Start-Tech Academy

# Neural Network

**Activation Function**

Q – Why do we use activation functions

$$w_1 x_1 + w_2 x_2 + b_1 = z$$

$$w_5 a_1 + w_6 a_2 + b_3 = z_3$$

x1

x2

y
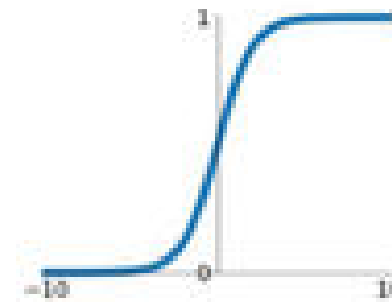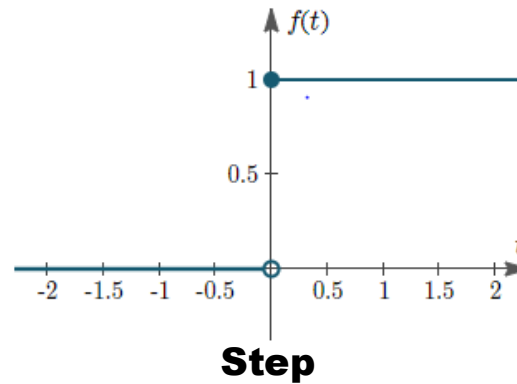
$$w_3 x_1 + w_4 x_2 + b_2 = z_2$$
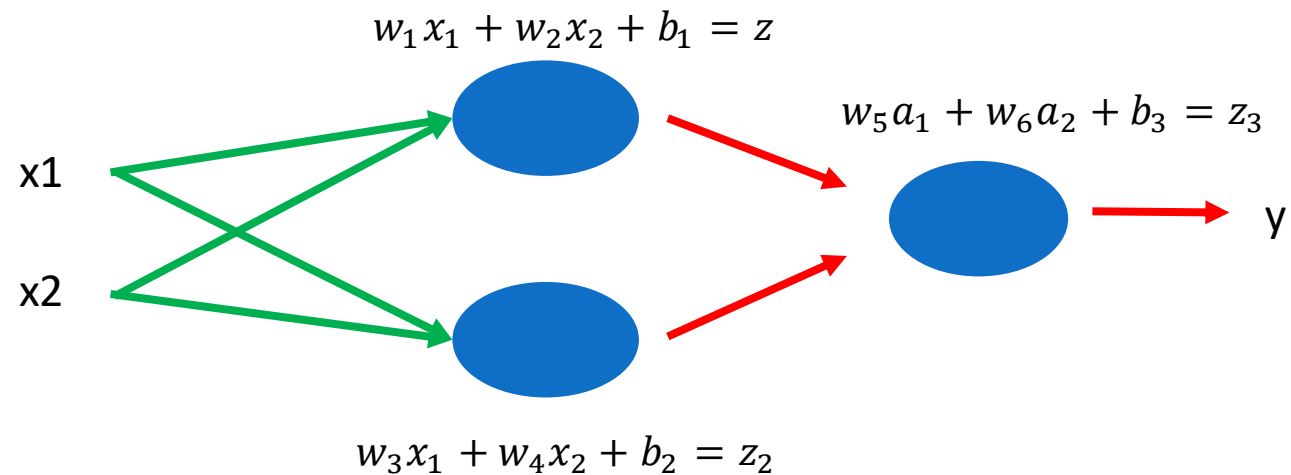
Ans

- To put special boundary conditions on the output

- To introduce non linearity and find complex patterns

# Neural Network

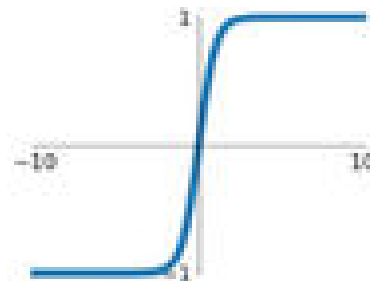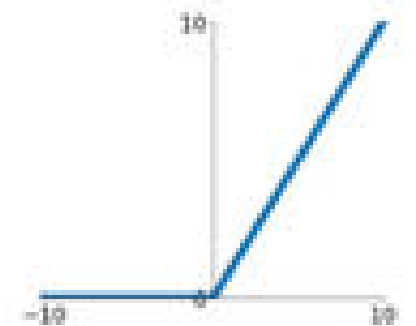Q – What are the different types of activation functions

**Activation Function**



**Step**



**Sigmoid**

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



**tanh**

$$\tanh(x)$$



**ReLU**

$$\max(0, x)$$

# Neural Network

**Activation Function**

Q – What are the different types of activation functions

| Function | Upper Boundary | Lower Boundary | Class /Reg | Layer |
|---|---|---|---|---|
| Step | 1 | 0 | Classification | Mostly Output |
| Sigmoid | 1 | 0 | Classification | Hidden & Output |
| Hyperbolic Tangent (TanH) | 1 | -1 | Classification | Hidden & Output |
| Rectified Linear Unit (ReLU) | 0 | infinity | Regression/ classification | Hidden |

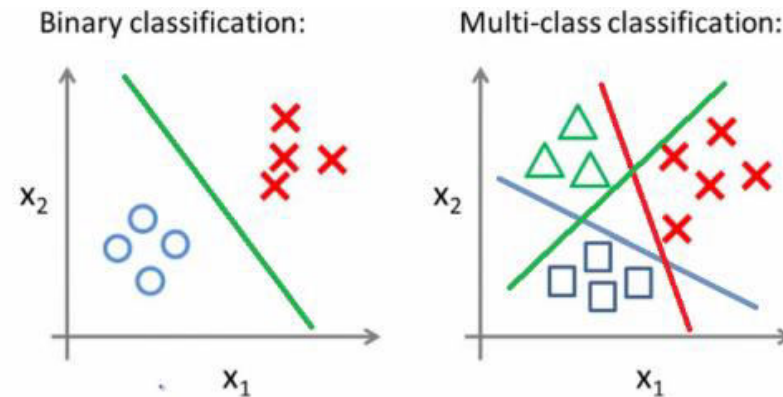# Neural Network

**Activation Function**

Q – Can Hidden layers and output layers have different activation functions?

Ans - Yes

# Neural Network

**Activation Function**

Q – What is multi class classification? Is there any specific activation function for this?



Ans

- Two classes like 'Yes' or 'No' => Binary Classification

- More than 2 classes like 'shirts', 'trousers' or 'socks' => Multiclass classification

- For multiclass, we use softmax activation

**Activation Function**

Q – What is multi class classification? Is there any specific activation function for this?
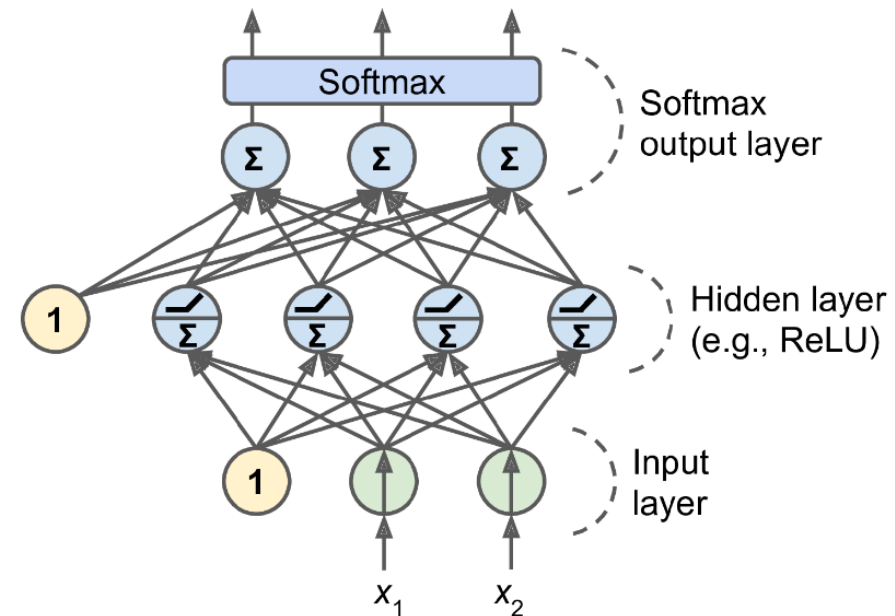


Ans

- For each class we keep one output neuron with sigmoid activation

- All the outputs go into softmax layer where each output is divided by the total sum to bring the total probability to one

# Neural Network

**Gradient descent**

Q – What is the difference between Gradient descent and stochastic gradient descent

- Stochastic gradient descent => Single training record, forward and backward propagation

- Gradient descent => Full training set, forward and backward propagation

- Mini Batch Gradient descent => small batch of training set, forward and backward propagation
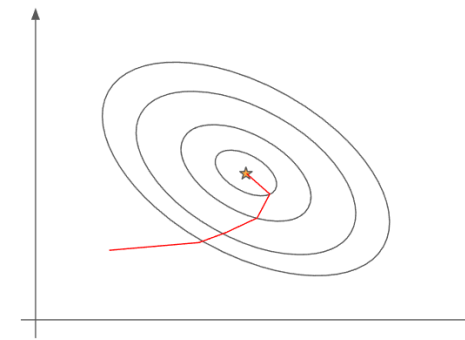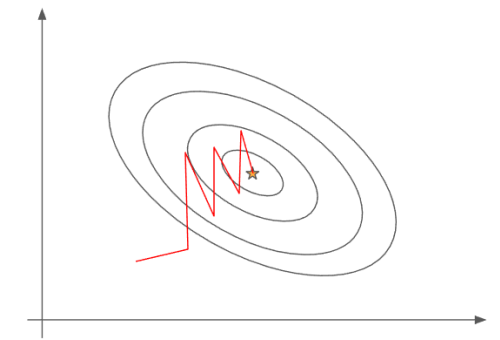


Gradient Descent                                 Stochastic Gradient Descent

# Neural Network

**Epoch**

Q – What is an Epoch

- Epoch is one cycle through the full training data

- It is different from iteration

- Example – Suppose we have 1000 training records, if we are doing SGD i.e. one record is input at a time, then 1000 iterations within one epoch

- If we enter 1000 records 2 time => Epoch is 2

# Neural Network

**Classification Hyperparameters**

| Hyperparameter | Typical value |
|---|---|
| # input neurons | One per input feature |
| # hidden layers | Depends on the problem, but typically 1 to 5 |
| Hidden activation | ReLU |

| Hyperparameter | Binary classification | Multilabel binary classification | Multiclass classification |
|---|---|---|---|
| # output neurons | 1 | 1 per label | 1 per class |
| Output layer activation | Logistic | Logistic | Softmax |
| Loss function | Cross entropy | Cross entropy | Cross entropy |

# Neural Network

**Regression Hyperparameters**

| Hyperparameter | Typical value |
|---|---|
| # input neurons | One per input feature |
| # hidden layers | Depends on the problem, but typically 1 to 5 |
| # neurons per hidden layer | Depends on the problem, but typically 10 to 100 |
| # output neurons | 1 per prediction dimension |
| Hidden activation | ReLU |
| Output activation | None |
| Loss function | MSE |

# Neural Network

**Keras & Tensorflow**

Keras is a model-level library, providing high-level building blocks for developing deep-learning models

# Neural Network

**Keras & Tensorflow**

**28 * 28 pixel**

Input Layer

Hidden Layer 1

Hidden Layer 2

Output Layer

**ReLu Activation**

**Softmax Activation**

**10 Categories**