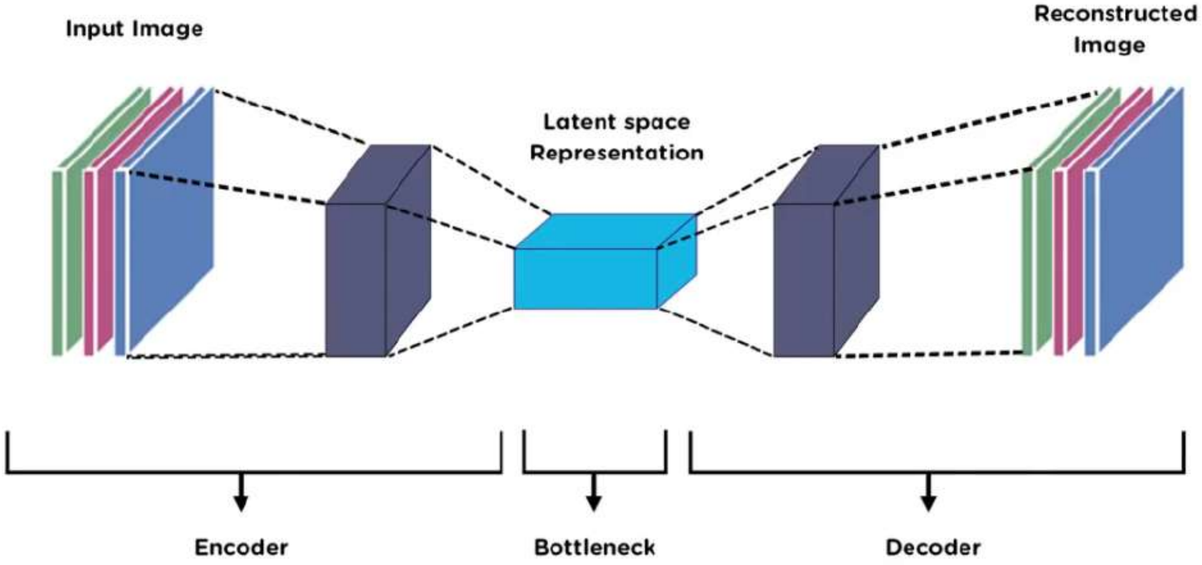
29

What is an auto-encoder?



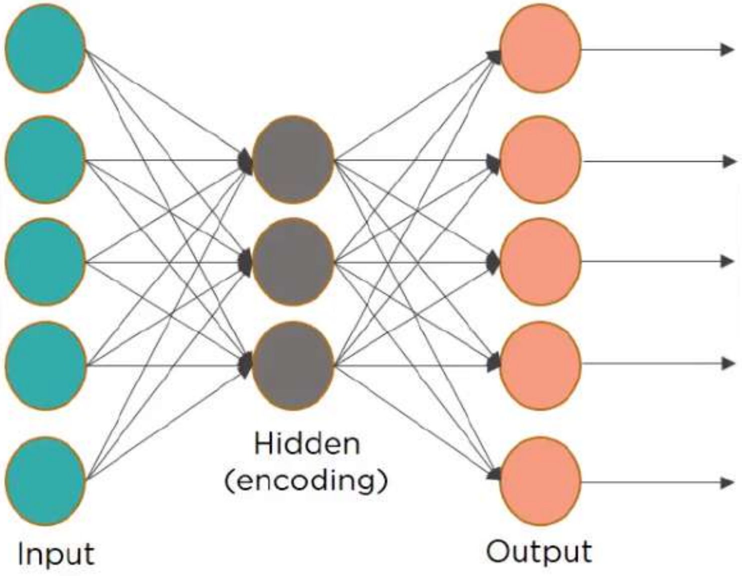
It works by compressing the input to a latent-space representation and then reconstructing the output from this representation.

29

What is an auto-encoder?

The network is trained to reconstruct its inputs. Here, the input neurons are equal to the output neurons

It is a neural network that has 3 layers

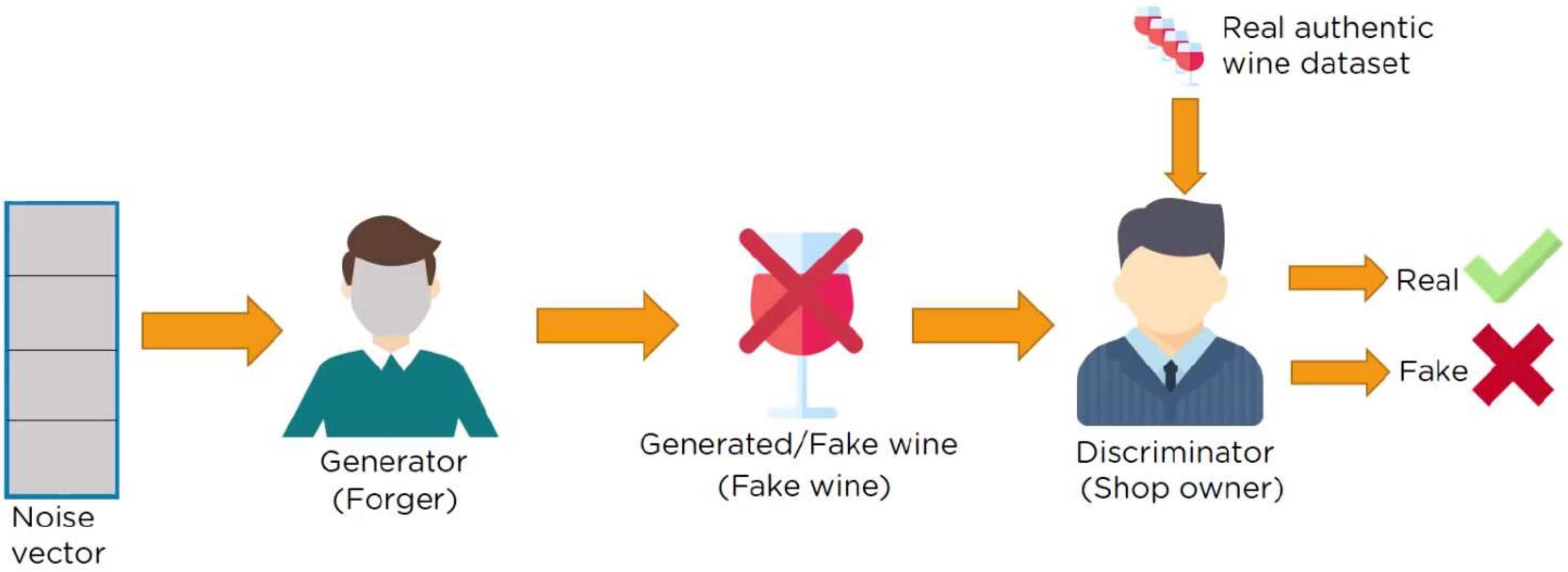


The network's target output is same as the input. It uses dimensionality reduction to restructure the input.

Outputs are same as inputs

28 Explain Generative Adversarial Network along with an example.

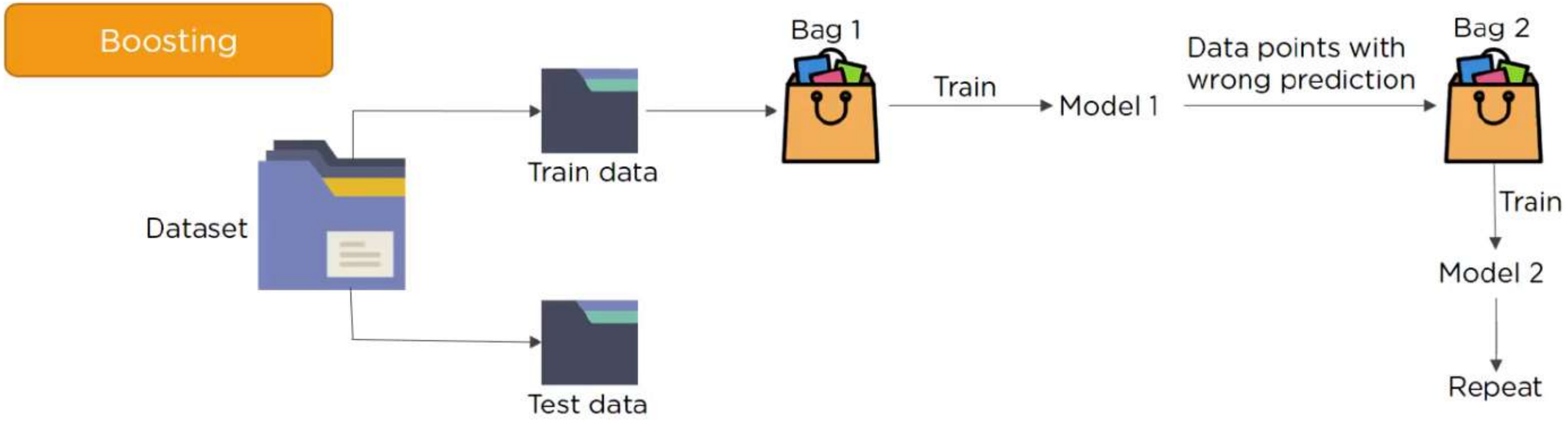
There are 2 main components of GAN: Generator and Discriminator



30

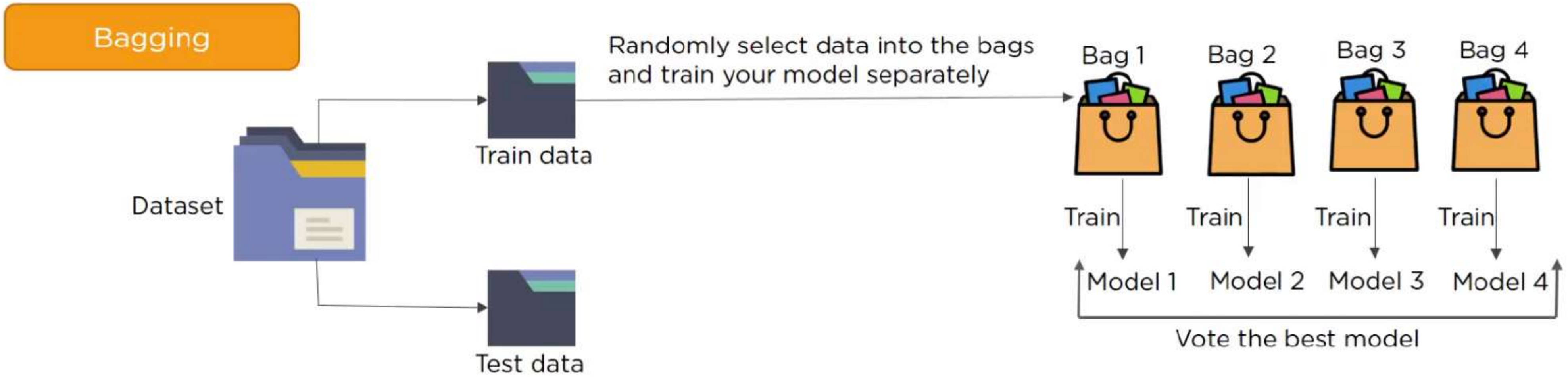
What is Bagging and Boosting?

- ❑ In Boosting, the emphasis is to select the data points which give wrong output in order the improve the accuracy



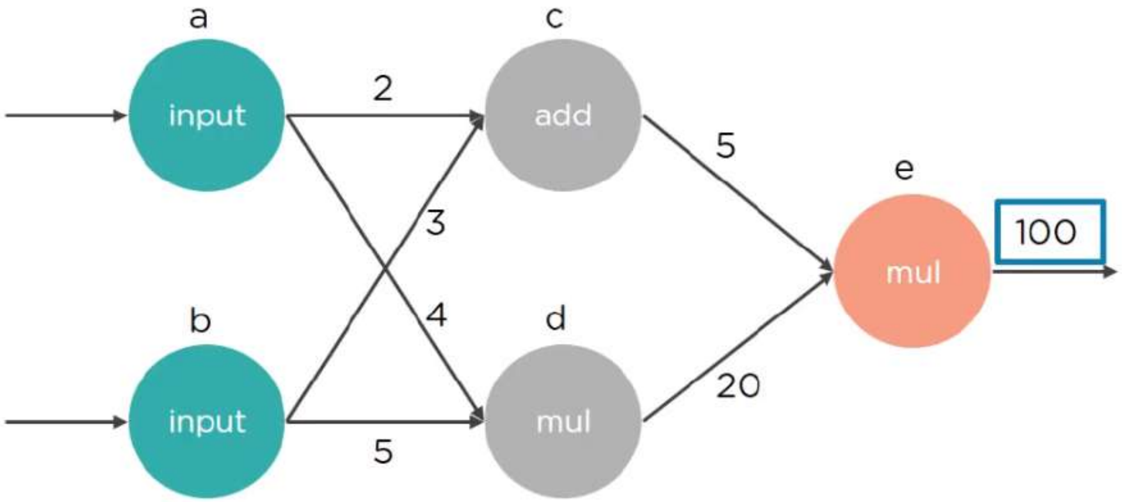
30 What is Bagging and Boosting?

❑ Bagging and Boosting are ensemble techniques where the idea is to train multiple models using the same learning algorithm and then take a call



27

What do you understand by a Computational Graph?



Nodes ----------> Mathematical operation
Edges ----------> Data flow

27 What do you understand by a Computational Graph?

- Everything in TensorFlow is based on creating a computational graph
- It has a network of nodes where each node performs an operation
- Nodes represent mathematical operation and edges represent tensors
- Since data flows in the form a graph, it is also called as DataFlow Graph

26

What are the programming elements in TensorFlow?

Placeholders

Placeholders allow us to feed data to a tensorflow model from outside a model. It permits a value to be assigned later. To define a placeholder, we use *tf.placeholder()* command.

Example:

```
a = tf.placeholder(tf.float32)
b = a*2
with tf.Session() as sess:
result = sess.run(b,feed_dict={a:3.0})
print result
```

Session

A *Session* is run to evaluate the nodes. This is called as the *TensorFlow runtime*

Example:

```
a = tf.constant(2.0)
b = tf.constant(4.0)
c = a+b
# Launch Session
sess = tf.Session()
# Evaluate the tensor c
print(sess.run(c))
```

26

What are the programming elements in TensorFlow?

Placeholders

Placeholders allow us to feed data to a tensorflow model from outside a model. It permits a value to be assigned later. To define a placeholder, we use *tf.placeholder()* command

Example:

```
a = tf.placeholder(tf.float32)
b = a*2
with tf.Session() as sess:
result = sess.run(b,feed_dict={a:3.0})
print result
```

Session

A *Session* is run to evaluate the nodes. This is called as the *TensorFlow runtime.*

Example:

```
a = tf.constant(2.0)
b = tf.constant(4.0)
c = a+b
# Launch Session
sess = tf.Session()
# Evaluate the tensor c
print(sess.run(c))
```

Files

26 What are the programming elements in TensorFlow?

Constants

Constants are parameters whose value does not change. To define a constant, we use *tf.constant()* command.

Example:

```
a = tf.constant(2.0, tf.float32)
b = tf.constant(3.0)
Print(a, b)
```

Variables

Variables allow us to add new trainable parameters to graph. To define a variable, we use *tf.Variable()* command and initialize them before running the graph in a session

Example:

```
W = tf.Variable([.3],dtype=tf.float32)
b = tf.Variable([-.3],dtype=tf.float32)
```

26 What are the programming elements in TensorFlow?

Placeholders

Placeholders allow us to feed data to a tensorflow model from outside a model. It permits a value to be assigned later. To define a placeholder, we use *tf.placeholder()* command

Example:

```
a = tf.placeholder(tf.float32)
b = a*2
with tf.Session() as sess:
result = sess.run(b,feed_dict={a:3.0})
print result
```

Session

A *Session* is run to evaluate the nodes. This is called as the *TensorFlow runtime*.

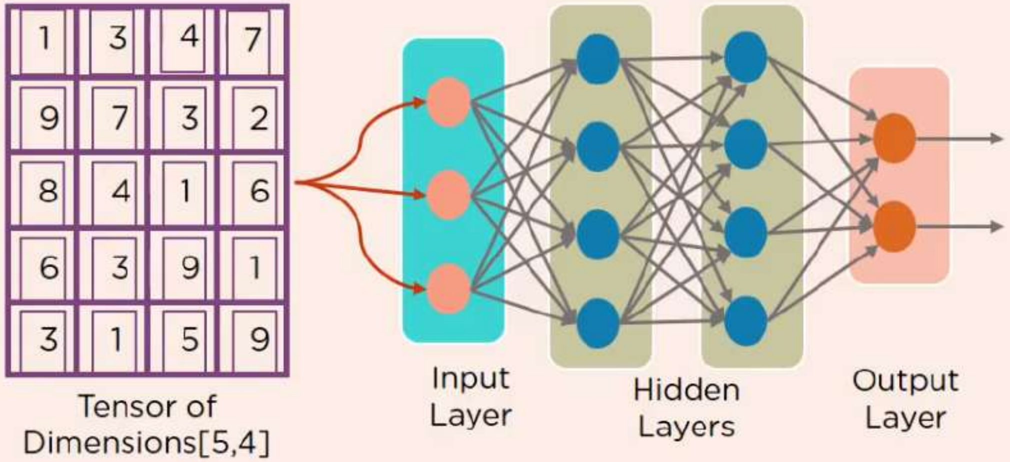
Example:

```
a = tf.constant(2.0)
b = tf.constant(4.0)
c = a+b
# Launch Session
sess = tf.Session()
# Evaluate the tensor c
print(sess.run(c))
```

25

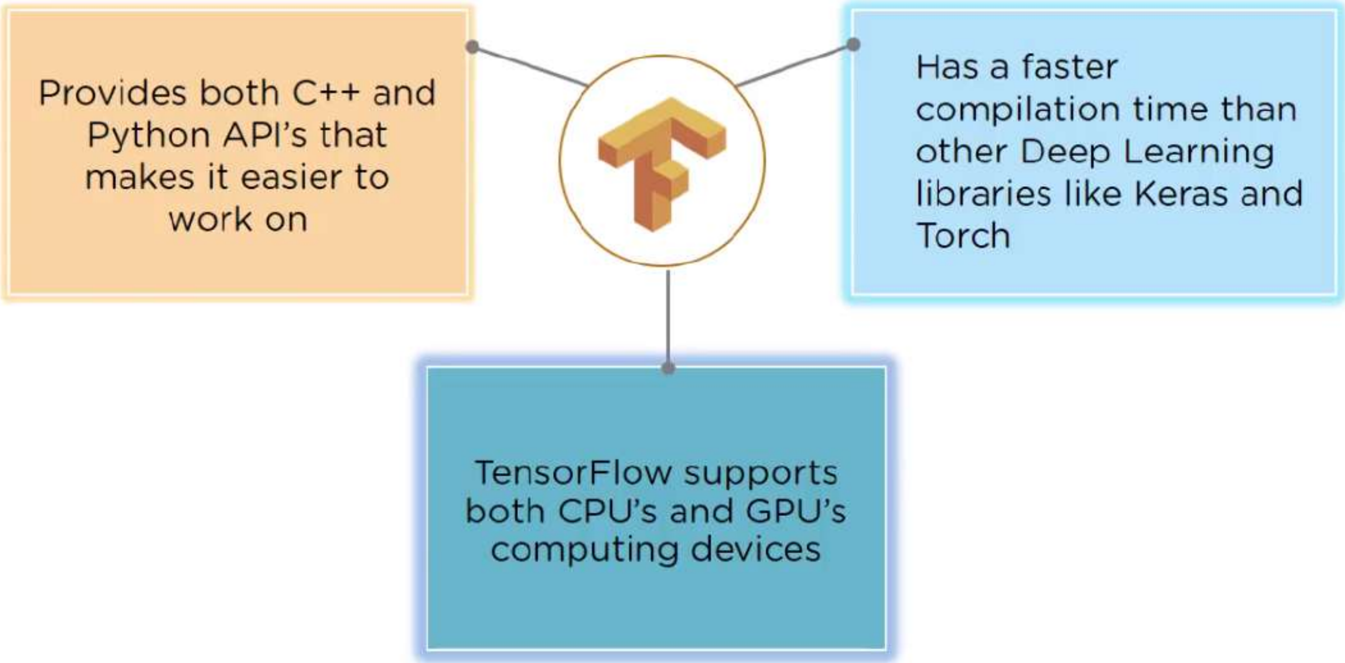
What do you mean by Tensor in TensorFlow?

Tensor is a mathematical object represented as arrays of higher dimensions. These arrays of data with different dimensions and ranks that are fed as input to the neural network are called ***Tensors***



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

24 Why TensorFlow is the most preferred library in Deep Learning?



Provides both C++ and Python API's that makes it easier to work on

Has a faster compilation time than other Deep Learning libraries like Keras and Torch

TensorFlow supports both CPU's and GPU's computing devices

23 What is the difference between Epoch, Batch and Iteration in Deep Learning?

Epoch

An Epoch represents one iteration over the entire dataset

Batch

We cannot pass the entire dataset into the neural network at once. So, we divide the dataset into number of batches

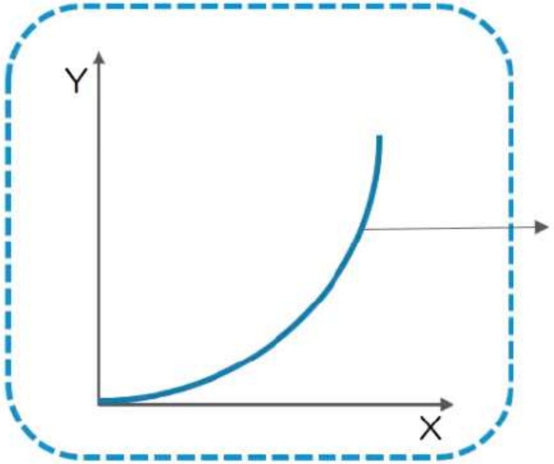
Iteration

If we have 10,000 images as data and a batch size of 200, then an epoch should run 10,000/200 = 50 iterations

22

What are Vanishing and Exploding gradients?

When the slope tends to grow exponentially instead of decaying, this problem is called *Exploding gradient*



Slope grows exponentially

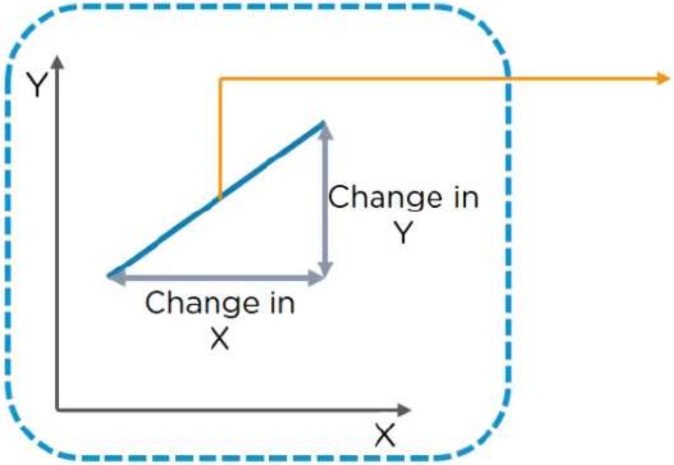
Issues in Gradient Problem

- Long training time
- Poor performance
- Low accuracy

22

What are Vanishing and Exploding gradients?

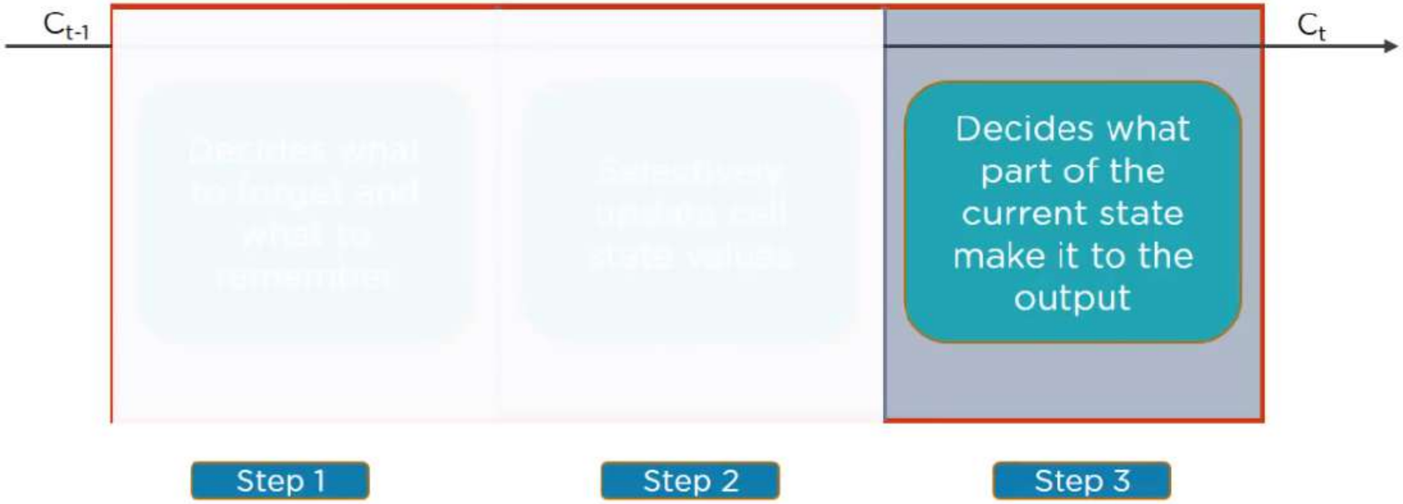
While training a RNN, your slope can become either too small or too large and this makes training difficult. When the slope is too small, the problem is know as ***Vanishing gradient***



Slope decreases gradually to a very small value (sometimes negative) and makes training difficult

21 How does LSTM network work?

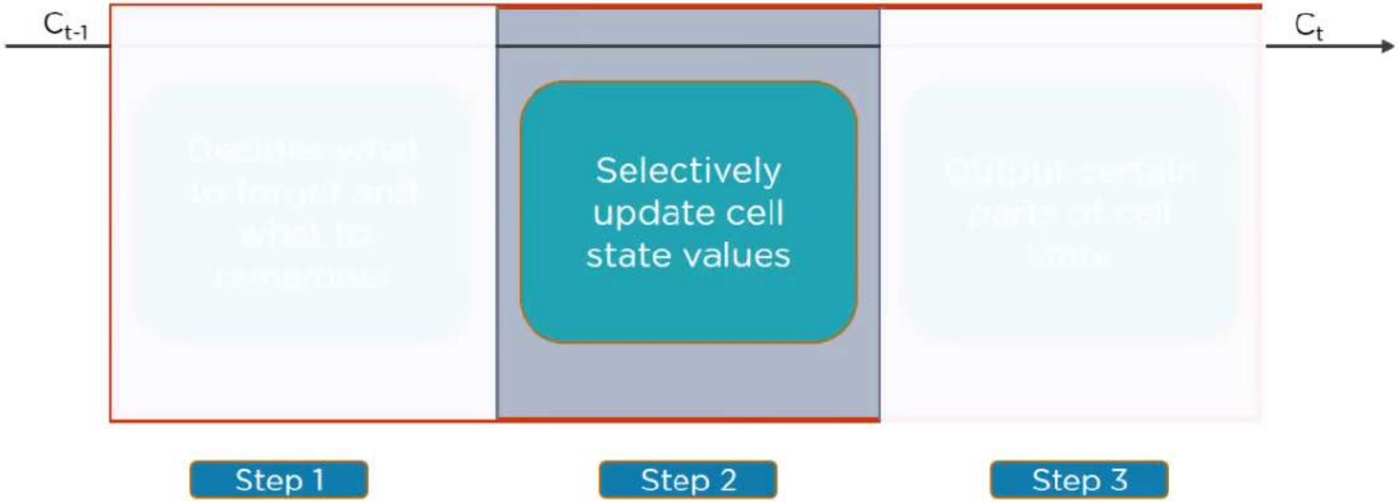
There are 3 steps in the process


C_{t-1}
C_t
Decides what to forget and what to remember
Selectively update cell state values
Decides what part of the current state make it to the output
Step 1
Step 2
Step 3

21

How does LSTM network work?

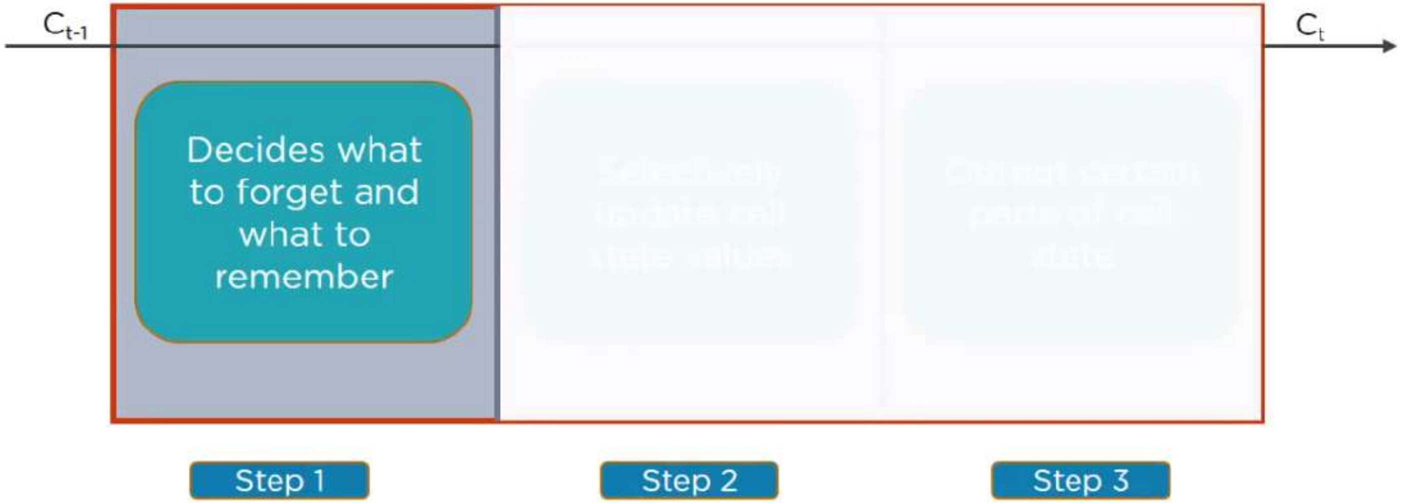
There are 3 steps in the process


C_{t-1}
Decides what to forget and what to remember
Selectively update cell state values
Output certain parts of cell state
C_t
Step 1
Step 2
Step 3

21

How does LSTM network work?

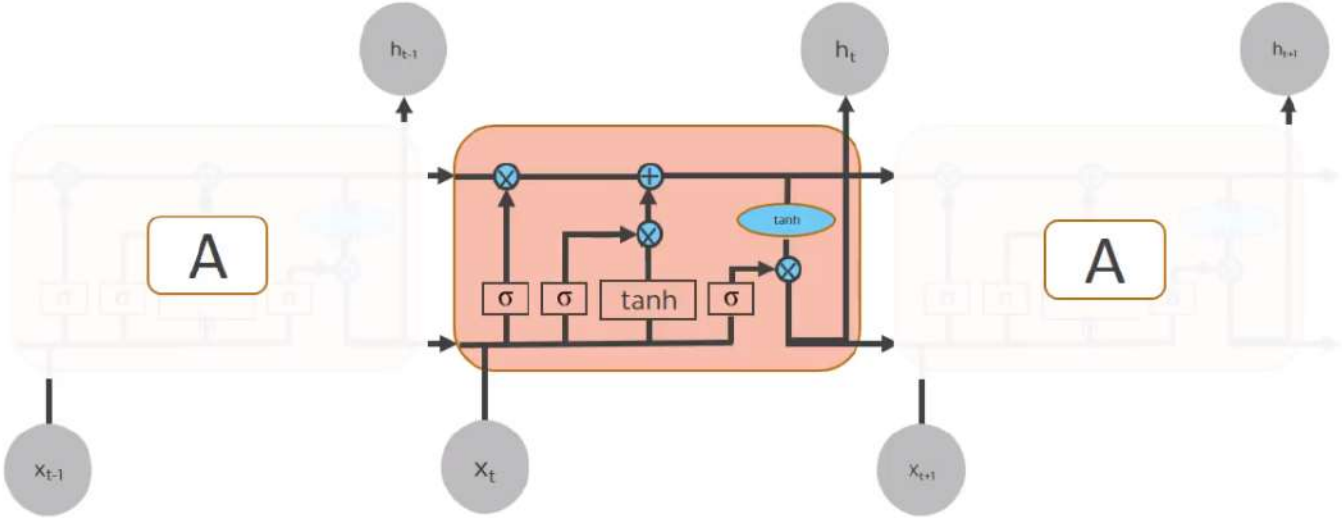
There are 3 steps in the process


C_{t-1}
C_t
Decides what to forget and what to remember
Selectively update cell state values
Output certain parts of cell state
Step 1
Step 2
Step 3

21

How does LSTM network work?

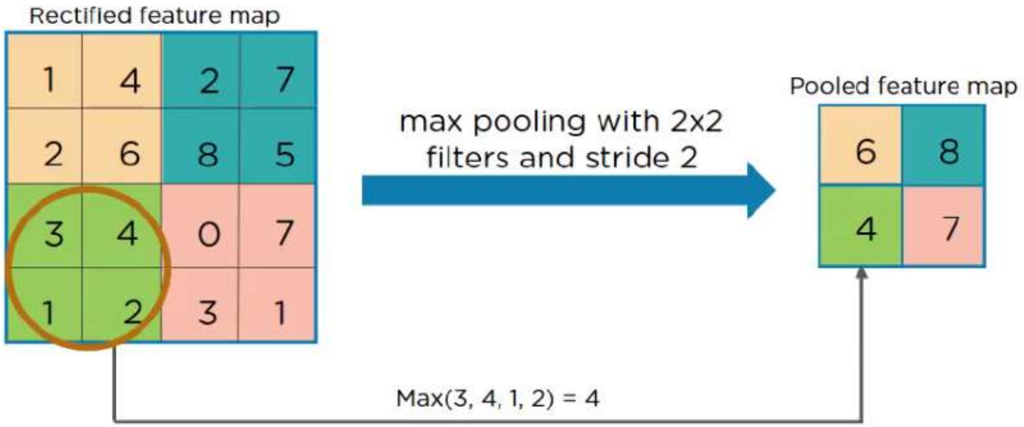
LSTMs are special kind of Recurrent Neural Networks, capable of learning long-term dependencies. Remembering information for long periods of time is their default behavior.



20 What is Pooling in CNN and how does it work?

Pooling

- Used to reduce the spatial dimensions of a CNN
- Performs down-sampling operation to reduce the dimensionality
- Creates a pooled feature map by sliding a filter matrix over the input matrix



Rectified feature map

1	4	2	7
2	6	8	5
3	4	0	7
1	2	3	1

max pooling with 2x2 filters and stride 2

Pooled feature map

6	8
4	7

Max(3, 4, 1, 2) = 4

19 What are the different layers in CNN?

1. Convolution Layer

Convolution layer that performs the convolution operation

2. ReLU Layer

ReLU brings non-linearity to the network and converts all the negative pixels to 0. Output is rectified feature map

3. Pooling Layer

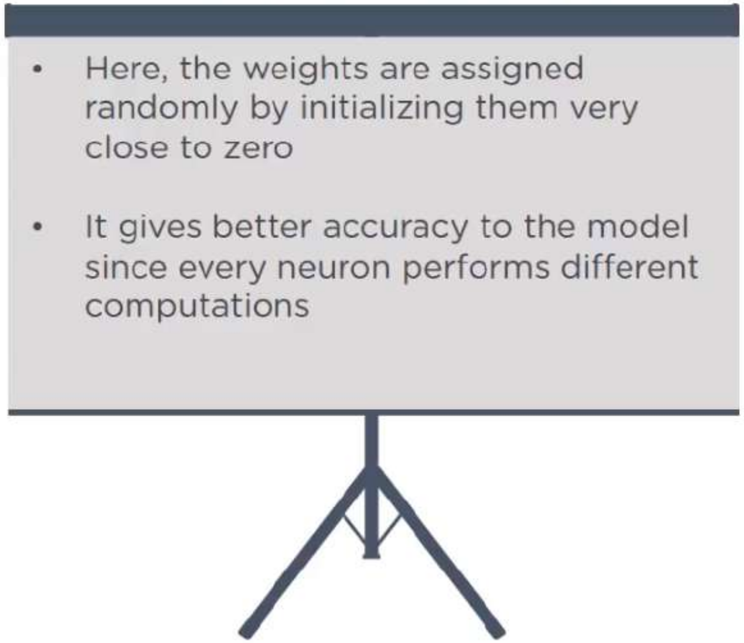
Pooling is a down-sampling operation that reduces the dimensionality of the feature map

4. Fully Connected Layer

Fully connected layer recognises and classifies the objects in the image

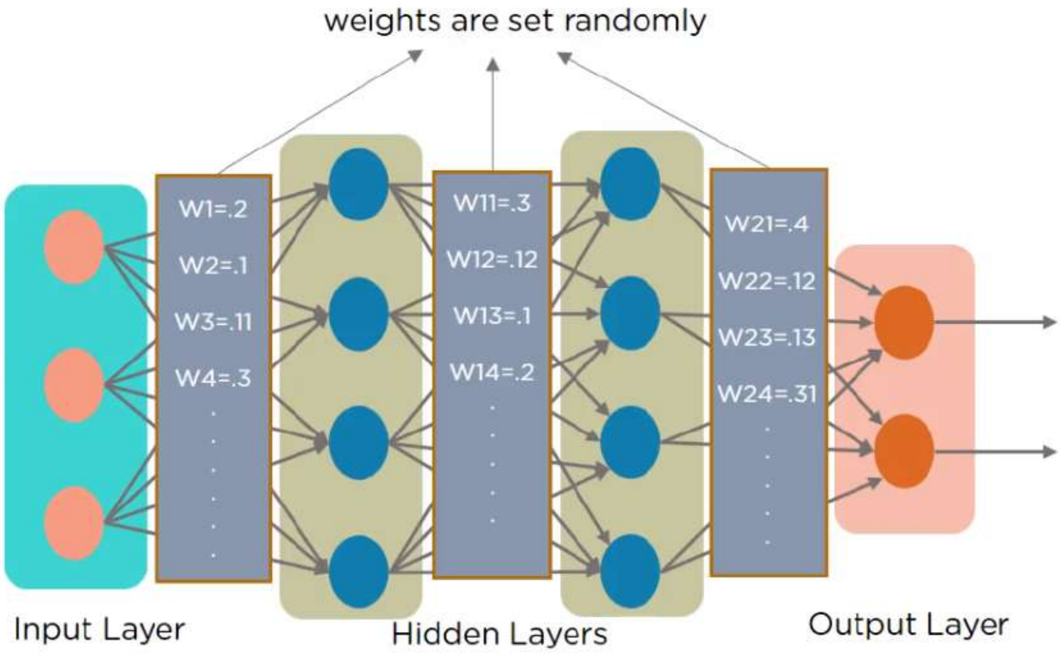
18

How are weights initialized in a network?



Initializing all weights randomly:

- Here, the weights are assigned randomly by initializing them very close to zero
- It gives better accuracy to the model since every neuron performs different computations

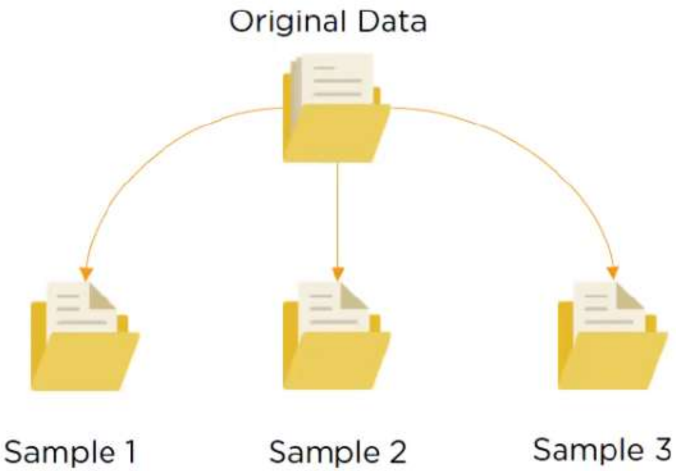


```
w=np.random.randn(layer_size[l],layer_size[l-1])
```

17 Explain Overfitting and Underfitting and how to combat them.

Combating Overfitting and Underfitting

- Resampling the data to estimate the model accuracy (k-fold cross validation)
- Having a validation dataset to evaluate the model



17 Explain Overfitting and Underfitting and how to combat them.

Underfitting

- Underfitting alludes to a model that is neither well trained on training data nor can generalize to new information
- Usually happens when there is less and improper data to train a model
- Has poor performance and accuracy

17 Explain Overfitting and Underfitting and how to combat them.

Overfitting

- Overfitting happens when a model learns the details and noise in the training data to the degree that it adversely impacts the execution of the model on new information

- It is more likely to occur with nonlinear models that have more flexibility when learning a target function

16 What is the difference between Batch Gradient Descent and Stochastic Gradient Descent?

Batch Gradient Descent

- Batch gradient computes the gradient using the entire dataset
- It takes time to converge because the volume of data is huge and weights update slowly

Stochastic Gradient Descent

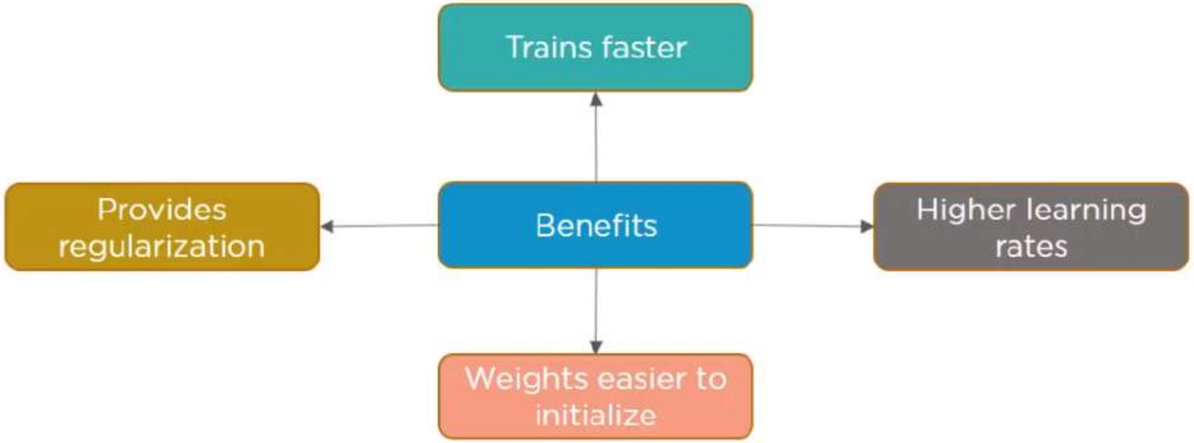
- Stochastic Gradient computes the gradient using a single sample
- It converges much faster than batch gradient because it updates weight more frequently

15

What is Dropout and Batch Normalization?

- **Batch Normalization** is the technique to improve the performance and stability of neural network

- The idea is to normalize the inputs in every layer so that they have mean output activation of zero and standard deviation of one

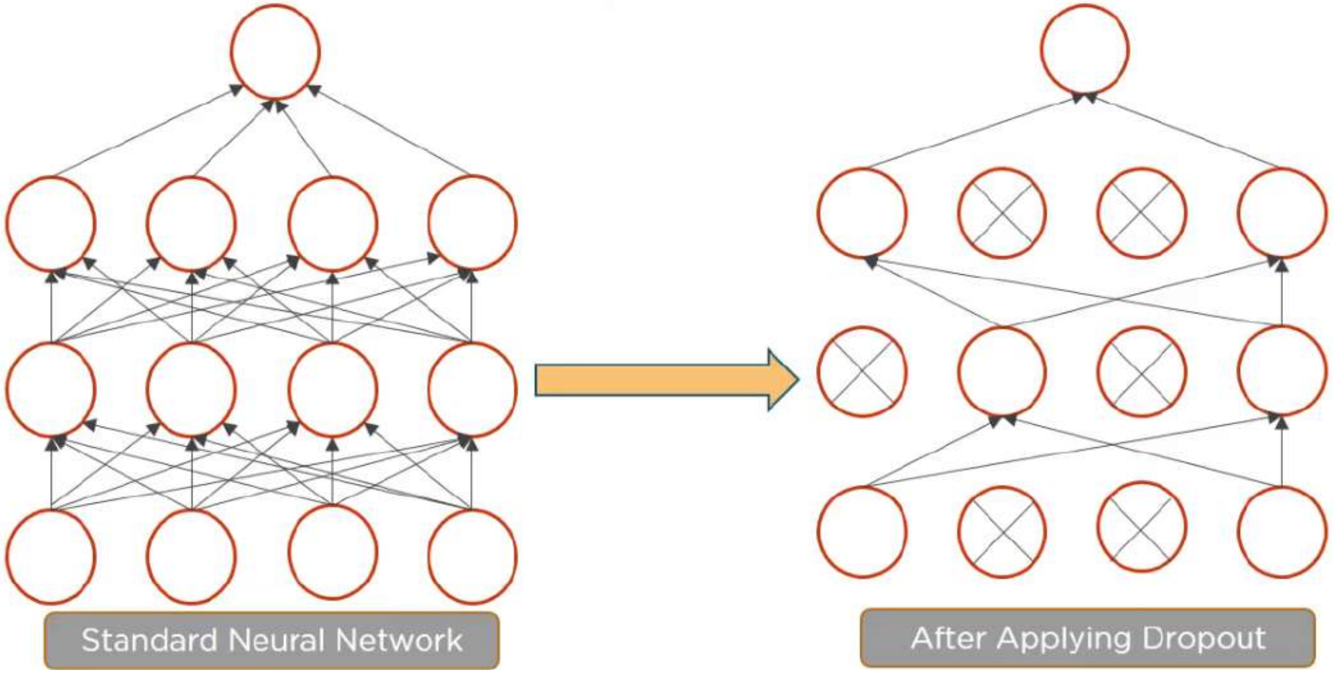


15

What is Dropout and Batch Normalization?

- ❑ **Dropout** is a technique of dropping out hidden and visible units of a network randomly to prevent overfitting of data

- ❑ It doubles the number of iterations needed to converge the network

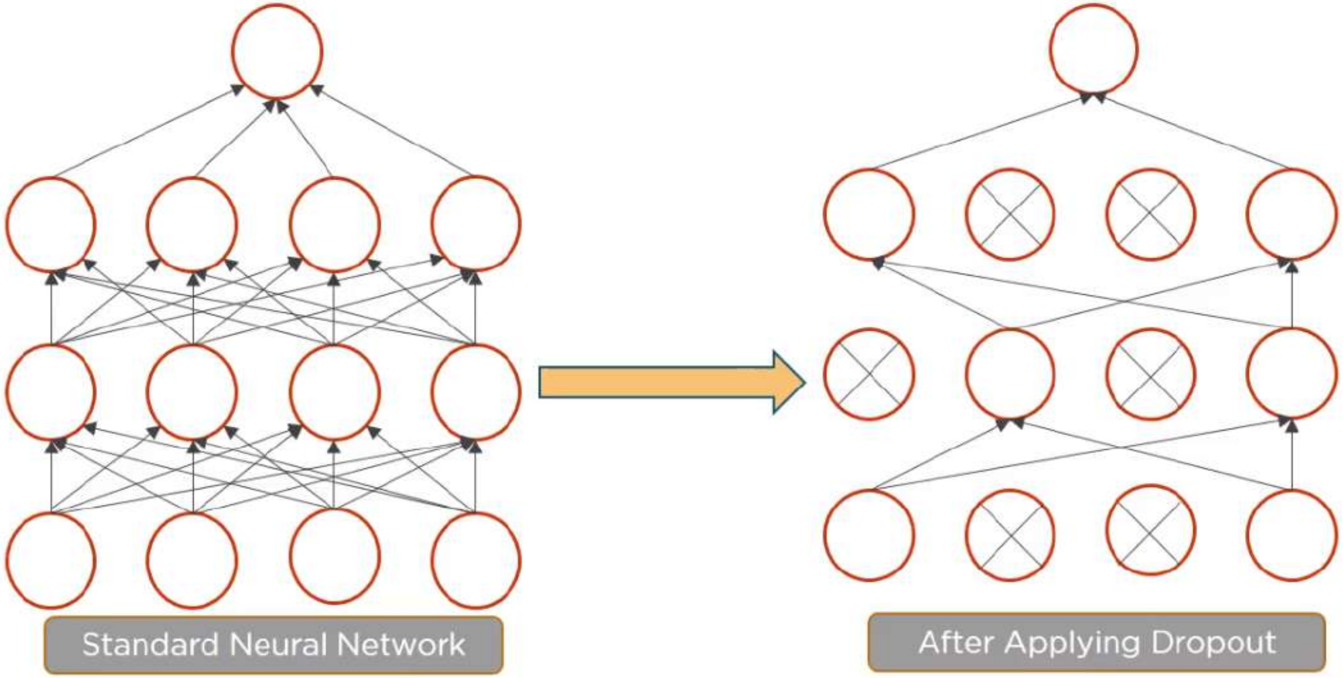


15

What is Dropout and Batch Normalization?

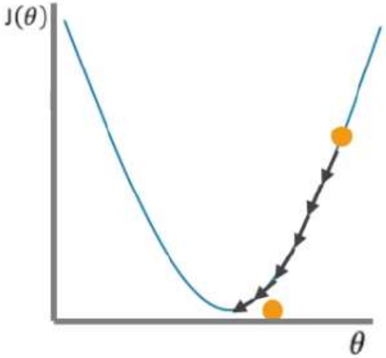
□ **Dropout** is a technique of dropping out hidden and visible units of a network randomly to prevent overfitting of data

□ It doubles the number of iterations needed to converge the network



14

What will happen if learning rate is set too low or too high?



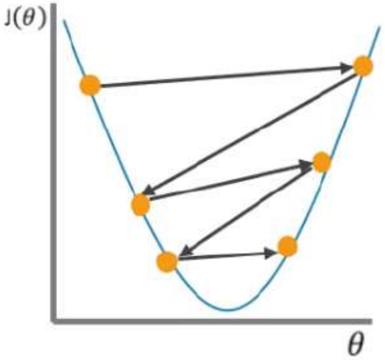
Learning rate too low

Learning rate too low

- Training of the model will progress very slowly as we are making very tiny updates to the weights
- Will take many updates before reaching the minimum point

Learning rate too high

- Causes undesirable divergent behavior to the loss function due to drastic updates in weights
- At times it may fail to converge or even diverge

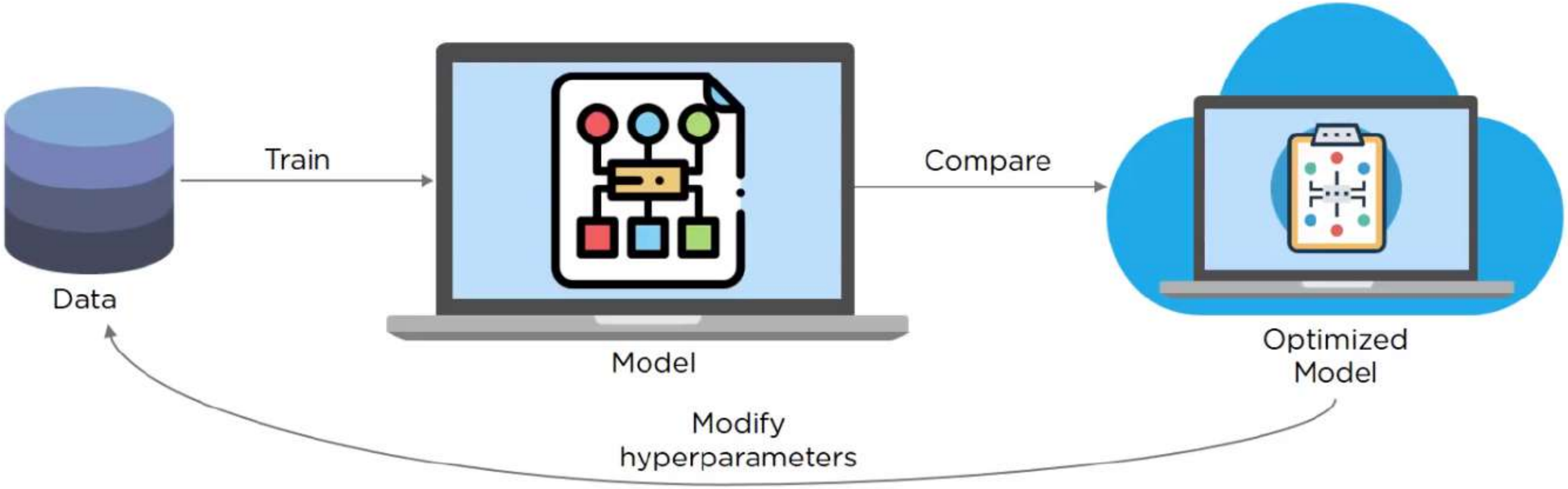


Learning rate too high

13

What are hyperparameters?

- ❑ A hyperparameter is a parameter whose value is set before the learning process begins
- ❑ Determines how a network is trained and the structure of the network
- ❑ **Number of hidden units**, **learning rate**, **epochs**, etc are some examples of hyperparameters

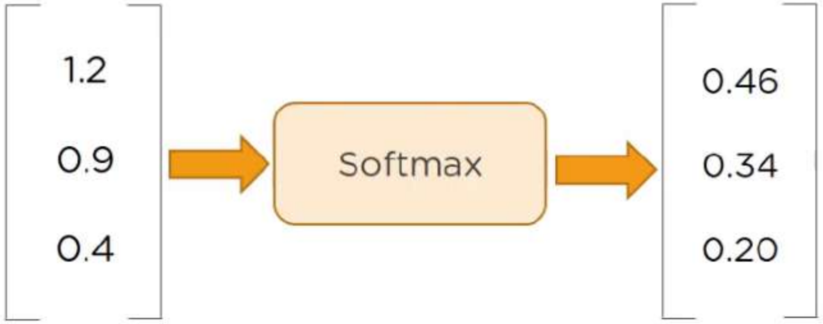


12

What are Softmax and ReLU functions?

- Softmax is an activation function that generates the output between 0 and 1
- It divides each output such that the total sum of the outputs is equal to 1
- It is often used in the output layers

$$\text{Softmax}(L_n) = \frac{e^{L_n}}{\| e^{L} \|}$$



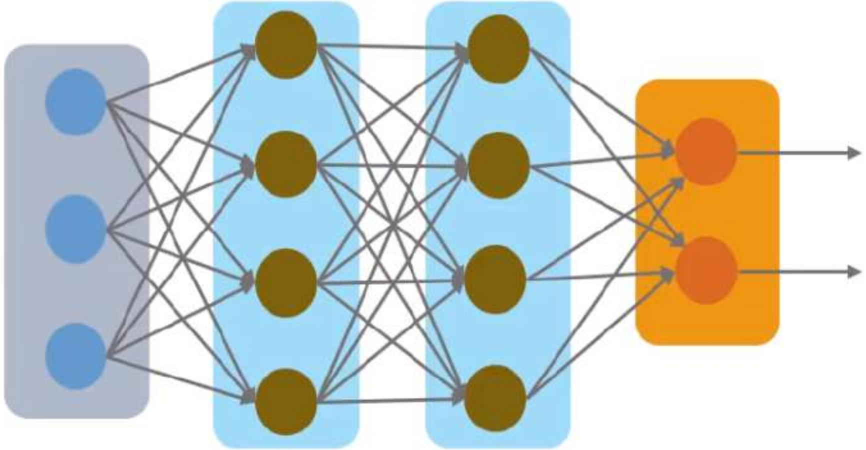
11 What are some applications of Recurrent Neural Network?

Time series problem like predicting the prices of stocks in a month or quarter or sale of products can be solved using RNN

10 What is the difference between Feedforward Neural Network and Recurrent Neural Network?

Feedforward Neural Network

- Signals travel in one direction, from input to output
- No feedback (loops)
- Considers only the current input
- Cannot memorize pervious inputs (eg: CNN)

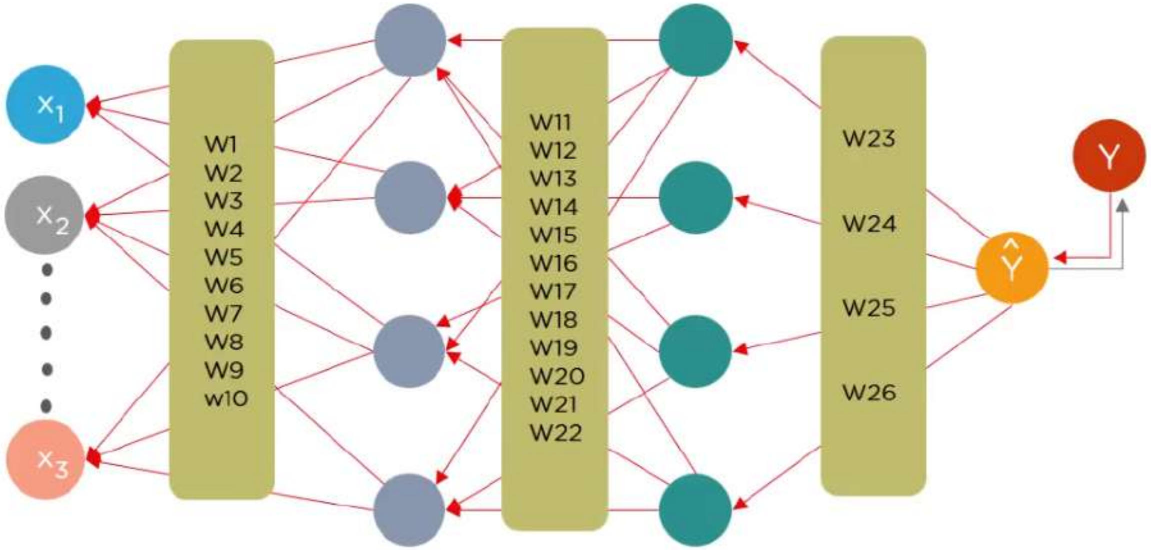


Feed Forward Neural Network

9

What do you understand by Backpropagation?

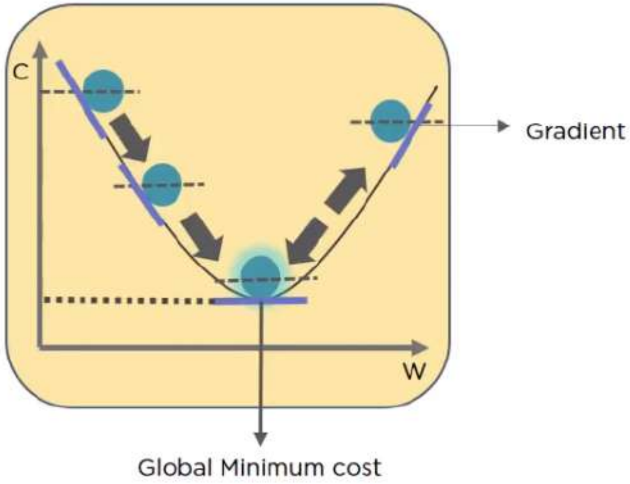
- Neural Network technique to minimize the cost function
- Helps to improve the performance of the network
- Backpropagates the error and updates the weights to reduce the error



8

What is Gradient Descent?

- ❑ Gradient Descent is an optimization algorithm to minimize the cost function in order to maximize the performance of the model
- ❑ Aim is to find the local or global minima of a function
- ❑ Determines the direction the model should take to reduce the error



7

What is a cost function?

□ Cost Function is the measure to evaluate how good your model's performance is

□ It is also referred as **loss** or **error**

□ Used to compute the error of the output layer during backpropagation

Mean Squared Error is an example of a popular cost function:

Cost Function: $C = \frac{1}{2}(Y - \hat{Y})^2$

Y -------> Original Output

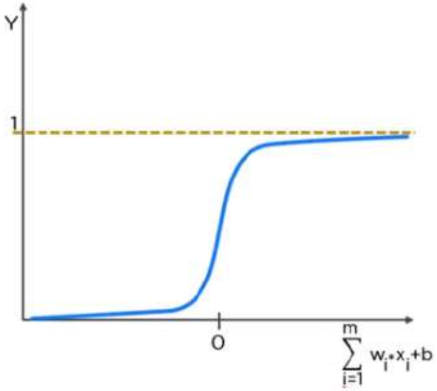
$\hat{Y}$ -------> Predicted Output

6

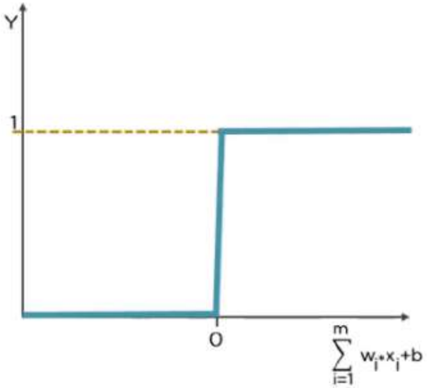
What is the role of Activation Functions in neural network?

- ❑ Activation Function decides whether a neuron should be fired or not
- ❑ Accepts the weighted sum of inputs and a bias as input to any activation function

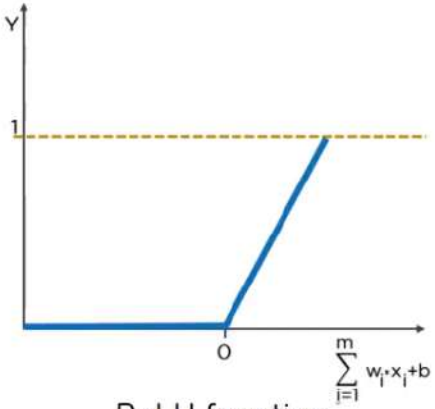
  $Y = \sum(\text{Weight} * \text{input}) + \text{bias}$
- ❑ The node which gets fired depends on the Y value
- ❑ Step function, Sigmoid, ReLU, Tanh, Softmax are some of the examples of Activations Functions



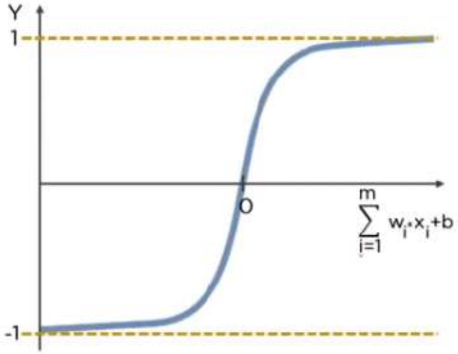
Sigmoid function



Step function



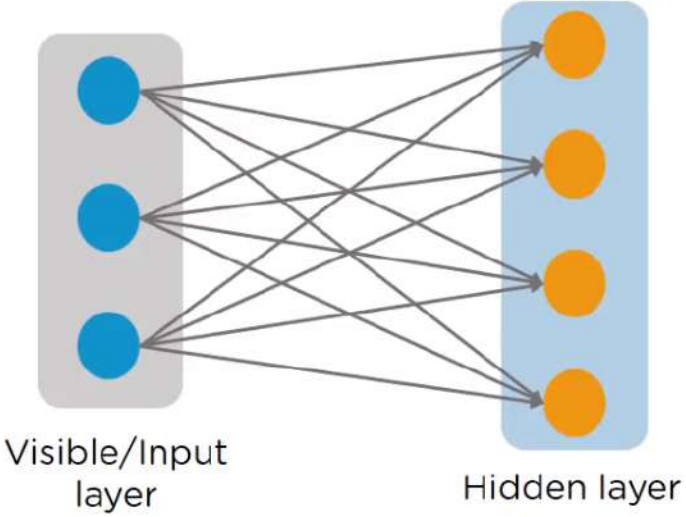
ReLU function



Tanh function

5

What is a Boltzmann Machine?



Shallow, two-layer neural nets that make stochastic decisions whether a neuron should be on or off

1st layer is the visible layer and 2nd layer is the hidden layer

Nodes are connected to each other across layers but no two nodes of the same layer are connected. Hence it is also known as Restricted Boltzmann Machine

4 What is Data Normalization and why do we need it?

Preprocessing step to standardize the data

Used to reduce and eliminate data redundancy

Rescale values to fit into a particular range for achieving better convergence

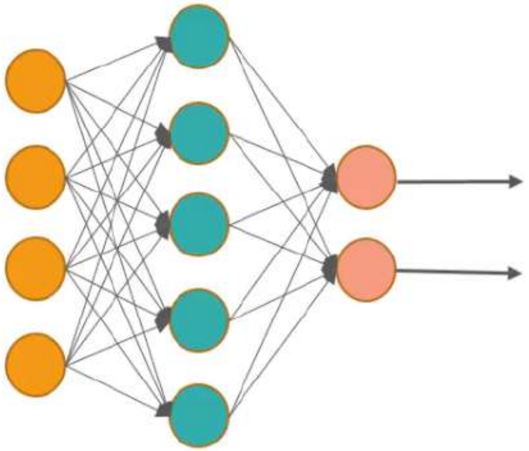
Restructure the data and improve integrity

3

What is a Multilayer Perceptron (MLP)?

Multilayer Perceptron

- It has the same structure of a single layer perceptron with one or more hidden layers.
- Except the input layer, each node in the other layers uses a nonlinear activation function
- MLP uses supervised learning method called ***backpropagation*** for training the model
- Single layer perceptron can classify only linearly separable classes with binary output (0, 1) but MLP can classify non-linear classes



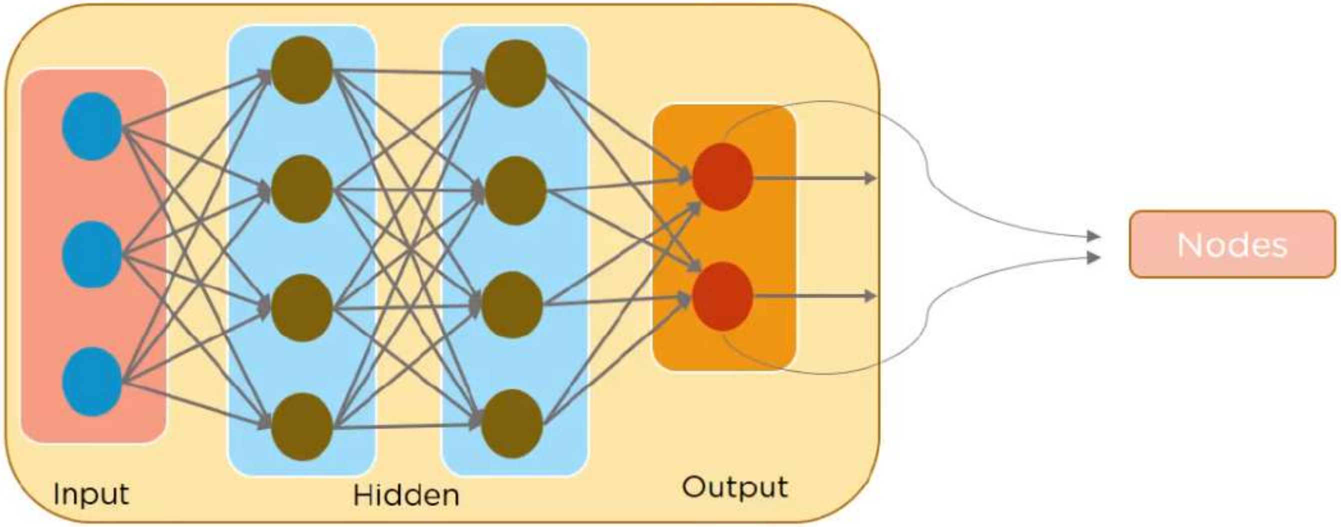
Multilayer Perceptron with an input layer, a hidden layer and an output layer

2

What is a Neural Network?

Human brain inspired systems which replicate the way humans learn

Consists of 3 layers of network (input, hidden and output)

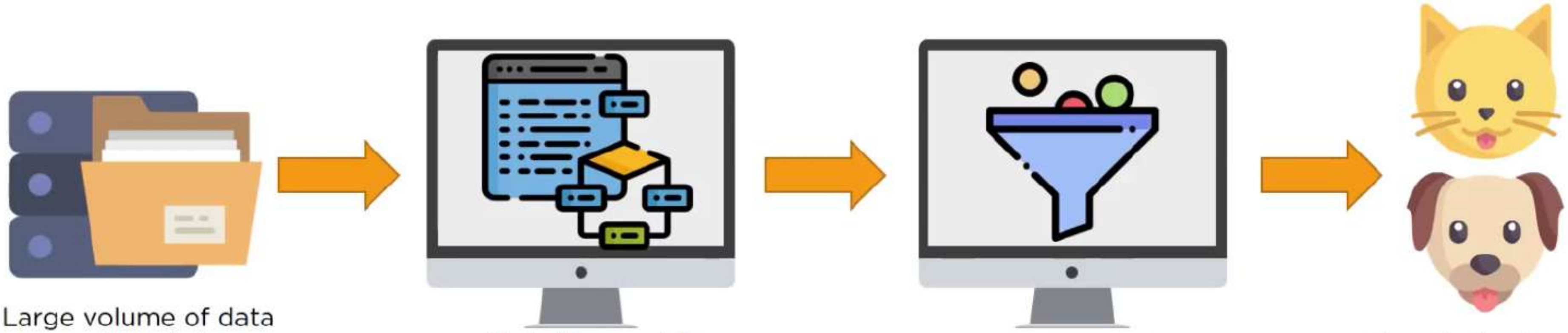


Used in Deep Learning algorithms like CNN, RNN, GAN, etc

Each layer contains neurons called as nodes that perform various operations

1

What is Deep Learning?



Large volume of data on cats and dogs

Train the model

Feature Extraction

Classify the image

Learns from large volumes of structured and unstructured data and uses complex algorithms to train a neural network

Performs complex operations to extract hidden patterns and features