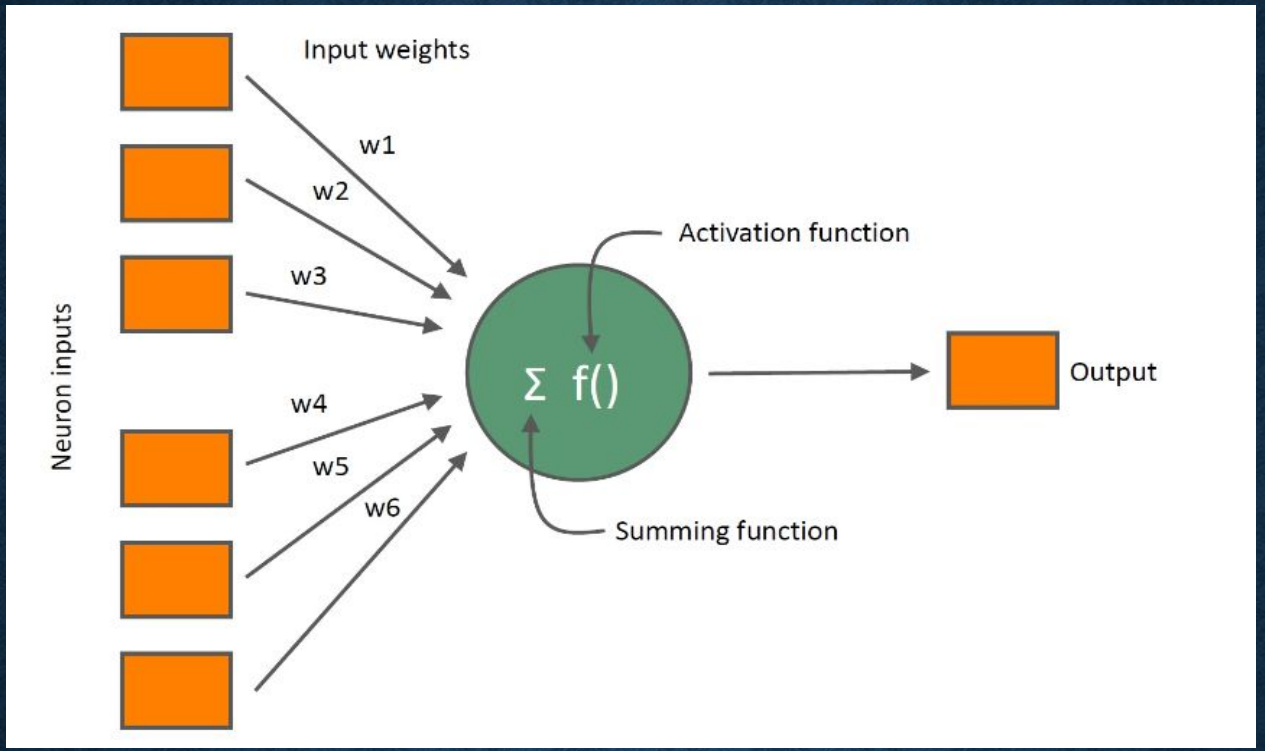


# Biological Neuron




- Neurons are the fundamental units of brain and nervous system
- Human Brain encompasses approximately 100 billion system
- ‘Synapse’ - Receiving input , sending motor instructions to muscles
- ‘Dendrites’ – Receiving input
- Body cell known as ‘Nucleus’ or ‘Soma’ where inputs from ‘Dendrites’ given weightage and stored in
- Instructions from cell body are passed through ‘Axon’ to other neurons

# Artificial Neuron




- Neuron inputs are multiplied by weights and summed together and add bias,

$$z = (x1.w1 + x2.w2 + x3.w3 + x4.w4 + x5.w5 + x6.w6 ) + b$$

- This looks familiar to equation of a straight line,

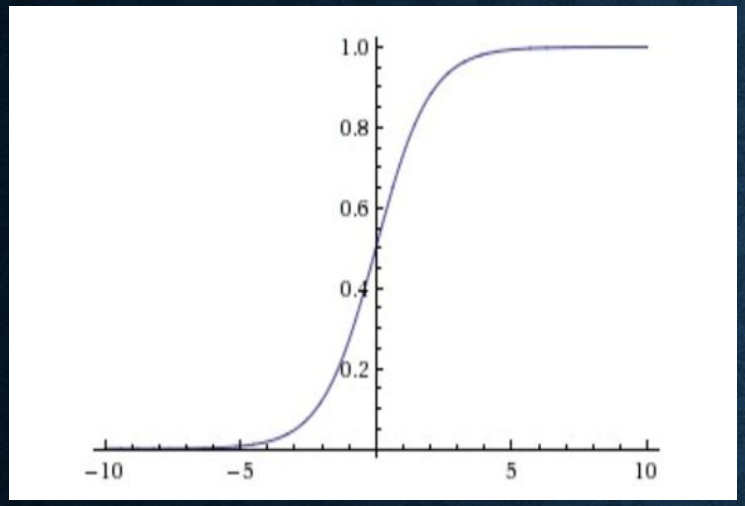
$$y = mx + c$$

- Introduce non – linearity by applying activation or transfer function on z to have output

$$z' = f(z)$$

# Artificial Neuron - Intuition

Consider Sigmoid Activation Function, squash the values in range (0,1)



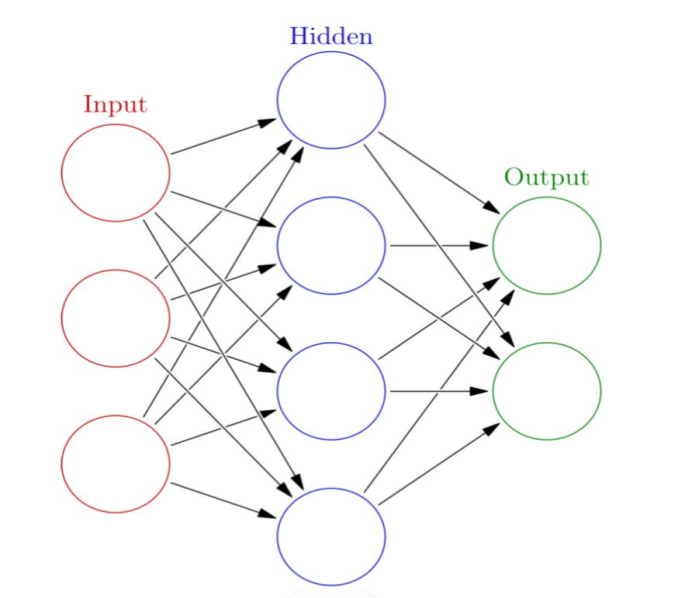

$$\sigma(x) = 1/(1 + e^{-x})$$

| | | | | |
|---|---|---|---|---|
| x1 | 0.2 | w1 | 0.2 | Bias = 0.2 |
| x2 | 0.3 | w2 | 0.3 | |
| x3 | 0.1 | w3 | 0.2 | |
| x4 | 0.4 | w4 | 0.4 | |
| x5 | 0.5 | w5 | 0.3 | |
| x6 | 0.6 | w6 | 0.5 | |

Z = (x1.w1 + x2.w2 + x3.w3 + x4.w4 + x5.w5 + x6.w6 )+ b
= ( 0.2 x 0.2 + 0.3 x 0.3 + 0.1 x 0.2 + 0.4 x 0.4 + 0.5 x 0.3 + 0.6 x 0.5 ) + 0.2
= (0.04 + 0.09 + 0.02 + 0.16 + 0.15 + 0.3) + 0.2
= 0.96

Sigmoid(Z) >> 0.276

# Artificial Neural Network




Input layer:

- Number of Neurons in the input layer is the number of inputs feeding to the network.
- No computation performed in input layer. It is just used for passing information from outside to the network.
- Each inputs will have some influence on predicting the output.

Hidden layer:

- Any layer between the input and out layer is hidden layer
- Hidden layer identifies patterns in the data
- Majorly responsible for deriving complex relationship between input and output and also for learning the data representation and for extracting the features
- Deep Neural Network have many number of hidden layers

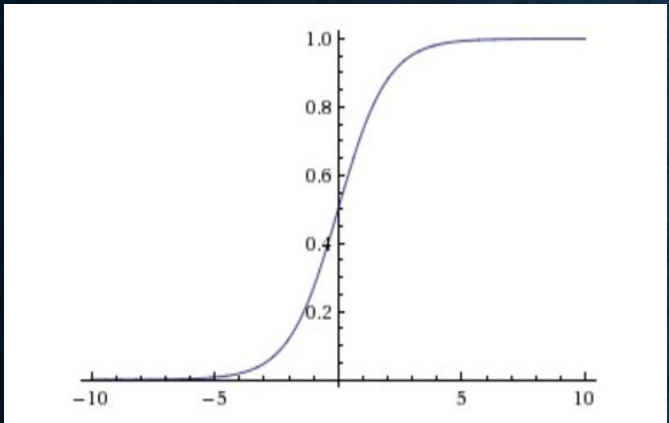
Output layer:

- After processing the input, the hidden layer sends its result to the output layer.
- Number of neurons in the output layer is based on type of problem network will solve.
- For binary Classification – one neuron at output layer
- For Multiclass Classification – Number of class is number of neuron at output layer
- For Regression Problem - One neuron in the output layer

# Exploring Commonly Used Activation Functions:

- Activation or Transfer function is used to introduce non-linearity in neural network to learn the complex underlying patterns in the data. Without this, Neural Network resembles the linear regression.

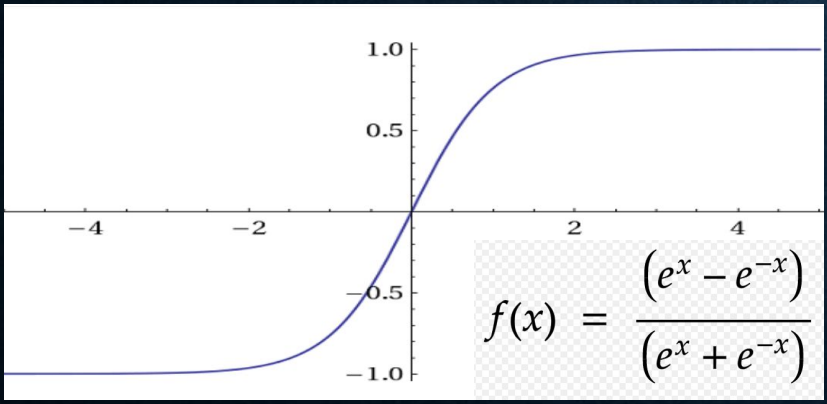
## The Sigmoid Function:




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

- Also known as Logistic Function
- Used for predicting the output as it squash the values in the probability range (0,1)
- If the value less than 0.5 , then neuron won't be activated
- If the value greater than 0.5 , then the neuron will be activated

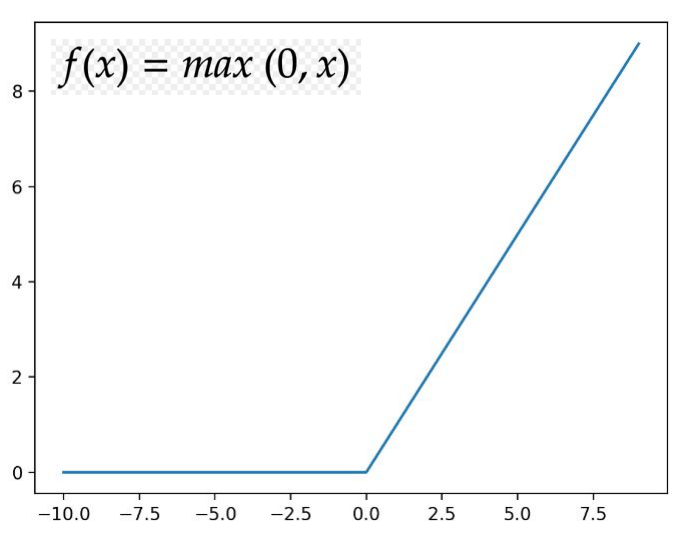
## The tanh Function:




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

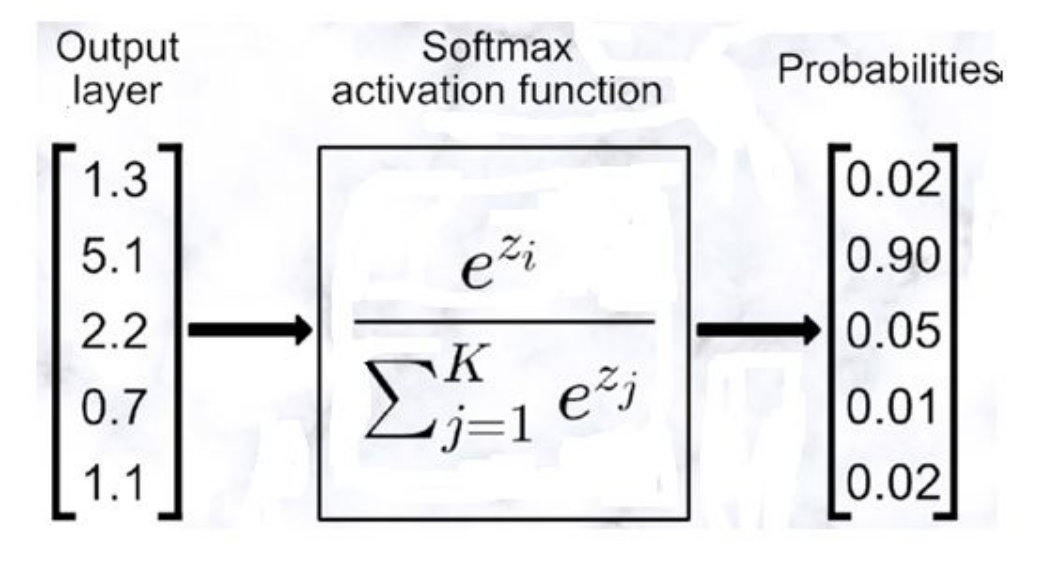
- Also known as a hyperbolic tangent function
- Outputs the value between -1 to +1
- '0' cantered
- If the value is less than 0, then the neuron won't be activated
- If the value greater than 0, then the neuron will be activated

## The Rectified Unit function (ReLU) :




- ReLU outputs a value from zero to infinity
- f(x) returns zero when the value of x is less than zero
- f(x) returns x when the value of x is greater than or equal to zero
- Neuron won't be activated when f(x) = 0
- Neuron will be activated when f(x) = x

## The SoftMax Function:




- SoftMax is basically the generalization of the sigmoid function
- It is usually applied to the final layer of the network while performing multi class classification tasks
- It gives probabilities of each class for being output
- The sum of the SoftMax values will always equals to 1