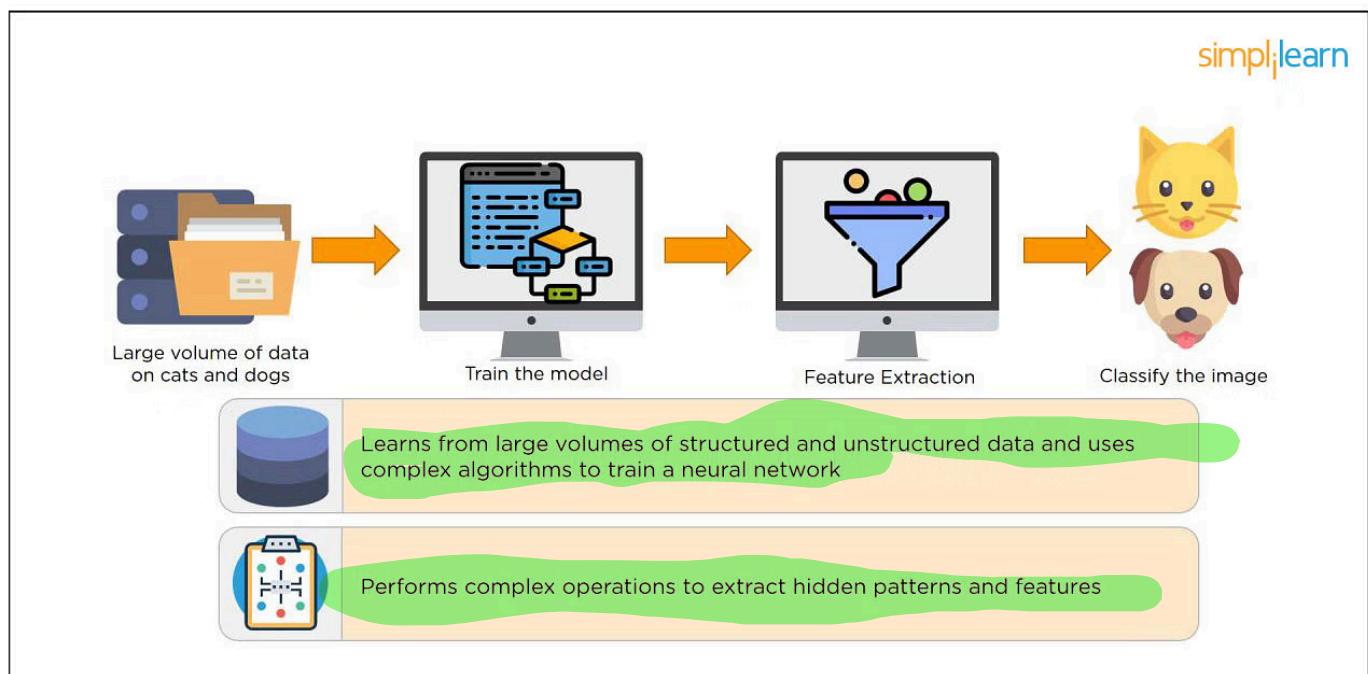


Check out some of the frequently asked deep learning interview questions below:

1. What is Deep Learning?

If you are going for a deep learning interview, you definitely know what exactly deep learning is. However, with this question the interviewee expects you to give an in-detail answer, with an example. **Deep Learning involves taking large volumes of structured or unstructured data and using complex algorithms to train neural networks. It performs complex operations to extract hidden patterns and features (for instance, distinguishing the image of a cat from that of a dog).**



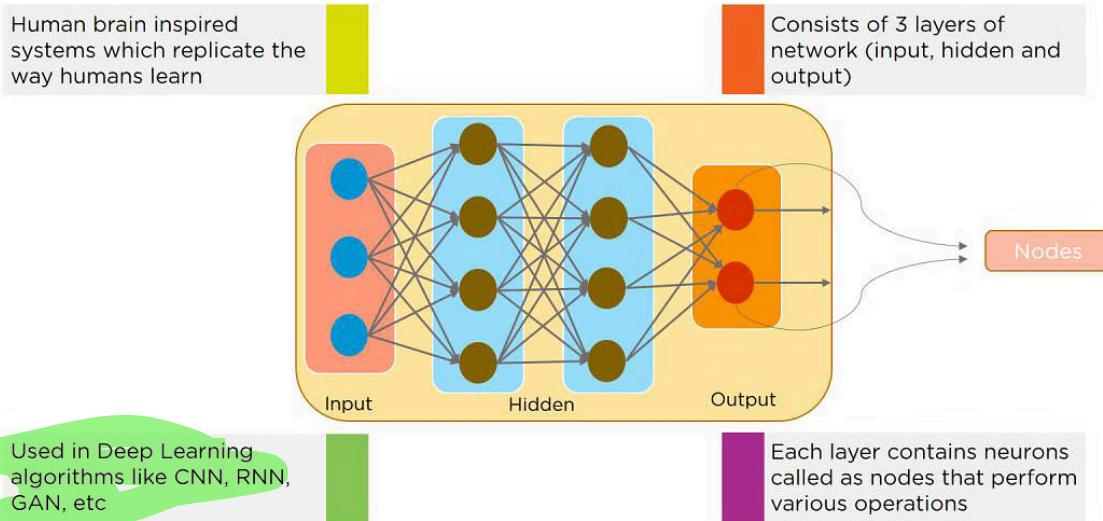
Your AI/ML Career is Just Around The Corner!

AI Engineer Master's Program

[EXPLORE PROGRAM](#)

2. What is a Neural Network?

Neural Networks replicate the way humans learn, inspired by how the neurons in our brains only much simpler.



The most common Neural Networks consist of three network layers:

1. An input layer
2. A hidden layer (this is the most important layer where feature extraction takes place, and adjustments are made to train faster and function better)
3. An output layer

Each sheet contains neurons called “nodes,” performing various operations. Neural Networks are used in [deep learning algorithms](#) like CNN, RNN, GAN, etc.

3. What Is a Multi-layer Perceptron(MLP)?

As in Neural Networks, [MLPs](#) have an input layer, a hidden layer, and an output layer. It has the same structure as a single layer perceptron with one or more hidden layers. A single layer perceptron can classify only linear separable classes with binary output (0,1), but MLP can classify nonlinear classes.

Except for the input layer, each node in the other layers uses a nonlinear activation function. This means the input layers, the data coming in, and the activation function is based upon all nodes and weights being added together, producing the output. MLP uses a [supervised learning](#) method called “backpropagation.” In backpropagation, the neural network calculates the error with the help of cost function. It propagates this error backward from where it came (adjusts the weights to train the model more accurately).

4. What Is Data Normalization, and Why Do We Need It?

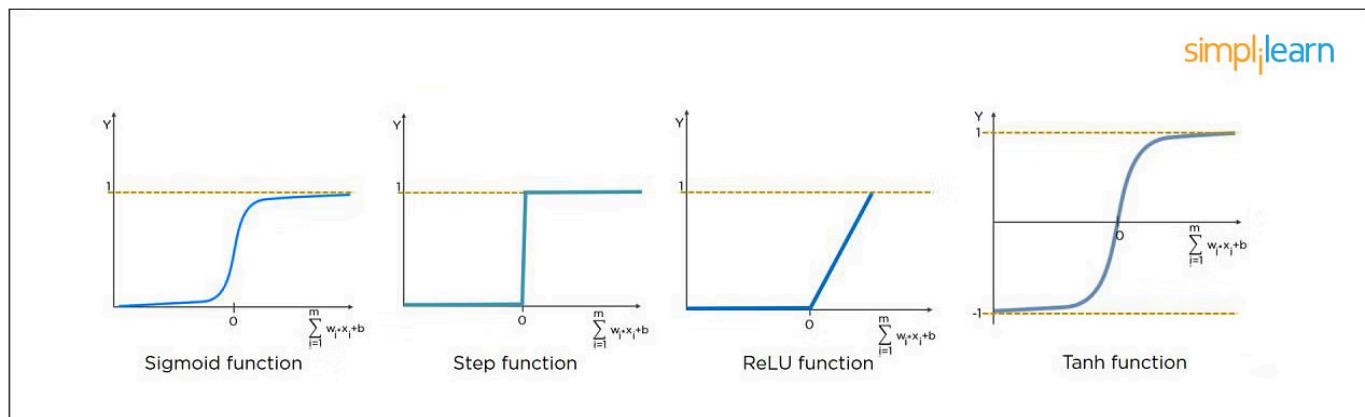
The process of standardizing and reforming data is called “Data Normalization.” It’s a pre-processing step to eliminate data redundancy. Often, data comes in, and you get the same information in different formats. In these cases, you should rescale values to fit into a particular range, achieving better convergence.

5. What is the Boltzmann Machine?

One of the most basic Deep Learning models is a Boltzmann Machine, resembling a simplified version of the Multi-Layer Perceptron. This model features a visible input layer and a hidden layer – just a two-layer neural net that makes stochastic decisions as to whether a neuron should be on or off. Nodes are connected across layers, but no two nodes of the same layer are connected.

6. What Is the Role of Activation Functions in a Neural Network?

At the most basic level, an activation function decides whether a neuron should be fired or not. It accepts the weighted sum of the inputs and bias as input to any activation function. Step function, Sigmoid, ReLU, Tanh, and Softmax are examples of activation functions.



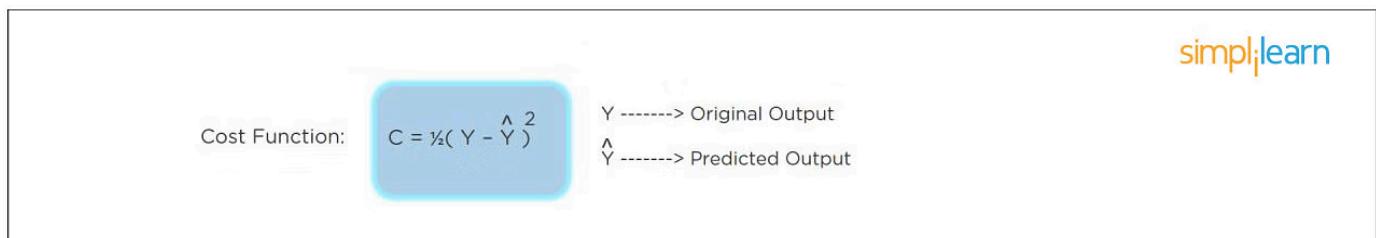
Master Gen AI Strategies for Businesses with

Generative AI for Business Transformation Program

[EXPLORE PROGRAM](#)

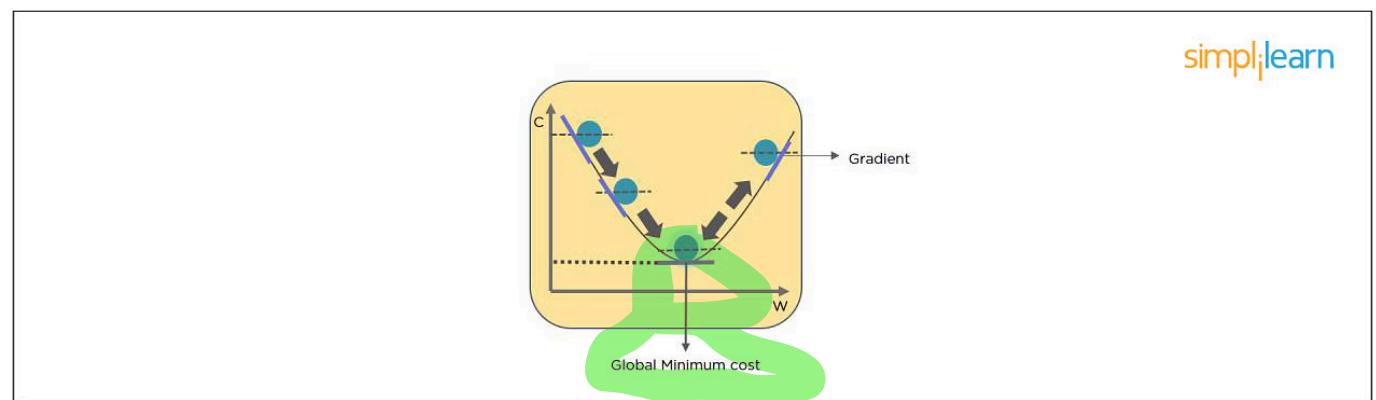
7. What Is the Cost Function?

Also referred to as “loss” or “error,” cost function is a measure to evaluate how good your model’s performance is. It’s used to compute the error of the output layer during backpropagation. We push that error backward through the neural network and use that during the different training functions.



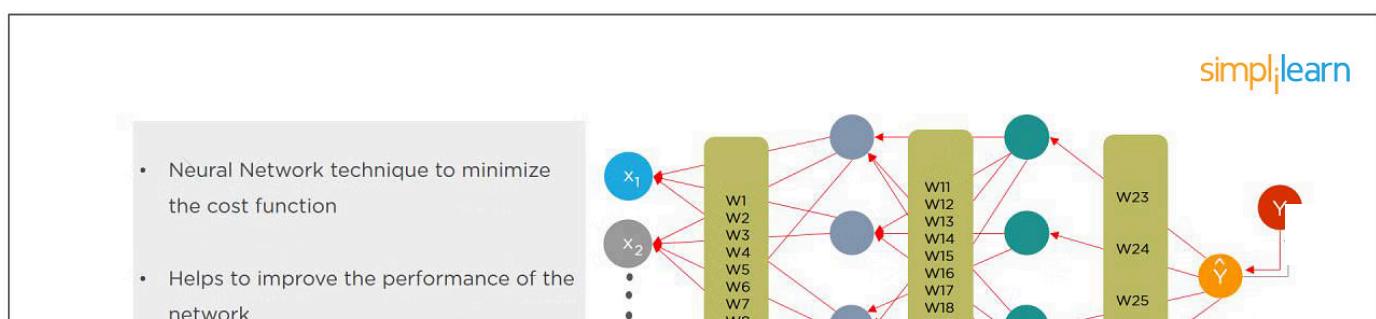
8. What Is Gradient Descent?

Gradient Descent is an optimal algorithm to minimize the cost function or to minimize an error. The aim is to find the local-global minima of a function. This determines the direction the model should take to reduce the error.

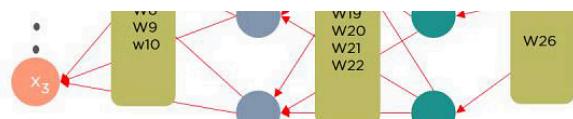


9. What Do You Understand by Backpropagation?

This is one of the most frequently asked deep learning interview questions. Backpropagation is a technique to improve the performance of the network. It backpropagates the error and updates the weights to reduce the error.



- Backpropagates the error and updates the weights to reduce the error



10. What Is the Difference Between a Feedforward Neural Network and Recurrent Neural Network?

In this deep learning interview question, the interviewee expects you to give a detailed answer.

A Feedforward Neural Network signals travel in one direction from input to output. There are no feedback loops; the network considers only the current input. It cannot memorize previous inputs (e.g., CNN).

A Recurrent Neural Network's signals travel in both directions, creating a looped network. It considers the current input along with the previously received inputs for generating the output of a layer and can memorize past data due to its internal memory.

simplilearn

Recurrent Neural Network

- Signals travel in both directions, making it a looped network
- Considers the current input along with the previously received inputs for generating the output of a layer
- Has the ability to memorize past data due to its internal memory

The diagram shows a recurrent neural network cell. It has three main components: an input x (represented by a teal circle), a hidden state h (represented by a brown circle), and an output y (represented by a blue circle). The input x is connected to the hidden state h via a connection labeled B . The hidden state h is connected to the output y via a connection labeled A . Additionally, there is a feedback connection from the hidden state h back to itself, labeled C .

Recurrent Neural Network

11. What Are the Applications of a Recurrent Neural Network (RNN)?

The RNN can be used for sentiment analysis, text mining, and image captioning. Recurrent Neural Networks can also address time series problems such as predicting the prices of stocks in a month or quarter.

Your AI/ML Career is Just Around The Corner!

EXPLORE PROGRAM

12. What Are the Softmax and ReLU Functions?

Softmax is an activation function that generates the **output between zero and one**. It divides each output, such that the **total sum of the outputs is equal to one**. Softmax is often used for output layers.

simplilearn

- 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

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

| | | | | |
|-----------------------|---|---------|---|--------------------------|
| [1.2 0.9 0.4] | → | Softmax | → | [0.46 0.34 0.20] |
|-----------------------|---|---------|---|--------------------------|

ReLU (or Rectified Linear Unit) is the most widely used activation function. **It gives an output of X if X is positive and zeros otherwise. ReLU is often used for hidden layers.**

simplilearn

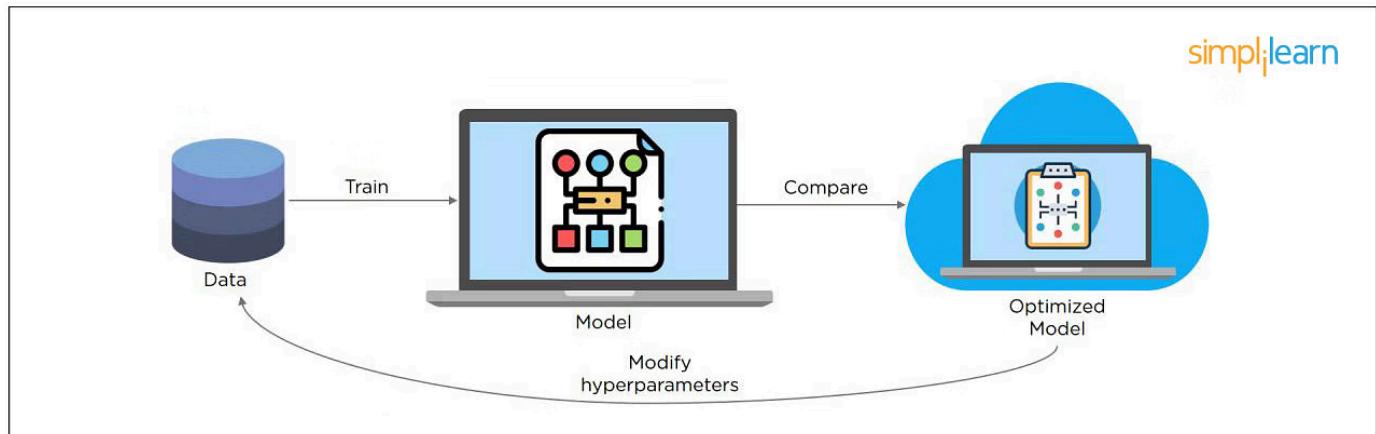
- ReLU stands for Rectified Linear Unit and is the most widely used activation function
- It gives an output of X if X is positive and 0 otherwise
- It is often used in the hidden layers

The graph shows the ReLU function $\phi(x) = \max(x, 0)$. The x-axis is labeled $\sum_{j=1}^n w_j \cdot x_j + b$. The function is zero for all $x < 0$ and increases linearly with slope 1 for all $x > 0$.

13. What Are Hyperparameters?

This is another frequently asked deep learning interview question. With neural networks, you're

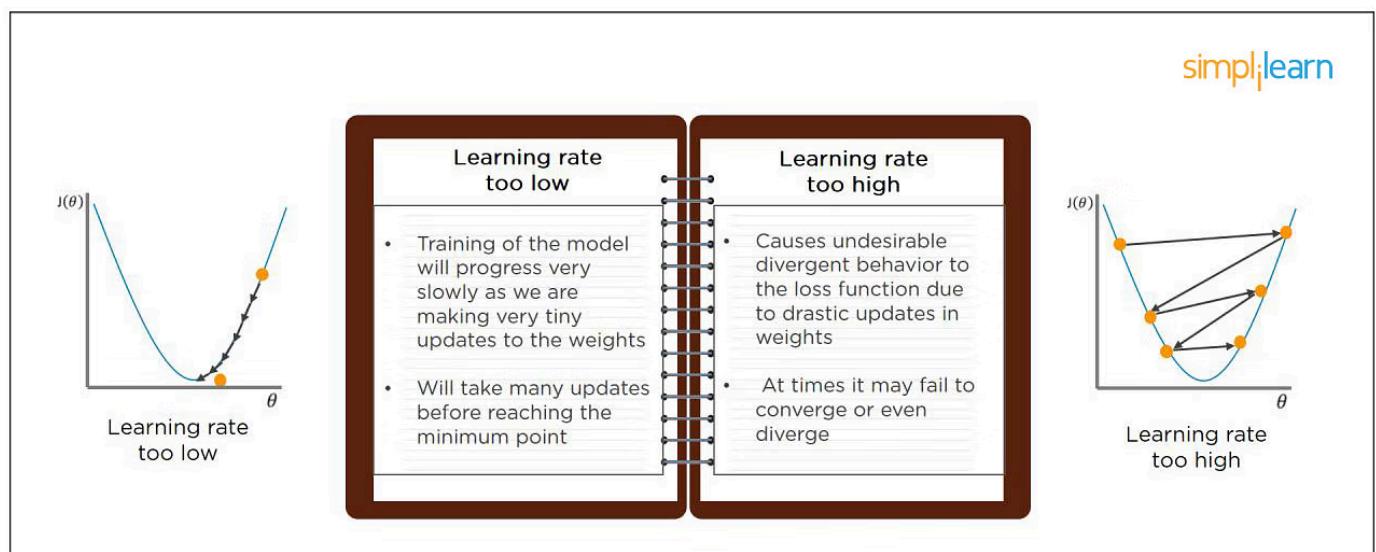
usually working with hyperparameters once the data is formatted correctly. A hyperparameter is a parameter whose value is set before the learning process begins. It determines how a network is trained and the structure of the network (such as the number of hidden units, the learning rate, epochs, etc.).



14. What Will Happen If the Learning Rate Is Set Too Low or Too High?

When your learning rate is too low, training of the model will progress very slowly as we are making minimal updates to the weights. It will take many updates before reaching the minimum point.

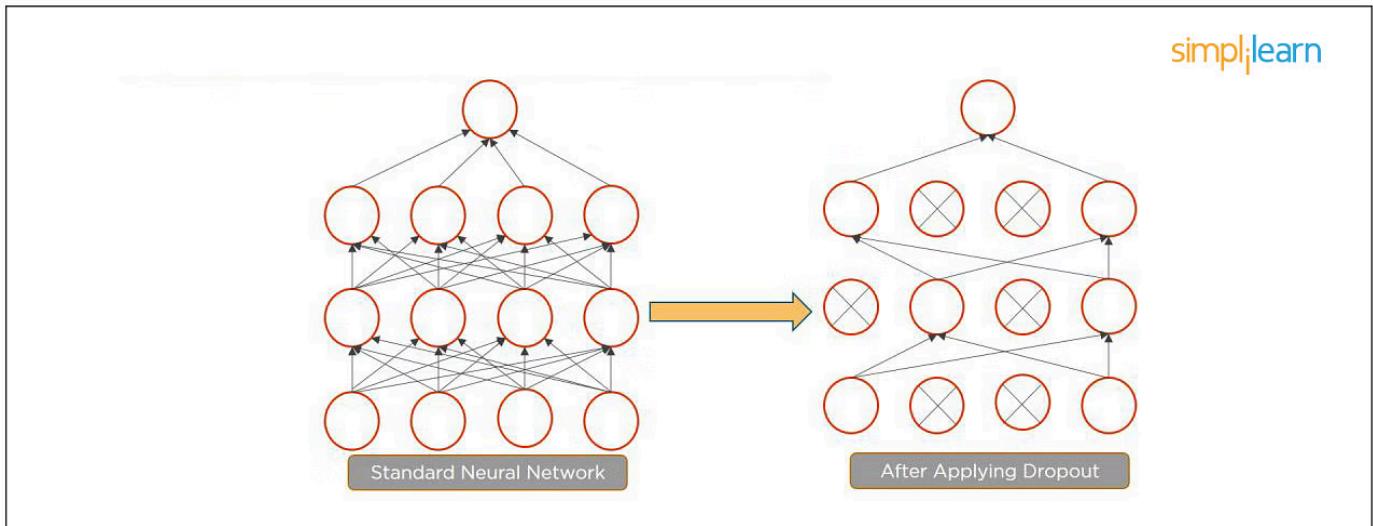
If the learning rate is set too high, this causes undesirable divergent behavior to the loss function due to drastic updates in weights. It may fail to converge (model can give a good output) or even diverge (data is too chaotic for the network to train).



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 (typically dropping 20 percent of the nodes). It doubles the number of

iterations needed to converge the network.



Batch normalization is the technique to improve the performance and stability of neural networks by normalizing the inputs in every layer so that they have mean output activation of zero and standard deviation of one.

The next step on this top Deep Learning interview questions and answers blog will be to discuss intermediate questions.

16. What Is the Difference Between Batch Gradient Descent and Stochastic Gradient Descent?

| Batch Gradient Descent | Stochastic Gradient Descent |
|---|--|
| <p>The batch gradient computes the gradient using the entire dataset.</p> <p>It takes time to converge because the volume of data is huge, and weights update slowly.</p> | <p>The stochastic gradient computes the gradient using a single sample.</p> <p>It converges much faster than the batch gradient because it updates weight more frequently.</p> |

Become a Data Scientist with Hands-on Training!

EXPLORE PROGRAM

17. What is Overfitting and Underfitting, and How to Combat Them?

Overfitting occurs when the 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. An example would be if a model is looking at cars and trucks, but only recognizes trucks that have a specific box shape. It might not be able to notice a flatbed truck because there's only a particular kind of truck it saw in training. The model performs well on training data, but not in the real world.

Underfitting alludes to a model that is neither well-trained on data nor can generalize to new information. This usually happens when there is less and incorrect data to train a model. Underfitting has both poor performance and accuracy.

To combat overfitting and underfitting, you can resample the data to estimate the model accuracy (k-fold cross-validation) and by having a validation dataset to evaluate the model.

18. How Are Weights Initialized in a Network?

There are two methods here: we can either initialize the weights to zero or assign them randomly.

Initializing all weights to 0: This makes your model similar to a linear model. All the neurons and every layer perform the same operation, giving the same output and making the deep net useless.

Initializing all weights randomly: Here, the weights are assigned randomly by initializing them very close to 0. It gives better accuracy to the model since every neuron performs different computations. This is the most commonly used method.

19. What Are the Different Layers on CNN?

There are four layers in CNN:

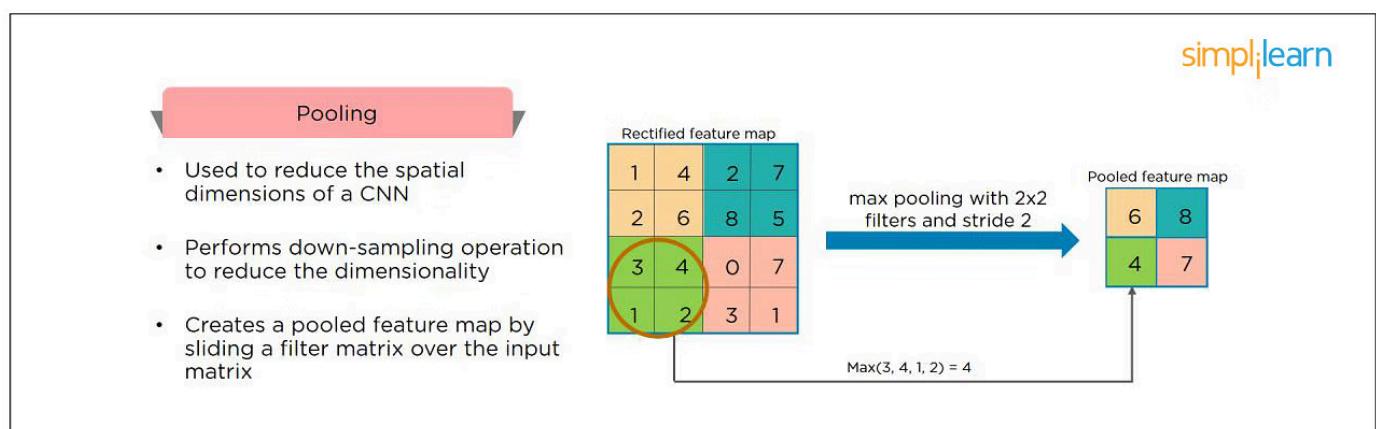
1. Convolutional Layer - the layer that performs a convolutional operation, creating several smaller picture windows to go over the data.

2. ReLU Layer - it brings non-linearity to the network and converts all the negative pixels to zero.
The output is a rectified feature map.
3. Pooling Layer - pooling is a down-sampling operation that reduces the dimensionality of the feature map.
4. Fully Connected Layer - this layer recognizes and classifies the objects in the image.

Master the deep learning concepts and TensorFlow open-source framework with the [Deep Learning Course with TensorFlow Certification](#). Check out the course preview today.

20. What is Pooling on CNN, and How Does It Work?

Pooling is used to reduce the spatial dimensions of a CNN. It performs down-sampling operations to reduce the dimensionality and creates a pooled feature map by sliding a filter matrix over the input matrix.



Your AI/ML Career is Just Around The Corner!

AI Engineer Master's Program

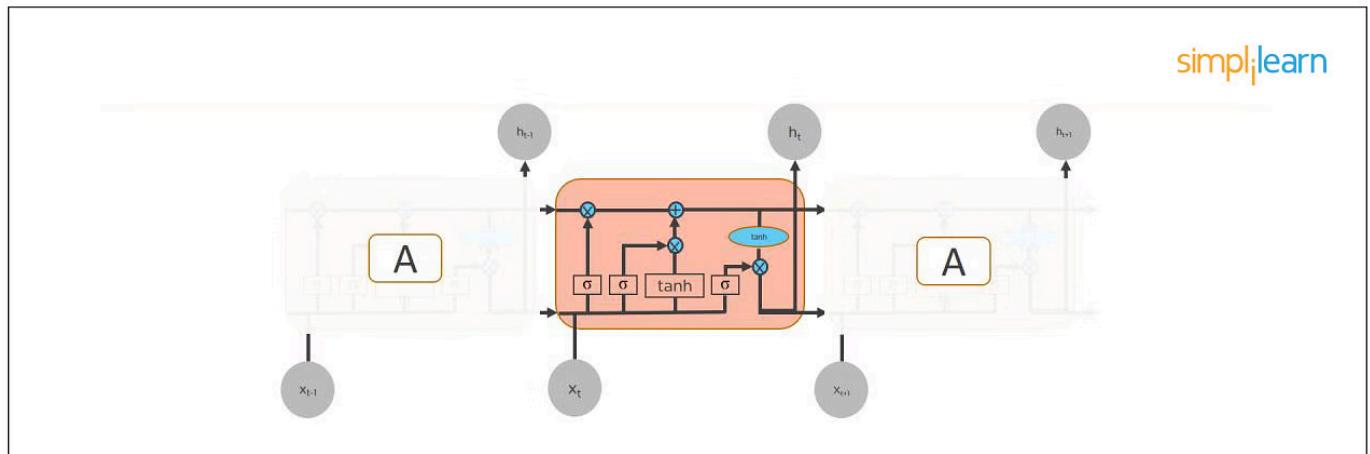
[EXPLORE PROGRAM](#)

21. How Does an LSTM Network Work?

Long-Short-Term Memory (LSTM) is a special kind of recurrent neural network capable of

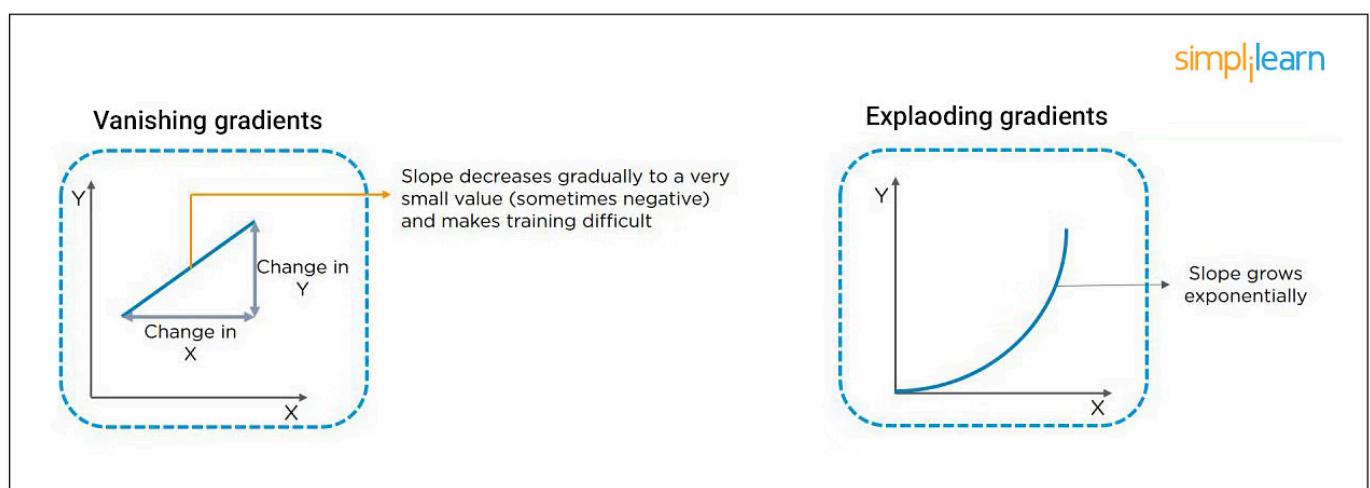
learning long-term dependencies, remembering information for long periods as its default behavior. There are three steps in an LSTM network:

- Step 1: The network decides what to forget and what to remember.
- Step 2: It selectively updates cell state values.
- Step 3: The network decides what part of the current state makes it to the output.



22. What Are Vanishing and Exploding Gradients?

While training an RNN, your slope can become either too small or too large; this makes the training difficult. When the slope is too small, the problem is known as a “Vanishing Gradient.” When the slope tends to grow exponentially instead of decaying, it’s referred to as an “Exploding Gradient.” Gradient problems lead to long training times, poor performance, and low accuracy.



Your AI/ML Career is Just Around The Corner!

EXPLORE PROGRAM

23. What Is the Difference Between Epoch, Batch, and Iteration in Deep Learning?

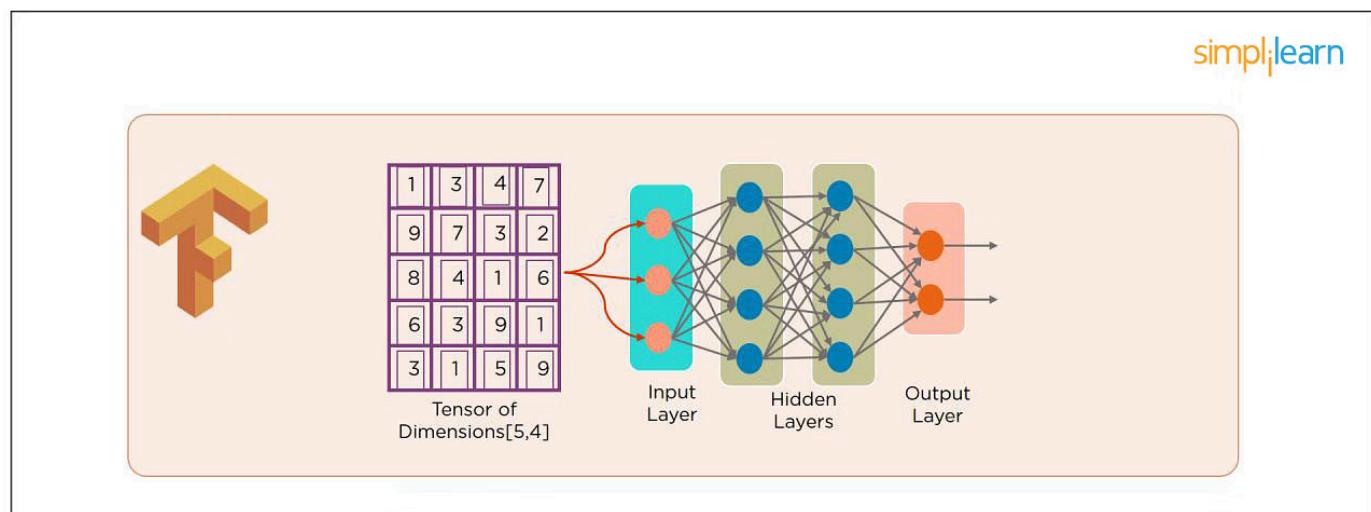
- **Epoch** - Represents one iteration over the entire dataset (everything put into the training model).
- **Batch** - Refers to when we cannot pass the entire dataset into the neural network at once, so we divide the dataset into several batches.
- **Iteration** - if we have 10,000 images as data and a batch size of 200. then an epoch should run 50 iterations (10,000 divided by 50).

24. Why is Tensorflow the Most Preferred Library in Deep Learning?

Tensorflow provides both **C++** and Python APIs, making it easier to work on and has a faster compilation time compared to other Deep Learning libraries like **Keras** and **Torch**. Tensorflow supports both CPU and GPU computing devices.

25. What Do You Mean by Tensor in Tensorflow?

This is another most frequently asked deep learning interview question. A tensor is a mathematical object represented as arrays of higher dimensions. These arrays of data with different dimensions and ranks fed as input to the neural network are called “Tensors.”



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. For 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 the `tf.Variable()` command and initialize them before running the graph in a session. An example:

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

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

Placeholders - these 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 the `tf.placeholder()` command. An 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
```

Sessions - a session is run to evaluate the nodes. This is called the “Tensorflow runtime.” For 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))
```

27. Explain a Computational Graph.

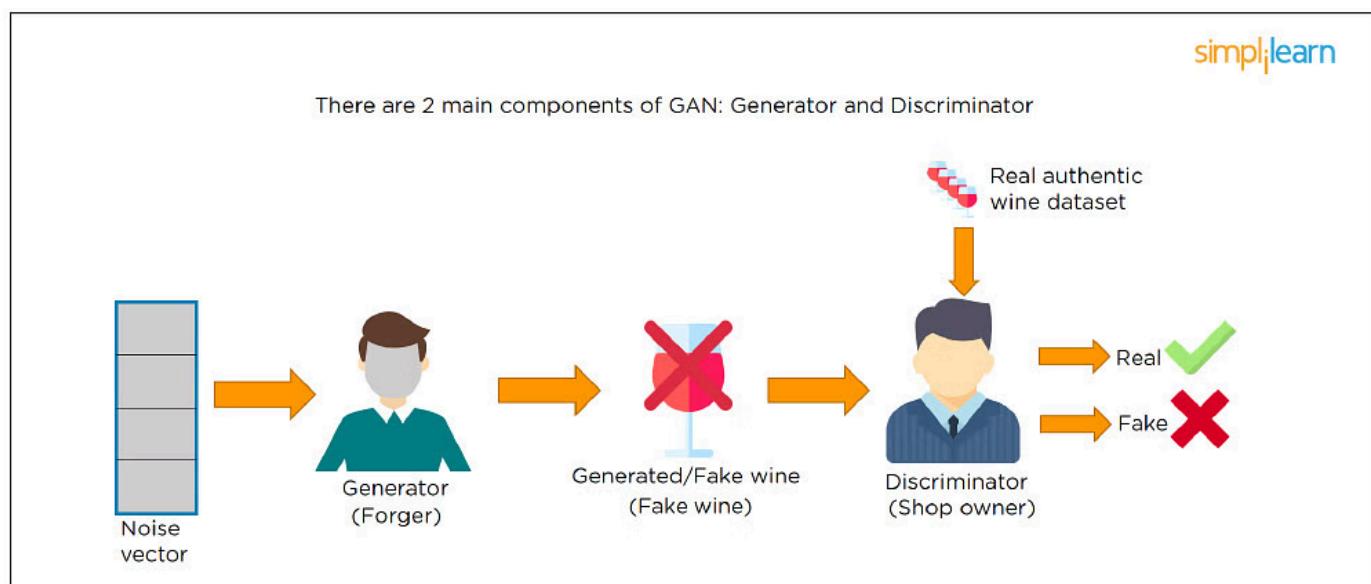
Everything in a [tensorflow](#) is based on creating a computational graph. It has a network of nodes where each node operates, Nodes represent mathematical operations, and edges represent tensors. Since data flows in the form of a graph, it is also called a “DataFlow Graph.”

28. Explain [Generative Adversarial Network](#).

Suppose there is a wine shop purchasing wine from dealers, which they resell later. But some dealers sell fake wine. In this case, the shop owner should be able to distinguish between fake and authentic wine.

The forger will try different techniques to sell fake wine and make sure specific techniques go past the shop owner's check. The shop owner would probably get some feedback from wine experts that some of the wine is not original. The owner would have to improve how he determines whether a wine is fake or authentic.

The forger's goal is to create wines that are indistinguishable from the authentic ones while the shop owner intends to tell if the wine is real or not accurately.



Let us understand this example with the help of an image shown above.

There is a noise vector coming into the forger who is generating fake wine.

Here the forger acts as a Generator.

The shop owner acts as a Discriminator.

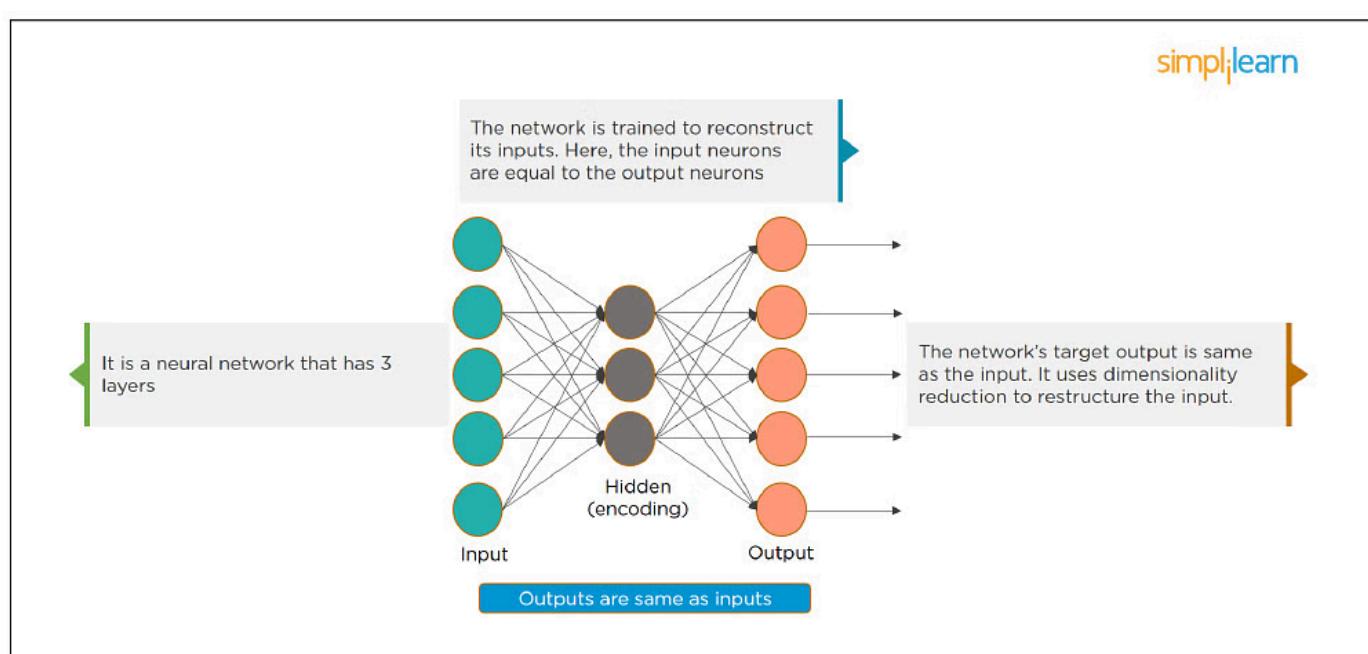
The Discriminator gets two inputs; one is the fake wine, while the other is the real authentic wine.
The shop owner has to figure out whether it is real or fake.

So, there are two primary components of Generative Adversarial Network (GAN) named:

1. Generator
2. Discriminator

The generator is a CNN that keeps keys producing images and is closer in appearance to the real images while the discriminator tries to determine the difference between real and fake images
The ultimate aim is to make the discriminator learn to identify real and fake images.

29. What Is an Auto-encoder?



This Neural Network has three layers in which the input neurons are equal to the output neurons. The network's target outside is the same as the input. It uses dimensionality reduction to restructure the input. It works by compressing the image input to a latent space representation

then reconstructing the output from this representation.

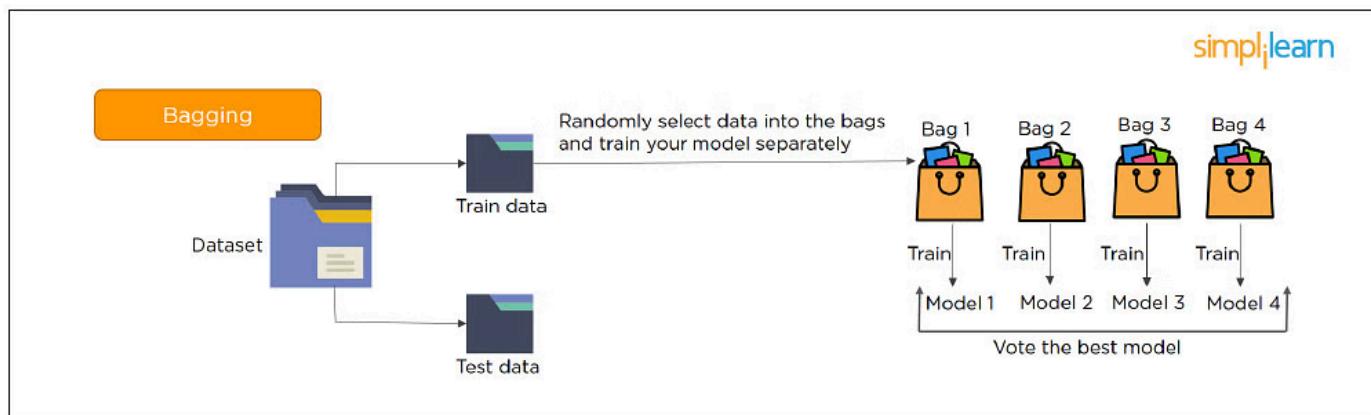
Your AI/ML Career is Just Around The Corner!

AI Engineer Master's Program

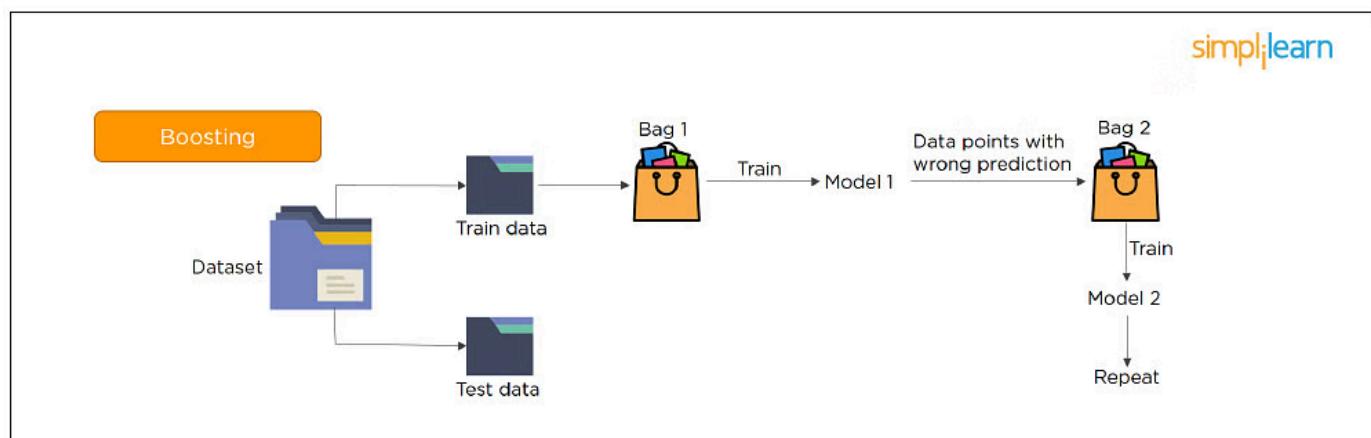
[EXPLORE PROGRAM](#)

30. What Is Bagging and Boosting?

Bagging and Boosting are ensemble techniques to train multiple models using the same learning algorithm and then taking a call.



With Bagging, we take a dataset and split it into training data and test data. Then we randomly select data to place into the bags and train the model separately.



With Boosting, the emphasis is on selecting data points which give wrong output to improve the

accuracy.

The following are some of the most important advanced deep learning interview questions that you should know!

31. What is the significance of using the Fourier transform in Deep Learning tasks?

The Fourier transform function efficiently analyzes, maintains, and manages large datasets. You can use it to generate real-time array data that is helpful for processing multiple signals.

32. What do you understand by transfer learning? Name a few commonly used transfer learning models.

Transfer learning is the process of transferring the learning from a model to another model without having to train it from scratch. It takes critical parts of a pre-trained model and applies them to solve new but similar machine learning problems.

Some of the popular transfer learning models are:

- VGG-16
- BERT
- GTP-3
- Inception V3
- XCEPTION

33. What is the difference between SAME and VALID padding in Tensorflow?

Using the Tensorflow library, `tf.nn.max_pool` performs the max-pooling operation. `Tf.nn.max_pool` has a padding argument that takes 2 values - SAME or VALID.

With padding == "SAME" ensures that the filter is applied to all the elements of the input.

The input image gets fully covered by the filter and specified stride. The padding type is named SAME as the output size is the same as the input size (when stride=1).

With padding == "VALID" implies there is no padding in the input image. The filter window always stays inside the input image. It assumes that all the dimensions are valid so that the input image gets fully covered by a filter and the stride defined by you.

34. What are some of the uses of Autoencoders in Deep Learning?

- Autoencoders are used to convert black and white images into colored images.
- Autoencoder helps to extract features and hidden patterns in the data.
- It is also used to reduce the dimensionality of data.
- It can also be used to remove noises from images.

35. What is the Swish Function?

Swish is an activation function proposed by Google which is an alternative to the ReLU activation function.

It is represented as: $f(x) = x * \text{sigmoid}(x)$.

The Swish function works better than ReLU for a variety of deeper models.

The derivative of Swist can be written as: $y' = y + \text{sigmoid}(x) * (1 - y)$

36. What are the reasons for mini-batch gradient being so useful?

- Mini-batch gradient is highly efficient compared to stochastic gradient descent.
- It lets you attain generalization by finding the flat minima.
- Mini-batch gradient helps avoid local minima to allow gradient approximation for the whole dataset.

37. What do you understand by Leaky ReLU activation function?

Leaky ReLU is an advanced version of the ReLU activation function. In general, the ReLU function defines the gradient to be 0 when all the values of inputs are less than zero. This deactivates the neurons. To overcome this problem, Leaky ReLU activation functions are used. It has a very small slope for negative values instead of a flat slope.

38. What is Data Augmentation in Deep Learning?

Data Augmentation is the process of creating new data by enhancing the size and quality of training datasets to ensure better models can be built using them. There are different techniques

to augment data such as numerical data augmentation, image augmentation, GAN-based augmentation, and text augmentation.

39. Explain the Adam optimization algorithm.

Adaptive Moment Estimation or Adam optimization is an extension to the stochastic gradient descent. This algorithm is useful when working with complex problems involving vast amounts of data or parameters. It needs less memory and is efficient.

Adam optimization algorithm is a combination of two gradient descent methodologies -

Momentum and Root Mean Square Propagation.

40. Why is a convolutional neural network preferred over a dense neural network for an image classification task?

- The number of parameters in a convolutional neural network is much more diminutive than that of a Dense Neural Network. Hence, a CNN is less likely to overfit.
- CNN allows you to look at the weights of a filter and visualize what the network learned. So, this gives a better understanding of the model.
- CNN trains models in a hierarchical way, i.e., it learns the patterns by explaining complex patterns using simpler ones.

41. Which strategy does not prevent a model from over-fitting to the training data?

1. Dropout
2. Pooling
3. Data augmentation
4. Early stopping

Answer: b) Pooling - It's a layer in CNN that performs a downsampling operation.

42. Explain two ways to deal with the vanishing gradient problem in a deep neural network.

- Use the ReLU activation function instead of the sigmoid function
- Initialize neural networks using Xavier initialization that works with tanh activation.

43. Why is a deep neural network better than a shallow neural network?

Both deep and shallow neural networks can approximate the values of a function. But the deep neural network is more efficient as it learns something new in every layer. A shallow neural network has only one hidden layer. But a deep neural network has several hidden layers that create a deeper representation and computation capability.

Your AI/ML Career is Just Around The Corner!

AI Engineer Master's Program

[EXPLORE PROGRAM](#)

44. What is the need to add randomness in the weight initialization process?

If you set the weights to zero, then every neuron at each layer will produce the same result and the same gradient value during backpropagation. So, the neural network won't be able to learn the function as there is no asymmetry between the neurons. Hence, randomness to the weight initialization process is crucial.

45. How can you train hyperparameters in a neural network?

Hyperparameters in a neural network can be trained using four components:

Batch size: Indicates the size of the input data.

Epochs: Denotes the number of times the training data is visible to the neural network to train.

Momentum: Used to get an idea of the next steps that occur with the data being executed.

Learning rate: Represents the time required for the network to update the parameters and learn.

What's Next For You?