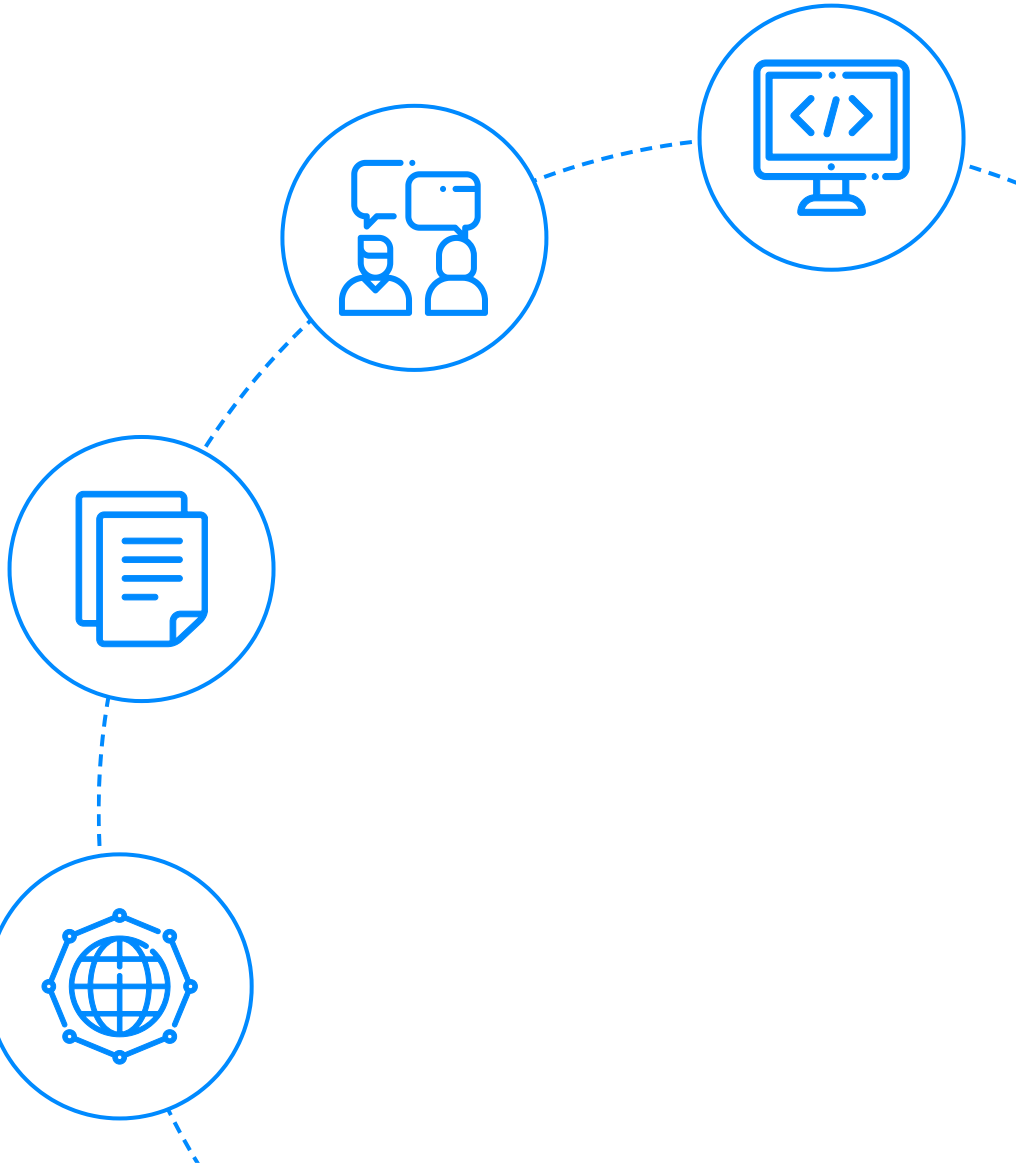
InterviewBit

Deep Learning Interview Questions



To view the live version of the page, click here.

© Copyright by Interviewbit

Contents

Deep Learning Interview Questions for Freshers

InterviewBit

© Copyright by Interviewbit

Deep Learning Interview Questions for Freshers

(.....Continued)

21. Explain Batch Gradient Descent.
22. Explain Stochastic Gradient Descent. How is it different from Batch Gradient Descent?
23. Which deep learning algorithm is the best for face detection?
24. What is an activation function? What is the use of an activation function?
25. What do you mean by an epochs in the context of deep learning?

Deep Learning Interview Questions for Experienced

26. While building a neural network architecture, how will you decide how many neurons and the hidden layers should the neural network have?
27. Can a deep learning model be solely built on linear regression?
28. According to you, which one is more powerful - a two layer neural network without any activation function or a two layer decision tree?
29. Differentiate between bias and variance in the context of deep learning models. How can you achieve balance between the two?
30. How does Recurrent Neural Network backpropagation vary from Artificial Neural Network backpropagation?
31. What exactly do you mean by exploding and vanishing gradients?
32. What are autoencoders? Explain the different layers of autoencoders.
33. Mention the applications of autoencoders.
34. What do you know about Dropout?
35. Differentiate between Deep Learning and Machine Learning.
36. Explain the different types of activation functions.

© Copyright by Interviewbit

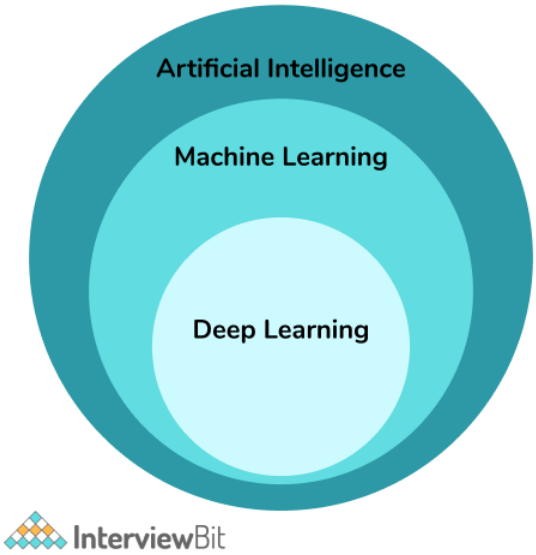
Let's get Started

What is Deep Learning?

© Copyright by Interviewbit

Deep learning is a subset of machine learning that is entirely based on artificial neural networks. Because neural networks are designed to mimic the human brain, deep learning is likewise a human brain mimic. We don't have to explicitly program everything in deep learning. We train a model on a training dataset and improvise it until the model predicts almost correctly on the testing and validation dataset as well. Deep learning models are capable of focusing on accurate features on their own, with only a little input from the programmer, and are highly useful in resolving the dimensionality problem. Deep learning is not a new concept. It has been around for quite some time. It's all the rage these days since we don't have nearly as much processing power or as much data as we do now. As processing power has increased tremendously over the last 20 years, deep learning and machine learning have entered the scene.

While working on a deep learning network in the mid-1960s, Alexey Grigorevich Ivakhnenko published the first general. Essentially, it is a machine learning subset that does feature extraction and transformation using a large number of nonlinear processing units. Each of the subsequent layers uses the output from the previous layer as input. Deep learning is appropriate for a variety of applications, including computer vision, speech recognition, natural language processing, and so on.



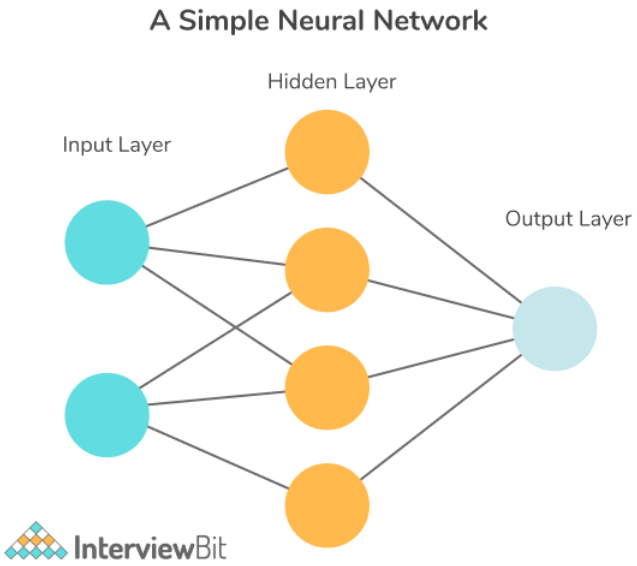
© Copyright by Interviewbit

The above image depicts that Machine Learning is a subset of Artificial Intelligence and Deep Learning is a subset of Machine Learning.

Deep Learning Interview Questions for Freshers

1. What do you understand about Neural Networks in the context of Deep Learning?

Neural Networks are artificial systems that have a lot of resemblance to the biological neural networks in the human body. A neural network is a set of algorithms that attempts to recognize underlying relationships in a batch of data using a method that mimics how the human brain works. Without any task-specific rules, these systems learn to do tasks by being exposed to a variety of datasets and examples. The notion is that instead of being programmed with a pre-coded understanding of these datasets, the system derives identifying traits from the data it is fed to. Neural networks are built on threshold logic computational models. Because neural networks can adapt to changing input, they can produce the best possible outcome without requiring the output criteria to be redesigned.



2. What are the applications of deep learning?

© Copyright by Interviewbit

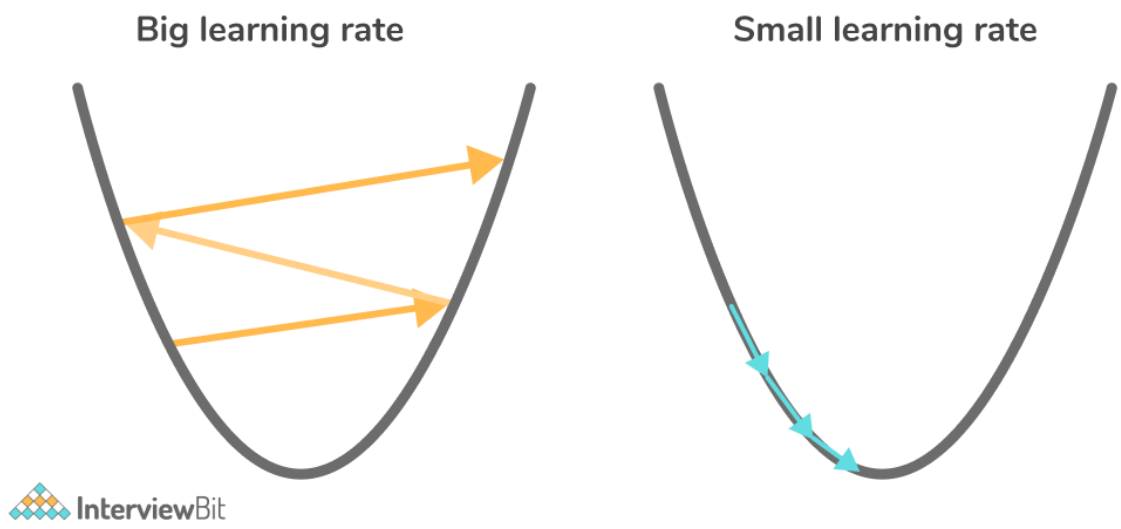
Following are some of the applications of deep learning:-

- Pattern recognition and natural language processing.
- Recognition and processing of images.
- Automated translation.
- Analysis of sentiment.
- System for answering questions.
- Classification and Detection of Objects.
- Handwriting Generation by Machine.
- Automated text generation.
- Colorization of Black and White images.

3. Explain learning rate in the context of neural network models. What happens if the learning rate is too high or too low?

Learning rate is a number that ranges from 0 to 1. It is one of the most important tunable hyperparameters in neural network training models. The learning rate determines how quickly or slowly a neural network model adapts to a given situation and learns. A higher learning rate value indicates that the model only needs a few training epochs and produces rapid changes, whereas a lower learning rate indicates that the model may take a long time to converge or may never converge and become stuck on a poor solution. As a result, it is recommended that a good learning rate value be established by trial and error rather than using a learning rate that is too low or too high.

© Copyright by Interviewbit



In the above image, we can clearly see that a big learning rate leads us to move away from the desired output. However, having a small learning rate leads us to the desired output eventually.

4. What are the advantages of neural networks?

Following are the advantages of neural networks:

- Neural networks are extremely adaptable, and they may be used for both classification and regression problems, as well as much more complex problems. Neural networks are also quite scalable. We can create as many layers as we wish, each with its own set of neurons. When there are a lot of data points, neural networks have been shown to generate the best outcomes. They are best used with non-linear data such as images, text, and so on. They can be applied to any data that can be transformed into a numerical value.
- Once the neural network mode has been trained, they deliver output very fast. Thus, they are time-effective.

5. What are the disadvantages of neural networks?

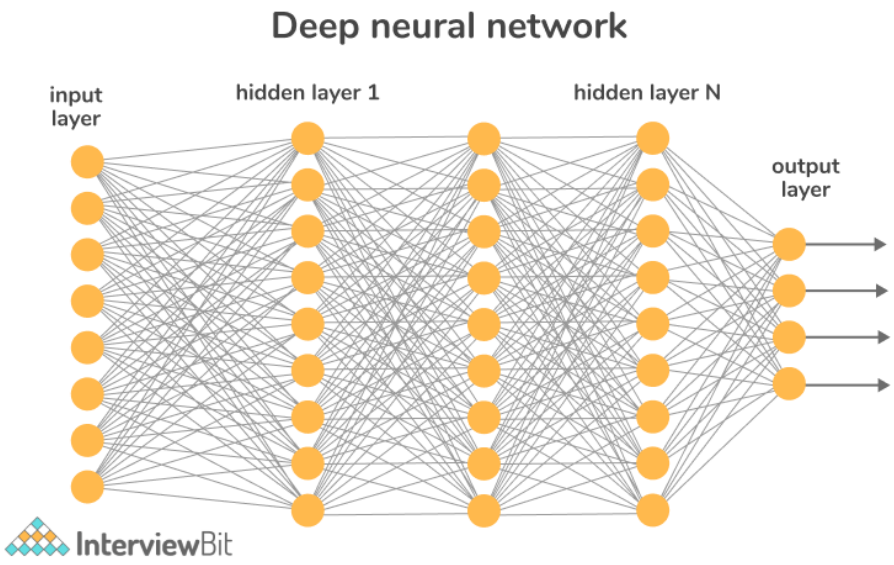
Following are the disadvantages of neural networks:-

© Copyright by Interviewbit

- The "black box" aspect of neural networks is a well-known disadvantage. That is, we have no idea how or why our neural network produced a certain result. When we enter a dog image into a neural network and it predicts that it is a duck, we may find it challenging to understand what prompted it to make this prediction.
- It takes a long time to create a neural network model.
- Neural networks models are computationally expensive to build because a lot of computations need to be done at each layer.
- A neural network model requires significantly more data than a traditional machine learning model to train.

6. Explain what a deep neural network is.

An artificial neural network (ANN) having numerous layers between the input and output layers is known as a deep neural network (DNN). Deep neural networks are neural networks that use deep architectures. The term "deep" refers to functions that have a higher number of layers and units in a single layer. It is possible to create more accurate models by adding more and larger layers to capture higher levels of patterns. The below image depicts a deep neural network.

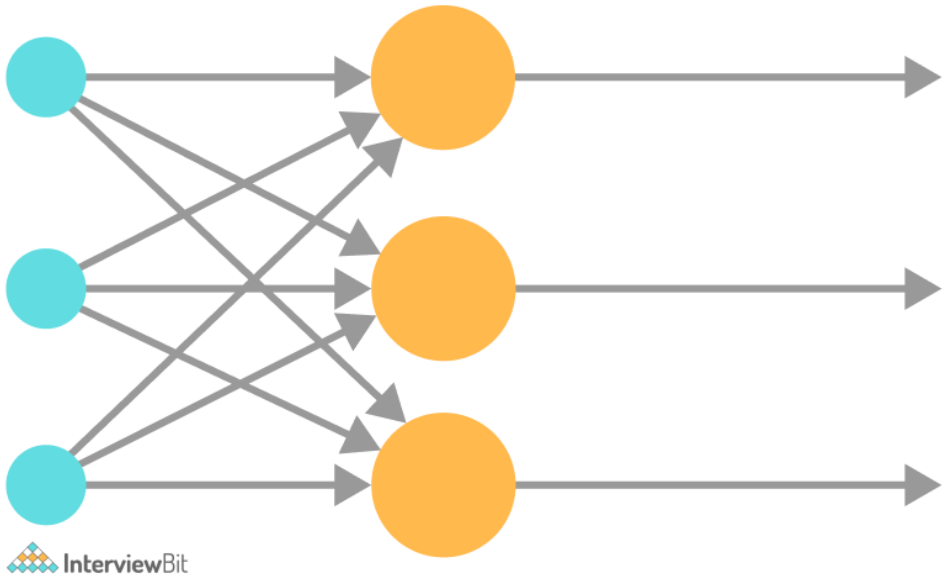


7. What are the different types of deep neural networks?

© Copyright by Interviewbit

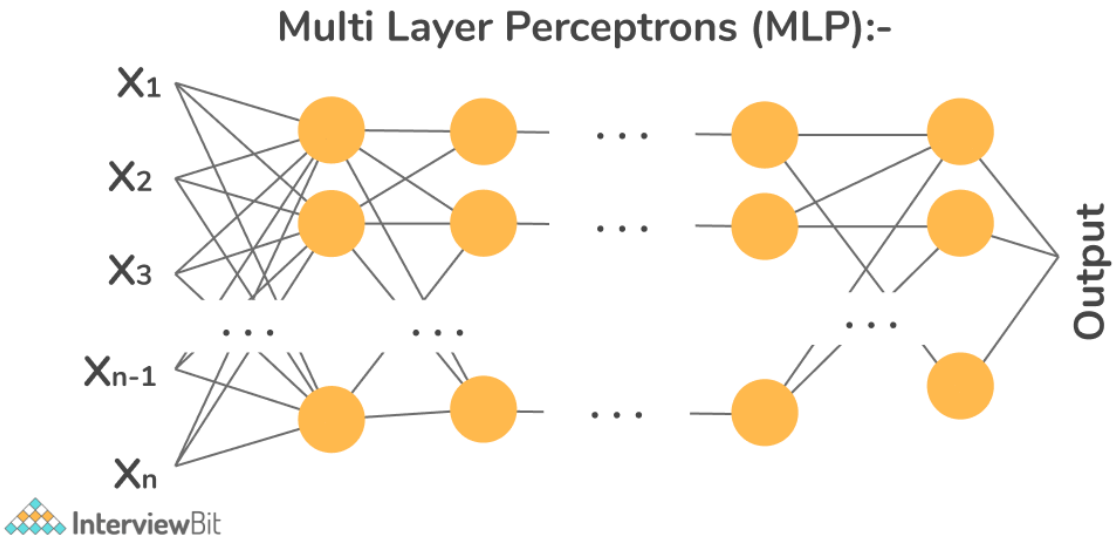
Following are the different types of deep neural networks:-

- **FeedForward Neural Network:-** This is the most basic type of neural network, in which flow control starts at the input layer and moves to the output layer. These networks only have a single layer or a single hidden layer. There is no backpropagation mechanism in this network because data only flows in one way. The input layer of this network receives the sum of the weights present in the input. These networks are utilised in the computer vision-based facial recognition method.



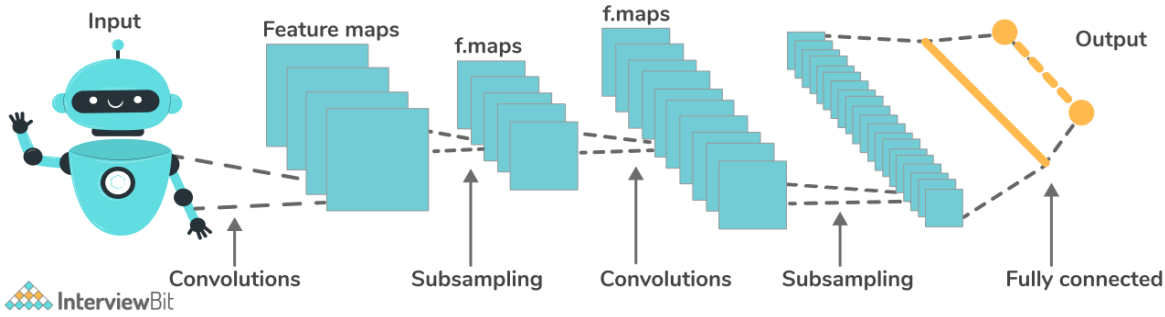
© Copyright by Interviewbit

- **Radial Basis Function Neural Network:-** This type of neural network usually has more than one layer, preferably two. The relative distance from any location to the center is determined in this type of network and passed on to the next layer. In order to avoid blackouts, radial basis networks are commonly employed in power restoration systems to restore power in the shortest period possible.
- **Multi-Layer Perceptrons (MLP):-** A multilayer perceptron (MLP) is a type of feedforward artificial neural network (ANN). MLPs are the simplest deep neural networks, consisting of a succession of completely linked layers. Each successive layer is made up of a collection of nonlinear functions that are the weighted sum of all the previous layer's outputs (completely linked). Speech recognition and other machine learning systems rely heavily on these networks.



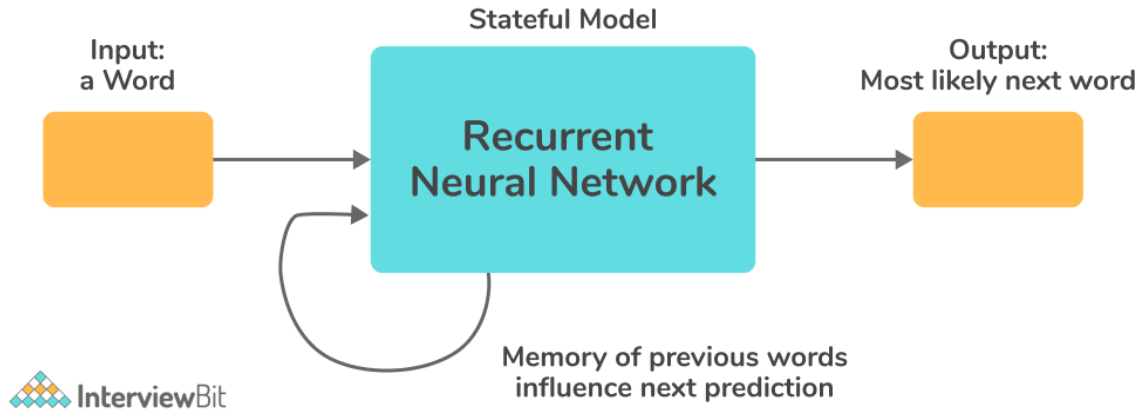
© Copyright by Interviewbit

- **Convolutional Neural Network (CNN):-** Convolutional Neural Networks are mostly used in computer vision. In contrast to fully linked layers in MLPs, one or more convolution layers extract simple characteristics from input by performing convolution operations in CNN models. Each layer is made up of nonlinear functions of weighted sums at various coordinates of spatially close subsets of the previous layer's outputs, allowing the weights to be reused.
  The AI system learns to automatically extract the properties of these inputs to fulfill a specific task, such as picture classification, face identification, and image semantic segmentation, given a sequence of images or videos from the actual world.



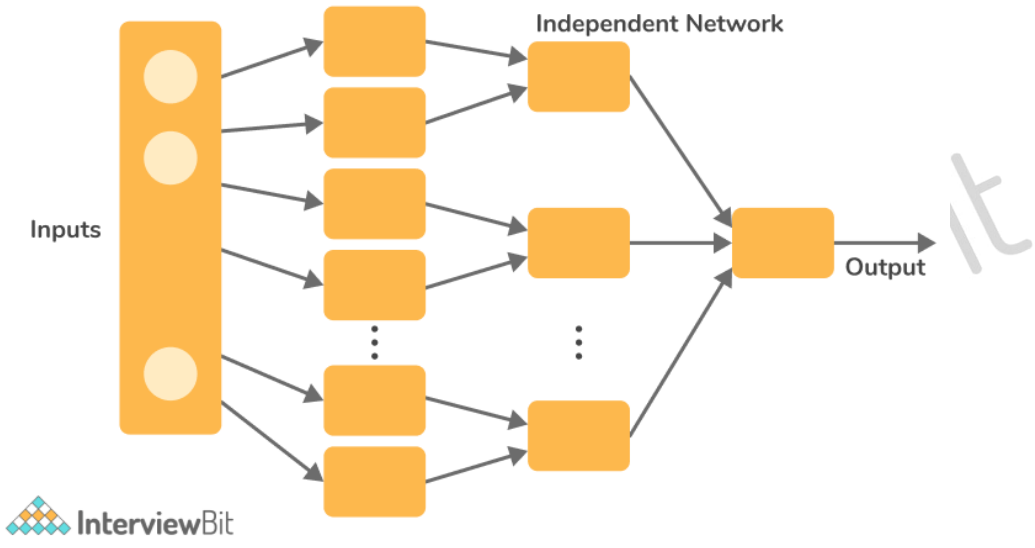
© Copyright by Interviewbit

- **Recurrent Neural Network (RNN):-** Recurrent Neural Networks were created to solve the sequential input data time-series problem. RNN's input is made up of the current input and prior samples. As a result, the node connections create a directed graph. Furthermore, each neuron in an RNN has an internal memory that stores the information from previous samples' computations. Because of their superiority in processing data with a variable input length, RNN models are commonly employed in natural language processing (NLP). The goal of AI in this case is to create a system that can understand human-spoken natural languages, such as natural language modeling, word embedding, and machine translation.
  Each successive layer in an RNN is made up of nonlinear functions of weighted sums of outputs and the preceding state. As a result, the basic unit of RNN is termed "cell," and each cell is made up of layers and a succession of cells that allow recurrent neural network models to be processed sequentially.



- **Modular Neural Network:-** This network is made up of numerous tiny neural networks, rather than being a single network. The sub-networks combine to form a larger neural network, which operates independently to achieve a common goal. These networks are extremely useful for breaking down a large-small problem into smaller chunks and then solving it.

© Copyright by Interviewbit



- **Sequence to Sequence Model:-** In most cases, this network is made up of two RNN networks. The network is based on encoding and decoding, which means it has an encoder that processes the input and a decoder that processes the output. This type of network is commonly employed for text processing when the length of the inputting text differs from the length of the outputted text.

8. What do you mean by end-to-end learning?

It's a deep learning procedure in which a model is fed raw data and the entire data is trained at the same time to create the desired result with no intermediate steps. It is a deep learning method in which all of the different steps are trained simultaneously rather than sequentially. End-to-end learning has the advantage of eliminating the requirement for implicit feature engineering, which usually results in lower bias. Driverless automobiles are an excellent example that you may use in your end-to-end learning content. They are guided by human input and are programmed to learn and interpret information automatically using a CNN to fulfill tasks. Another good example is the generation of a written transcript (output) from a recorded audio clip (input). The model here skips all of the steps in the middle, focusing instead on the fact that it can manage the entire sequence of steps and tasks.

© Copyright by Interviewbit

9. What do you understand about gradient clipping in the context of deep learning?

Gradient Clipping is a technique for dealing with the problem of exploding gradients (a situation in which huge error gradients build up over time, resulting in massive modifications to neural network model weights during training) that happens during backpropagation. The problem of exploding gradients occurs when the gradients get excessively big during training, causing the model to become unstable. If the gradient has crossed the anticipated range, the gradient values are driven element-by-element to a specific minimum or maximum value. Gradient clipping improves numerical stability while training a neural network, but it has little effect on the performance of the model.

10. Explain Forward and Back Propagation in the context of deep learning.

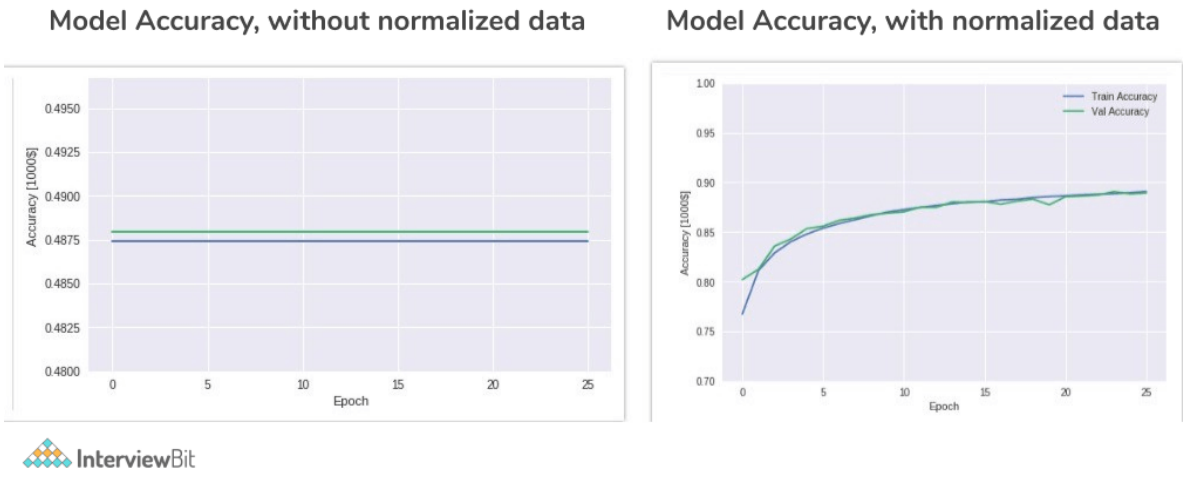
- **Forward Propagation:** The hidden layer, between the input layer and the output layer of the network, receives inputs with weights. We calculate the output of the activation at each node at each hidden layer, and this propagates to the next layer until we reach the final output layer. We go forward from the inputs to the final output layer, which is known as the forward propagation.
- **Back Propagation:** It sends error information from the network's last layer to all of the weights within the network. It's a technique for fine-tuning the weights of a neural network based on the previous epoch's (i.e., iteration) error rate. By fine-tuning the weights, you may lower error rates and improve the model's generalization, making it more dependable. The process of backpropagation can be broken down into the following steps: It can generate output by propagating training data through the network. It, then, computes the error derivative for output activations using the target and output values. It can backpropagate to compute the derivative of the error in the previous layer's output activation, and so on for all hidden layers. It calculates the error derivative for weights using the previously obtained derivatives and all hidden layers. The weights are updated based on the error derivatives obtained from the next layer.

© Copyright by Interviewbit

11. Explain Data Normalisation. What is the need for it?

Data Normalisation is a technique in which data is transformed in such a way that they are either dimensionless or have a similar distribution. It is also known as standardization and feature scaling. It's a pre-processing procedure for the input data that removes redundant data from the dataset.

Normalization provides each variable equal weights/importance, ensuring that no single variable biases model performance in its favour simply because it is larger. It vastly improves model precision by converting the values of numeric columns in a dataset to a similar scale without distorting the range of values.



12. What are the different techniques to achieve data normalization?

Following are the different techniques employed to achieve data normalization:-

© Copyright by Interviewbit

- **Rescaling:** Rescaling data is the process of multiplying each member of a data set by a constant term k, or changing each integer x to f(X), where f(x) = kx and k and x are both real values. The simplest of all approaches, rescaling (also known as "min-max normalization"), is calculated as:

  {"detectHand":false}

  This represents the rescaling factor for every data point x.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

InterviewBit

- **Mean Normalisation:** In the transformation process, this approach employs the mean of the observations:

  {"detectHand":false}

  This represents the mean normalizing factor for every data point x.

© Copyright by Interviewbit

$$x' = \left(\frac{x - \text{average}(x)}{\max(x) - \min(x)} \right)$$

InterviewBit

- **Z-score Normalisation:** This technique, also known as standardization, employs the Z-score or "standard score." SVM and logistic regression are two examples of machine learning algorithms that utilise it:

  {"detectHand":false}

  This represents the Z-score.

$$z = \left(\frac{x - \mu}{\sigma} \right)$$

InterviewBit

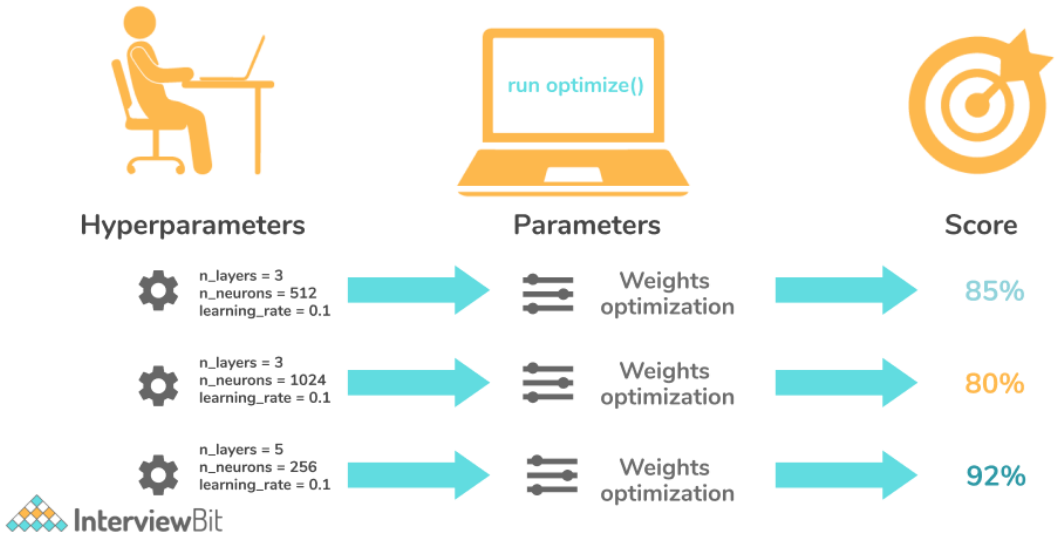
13. What do you mean by hyperparameters in the context of deep learning?

© Copyright by Interviewbit

Hyperparameters are variables that determine the network topology (for example, the number of hidden units) and how the network is trained (Eg: Learning Rate). They are set before training the model, that is, before optimizing the weights and the bias.

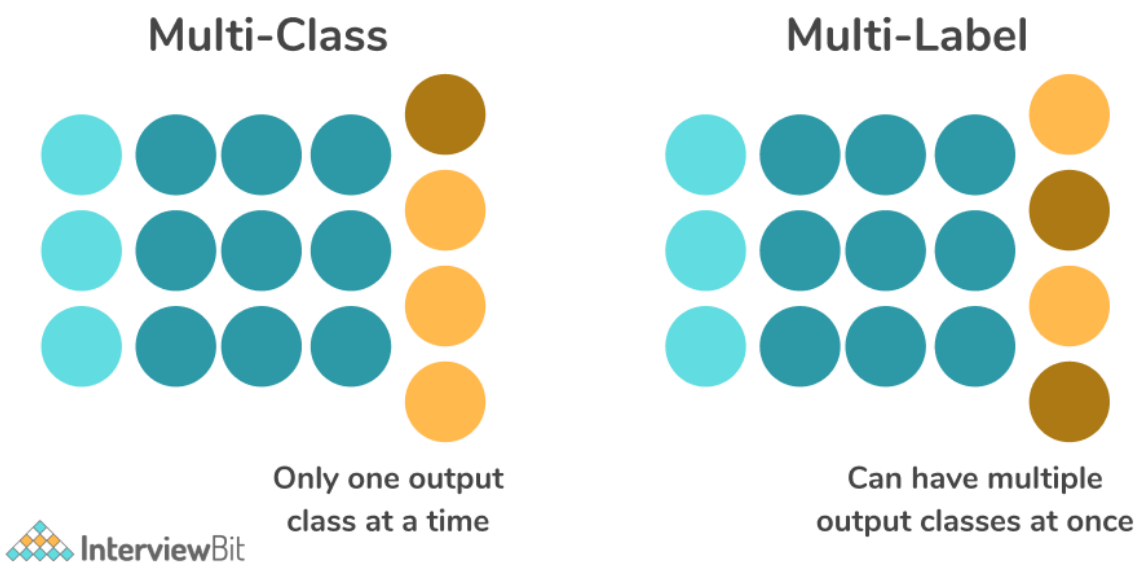
Following are some of the examples of hyperparameters:-

- **Number of hidden layers:** With regularisation techniques, many hidden units inside a layer can boost accuracy. Underfitting may occur if the number of units is reduced.
- **Learning Rate:** The learning rate is the rate at which a network's parameters are updated. The learning process is slowed by a low learning rate, but it eventually converges. A faster learning rate accelerates the learning process, but it may not converge. A declining Learning rate is usually desired.



14. Difference between multi-class and multi-label classification problems.

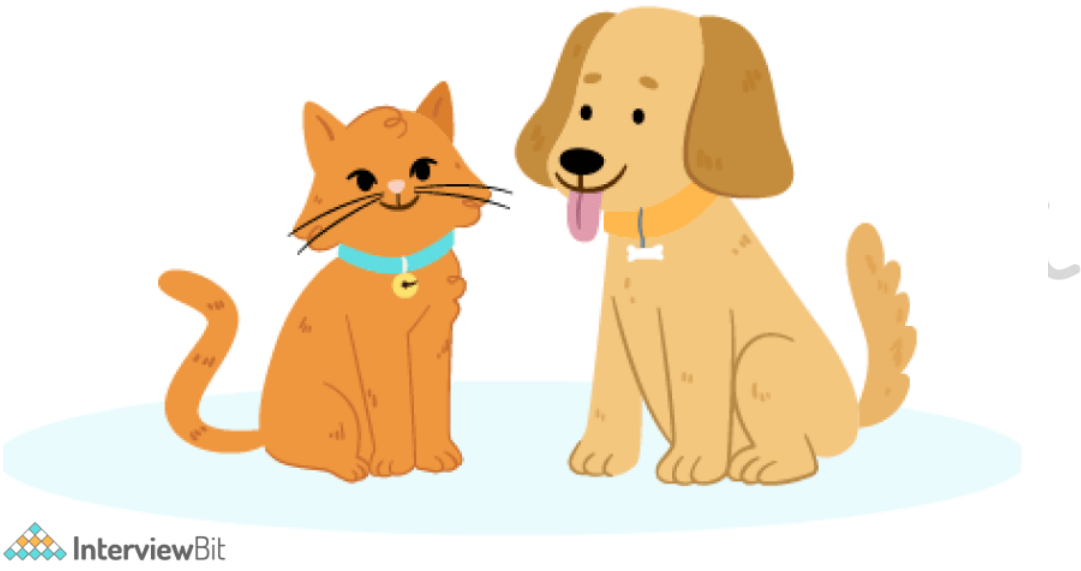
© Copyright by Interviewbit



The classification task in a multi-class classification problem has more than two mutually exclusive classes (classes that have no intersection or no attributes in common), whereas in a multi-label classification problem, each label has a different classification task, although the tasks are related in some way. For example, classifying a group of photographs of animals that could be cats, dogs, or bears is a multi-class classification problem that assumes each sample can be of only one type, implying that an image can be categorized as either a cat or a dog, but not both at the same time.

Now let us assume you wish to manipulate the image below.

© Copyright by Interviewbit

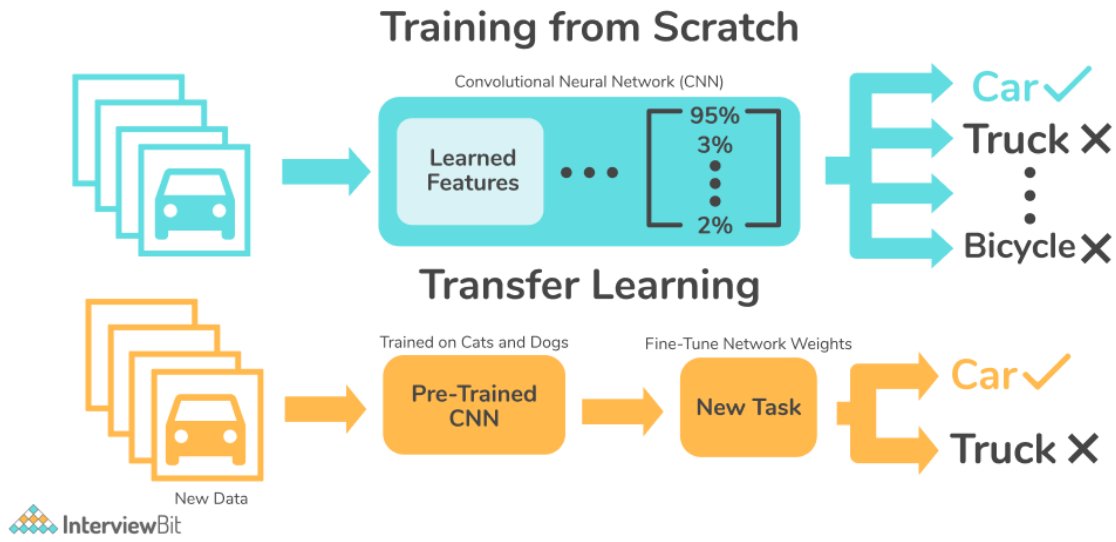


The image above must be categorized as both a cat and a dog because it depicts both creatures. A set of labels is allocated to each sample in a multi-label classification issue, and the classes are not mutually exclusive. In a multi-label classification problem, a pattern can belong to one or more classes.

15. Explain transfer learning in the context of deep learning.

Transfer learning is a learning technique that allows data scientists to use what they've learned from a previous machine learning model that was used for a similar task. The ability of humans to transfer their knowledge is used as an example in this learning. You can learn to operate other two-wheeled vehicles more simply if you learn to ride a bicycle. A model trained for autonomous automobile driving can also be used for autonomous truck driving. The features and weights can be used to train the new model, allowing it to be reused. When there is limited data, transfer learning works effectively for quickly training a model.

© Copyright by Interviewbit



In the above image, the first diagram represents training a model from scratch while the second diagram represents using a model already trained on cats and dogs to classify the different class of vehicles, thereby representing transfer learning.

16. What are the advantages of transfer learning?

Following are the advantages of transfer learning :

- **Better initial model:** In other methods of learning, you must create a model from scratch. Transfer learning is a better starting point because it allows us to perform tasks at a higher level without having to know the details of the starting model.
- **Higher learning rate:** Because the problem has already been taught for a similar task, transfer learning allows for a faster learning rate during training.
- **Higher accuracy after training:** Transfer learning allows a deep learning model to converge at a higher performance level, resulting in more accurate output, thanks to a better starting point and higher learning rate.

17. Is it possible to train a neural network model by setting all biases to 0? Also, is it possible to train a neural network model by setting all of the weights to 0?

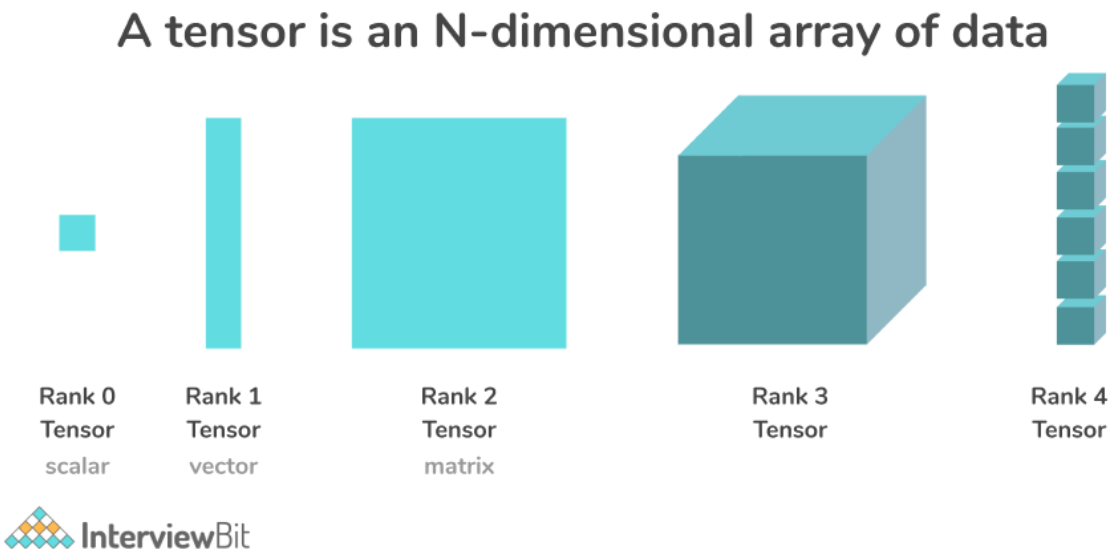
© Copyright by Interviewbit

Yes, even if all of the biases are set to zero, the neural network model has a chance of learning.

No, training a model by setting all of the weights to 0 is impossible since the neural network will never learn to complete a task. When all weights are set to zero, the derivatives for each w remain constant, resulting in neurons learning the same features in each iteration. Any constant initialization of weights, not simply zero, is likely to generate a poor result.

18. What is a tensor in deep learning?

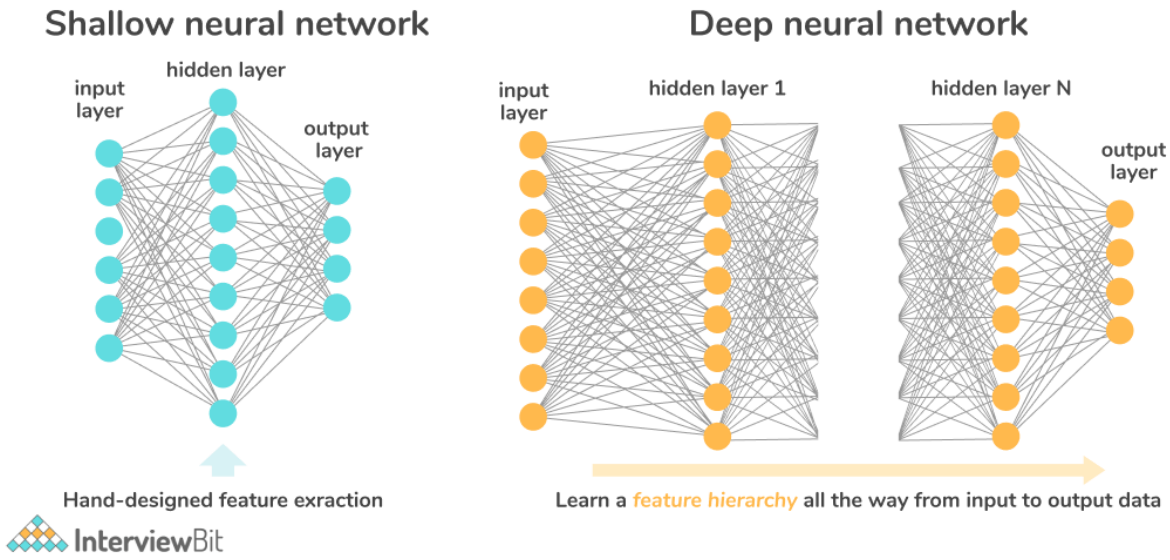
A tensor is a multidimensional array that represents a generalization of vectors and matrices. It is one of the key data structures used in deep learning. Tensors are represented as n-dimensional arrays of base data types. The data type of each element in the Tensor is the same, and the data type is always known. It's possible that only a portion of the shape (that is, the number of dimensions and the size of each dimension) is known. Most operations yield fully-known tensors if their inputs are likewise fully known, however, in other circumstances, the shape of a tensor can only be determined at graph execution time.



© Copyright by Interviewbit

19. Explain the difference between a shallow network and a deep network.

A hidden layer, as well as input and output layers, are present in every neural network. Shallow neural networks are those that have only one hidden layer, whereas deep neural networks include numerous hidden layers. Both shallow and deep networks can fit into any function, however, shallow networks require a large number of input parameters, whereas deep networks, because of their several layers, can fit functions with a small number of input parameters. Deep networks are currently favored over shallow networks because the model learns a new and abstract representation of the input at each layer. In comparison to shallow networks, they are also far more efficient in terms of the number of parameters and computations.



20. In a Convolutional Neural Network (CNN), how can you fix the constant validation accuracy?

© Copyright by Interviewbit

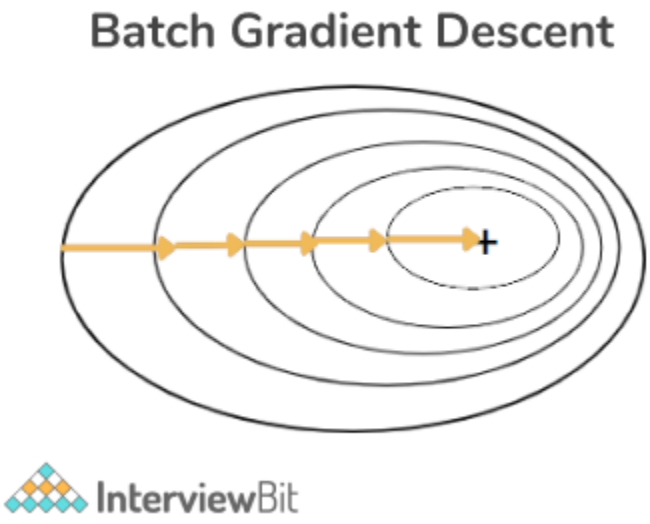
When training any neural network, constant validation accuracy is a common issue because the network just remembers the sample, resulting in an over-fitting problem. Over-fitting a model indicates that the neural network model performs admirably on the training sample, but the model's performance deteriorates on the validation set. Following are some ways for improving CNN's constant validation accuracy:

- It is always a good idea to split the dataset into three sections: training, validation, and testing.
- When working with limited data, this difficulty can be handled by experimenting with the neural network's parameters.
- By increasing the training dataset's size.
- By using batch normalization.
- By implementing regularization
- By reducing the complexity of the network

21. Explain Batch Gradient Descent.

Batch Gradient Descent: Batch Gradient Descent entails computation (involved in each step of gradient descent) over the entire training set at each step and hence it is highly slow on very big training sets. As a result, Batch Gradient Descent becomes extremely computationally expensive. This is ideal for error manifolds that are convex or somewhat smooth. Batch Gradient Descent also scales nicely as the number of features grows.

© Copyright by Interviewbit

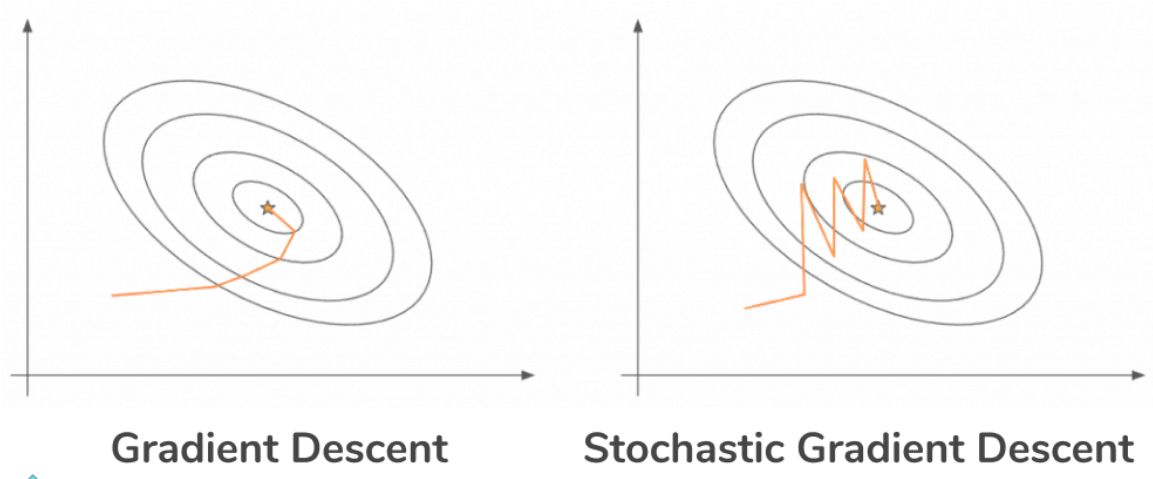


22. Explain Stochastic Gradient Descent. How is it different from Batch Gradient Descent?

Stochastic Gradient Descent: Stochastic Gradient Descent seeks to tackle the major difficulty with Batch Gradient Descent, which is the use of the entire training set to calculate gradients at each step. It is stochastic in nature, which means it chooses up a "random" instance of training data at each step and then computes the gradient, which is significantly faster than Batch Gradient Descent because there are much fewer data to modify at once. Stochastic Gradient Descent is best suited for unconstrained optimization problems. The stochastic nature of SGD has a drawback in that once it gets close to the minimum value, it doesn't settle down and instead bounces around, giving us a good but not optimal value for model parameters. This can be solved by lowering the learning rate at each step, which will reduce the bouncing and allow SGD to settle down at the global minimum after some time.

Following are the differences between the two:-

© Copyright by Interviewbit

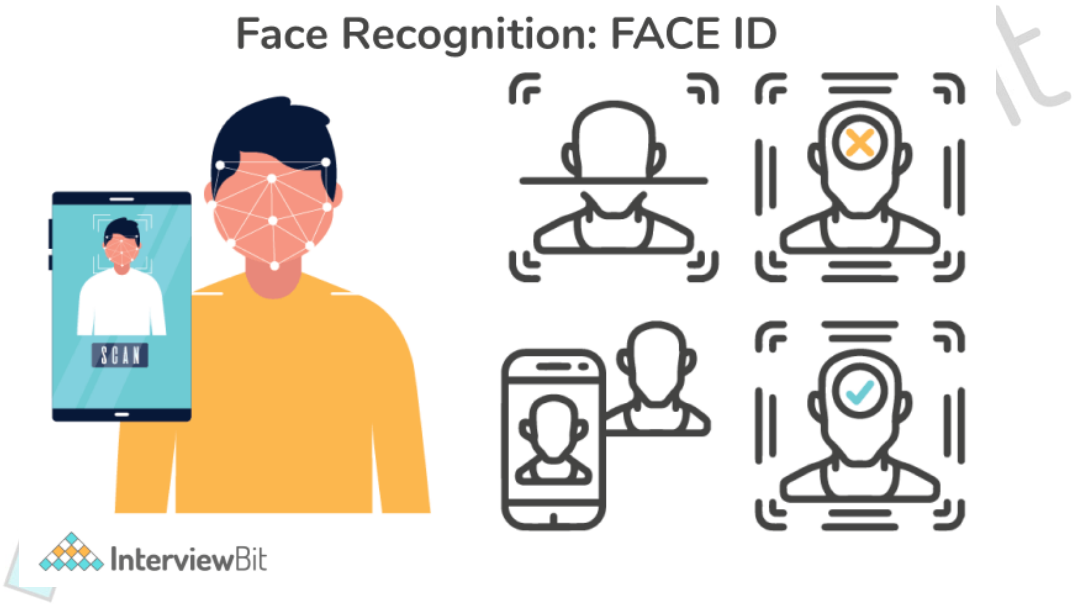


© Copyright by Interviewbit

Batch Gradient Descent	Stochastic Gradient Descent
The gradient is calculated using the entire training dataset.	A single training sample is used to compute the gradient.
It is slow and computationally more expensive than Stochastic Gradient Descent.	It is faster and less computationally expensive than Batch Gradient Descent.
It is not recommended for large training samples.	It is recommended for large training samples.
It is deterministic (not random) in nature.	It is stochastic (random) in nature.
Given enough time to converge, it returns the best answer.	It provides a good solution, but not the best.
There is no need to shuffle the data points at random.	Because we want the data sample to be in a random order, we'll shuffle the training set for each epoch.
In this, it is difficult to get out of shallow local minimas.	It has a better chance of escaping shallow local minima.
In this, the convergence is slow.	It arrives at the convergence point substantially faster.

© Copyright by Interviewbit

23. Which deep learning algorithm is the best for face detection?

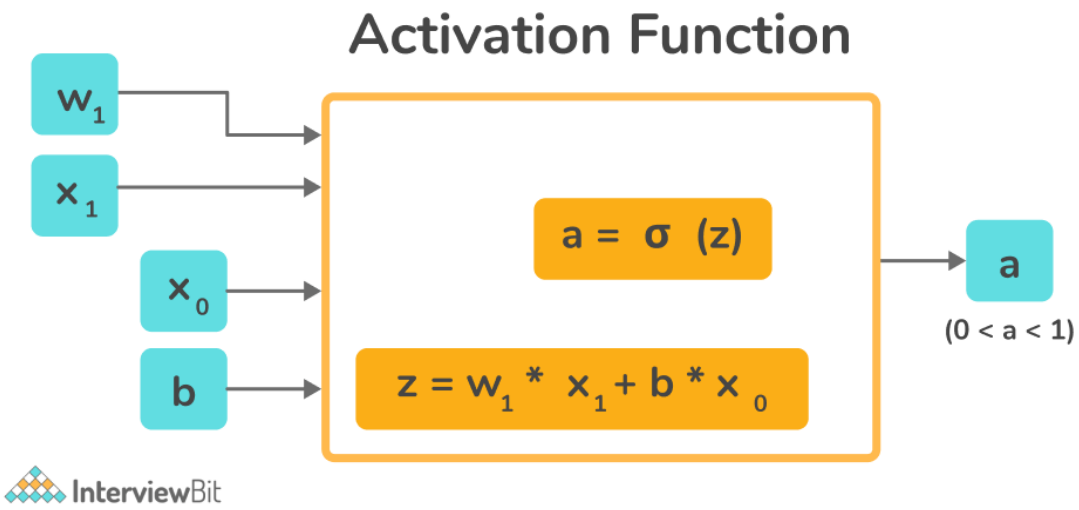


Face identification may be accomplished using a variety of machine learning methods, but the best ones use Convolutional Neural Networks and deep learning. The following are some notable face detection algorithms: FaceNet, Probablisit, Face Embedding, ArcFace, Cosface, and Spherface.

24. What is an activation function? What is the use of an activation function?

An artificial neural network's activation function is a function that is introduced to help the network learn complex patterns in the data. When compared to a neuron-based model seen in our brains, the activation function is responsible for determining what is to be fired to the next neuron at the end of the process. In an ANN, an activation function performs the same job. It takes the preceding cell's output signal and turns it into a format that may be used as input to the next cell.

© Copyright by Interviewbit



Here, x0 and x1 are the inputs. w1 is the weight and a is the activation function.

The activation function introduces non-linearity into the neural network, allowing it to learn more complex functions. The neural network would only be able to learn a function that is a linear combination of its input data if it didn't have the Activation function.

The activation function converts inputs to outputs. The activation function is in charge of determining whether or not a neuron should be stimulated. It arrives at a decision by calculating the weighted total and then adds bias. The activation function's main goal is to introduce non-linearity into a neuron's output.

25. What do you mean by an epochs in the context of deep learning?

An epoch is a terminology used in deep learning that refers to the number of passes the deep learning algorithm has made across the full training dataset. Batches are commonly used to group data sets (especially when the amount of data is very large). The term "iteration" refers to the process of running one batch through the model.

The number of epochs equals the number of iterations if the batch size is the entire training dataset. This is frequently not the case for practical reasons. Several epochs are used in the creation of many models.

© Copyright by Interviewbit

There is a general relation which is given by:-

d * e = i * b

where,

d is the dataset size

e is the number of epochs

i is the number of iterations

b is the batch size

Deep Learning Interview Questions for Experienced

26. While building a neural network architecture, how will you decide how many neurons and the hidden layers should the neural network have?

There is no clear and fast rule for determining the exact number of neurons and hidden layers required to design a neural network architecture given a business problem. The size of the hidden layer in a neural network should be somewhere between the size of the output layers and that of the input layers. However, there are a few basic ways that might help you get a head start on constructing a neural network architecture:

- The best method to approach any unique real-world predictive modelling problem is to start with some basic systematic experimentation to see what would perform best for any given dataset based on previous experience working with neural networks in similar real-world situations. The network configuration can be chosen based on one's understanding of the problem domain and previous expertise with neural networks. The number of layers and neurons employed on similar issues is always a good place to start when evaluating a neural network's configuration.
- It is best to start with simple neural network architecture and gradually increase the complexity of the neural network based on predicted output and accuracy.

© Copyright by Interviewbit

27. Can a deep learning model be solely built on linear regression?

Yes, if the problem is represented by a linear equation, deep networks can be built using a linear function as the activation function for each layer. A problem that is a composition of linear functions, on the other hand, is a linear function, and there is nothing spectacular that can be accomplished by implementing a deep network because adding more nodes to the network will not boost the machine learning model's predictive capacity.

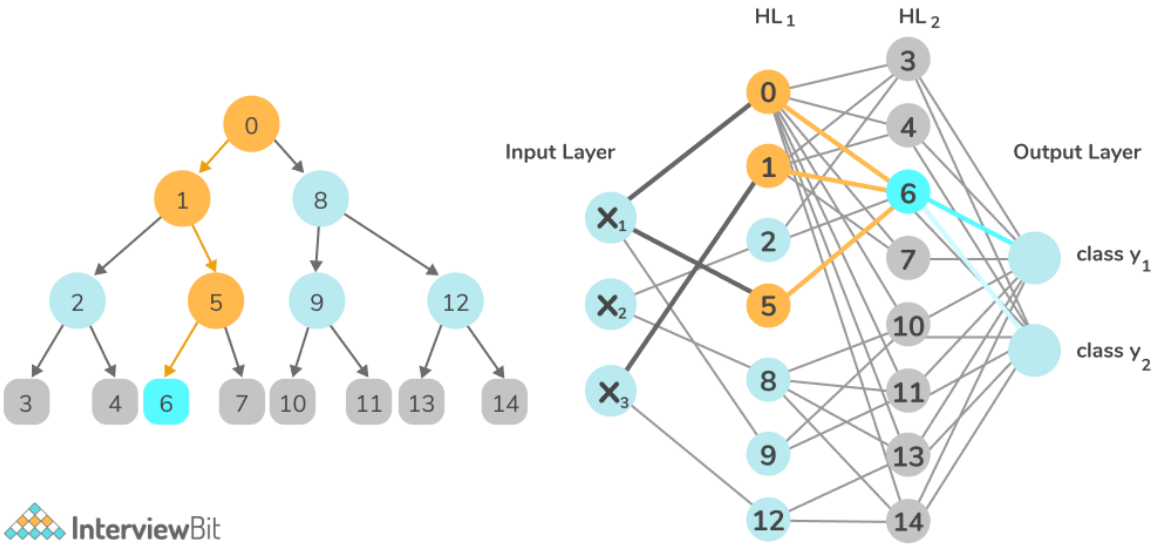
28. According to you, which one is more powerful - a two layer neural network without any activation function or a two layer decision tree?

A two-layer neural network is made up of three layers: one input layer, one hidden layer, and one output layer. When dealing with neural networks, an activation function is essential since it is required when dealing with complex and nonlinear functional mappings between inputs and response variables. When there is no activation function in a two-layer neural network, it is simply a linear network. A Neural Network without an Activation function is just a Linear Regression Model, which has limited capability and frequently fails to perform well.

A decision tree with a depth of two layers is known as a two-layer decision tree. Decision Trees are a type of supervised machine learning (that is, the machine is fed with what the input is and what the related output is in the training data) in which the data is continually split according to a parameter. Two entities, decision nodes, and leaves can be used to explain the tree. The decisions or final outcomes are represented by the leaves. And the data is separated at the decision nodes.

When comparing these two models, the two-layer neural network (without activation function) is more powerful than the two-layer decision tree, because the two-layer neural network will consider more attributes while building a model, whereas the two-layer decision tree will only consider 2 or 3 attributes.

© Copyright by Interviewbit



The figure on the left depicts a 2 layer decision tree and the figure on the right depicts a 2 layer neural network.

29. Differentiate between bias and variance in the context of deep learning models. How can you achieve balance between the two?

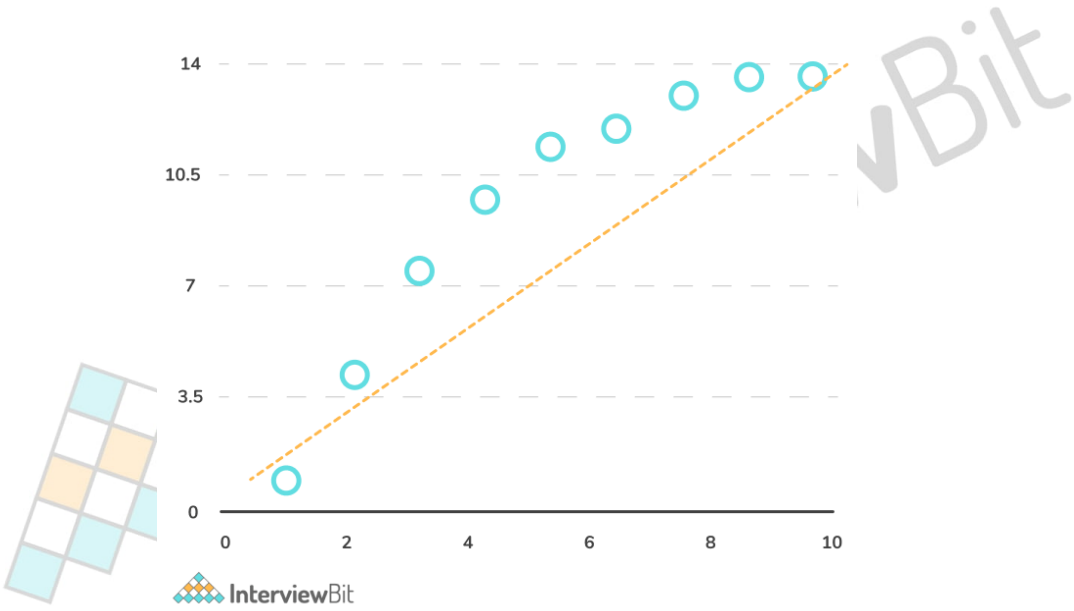
Comprehending prediction errors is crucial when it comes to understanding predictions. Reducible (errors that arise due to squared bias or squared variance) and irreducible (errors that arise due to the randomness or natural variability in a system and cannot be reduced by varying the model) mistakes are the two primary types of errors. There are two types of reducible errors: bias and variance. Gaining a thorough grasp of these flaws aids in the construction of an accurate model by preventing overfitting and underfitting.

Bias:

The bias is defined as the difference between the ML model's predicted values and the actual value. Biasing results in a substantial inaccuracy in both training and testing data. To avoid the problem of underfitting, it is advised that an algorithm be low biassed at all times.

© Copyright by Interviewbit

The data predicted is in a straight line format due to significant bias, and hence does not fit accurately in the data set. Underfitting of data is the term for this type of fitting. This occurs when the theory is too straightforward or linear. Consider the graph below as an illustration of a situation like this.



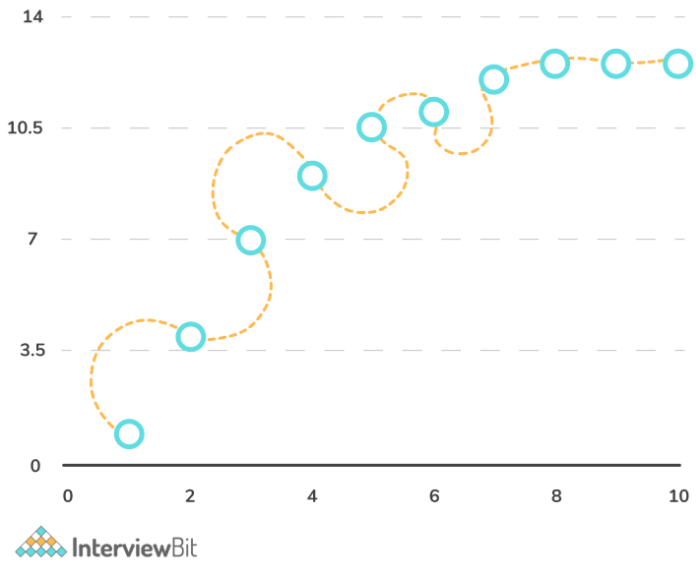
Variance:

The variance of the model is the variability of model prediction for a given data point, which tells us about the dispersion of our data. It is the difference between the validation error and the training error. The model with high variance has a very complex fit to the training data and so is unable to fit accurately on new data. As a result, while such models perform well on training data, they have high error rates when testing data.

When a model's variance is excessive, it's referred to as Overfitting of Data. Overfitting, which involves accurately fitting the training set using a complicated curve and a high order hypothesis, is not a viable option because the error with unknown data is considerable.

Variance should be kept to a minimum when training a data model.

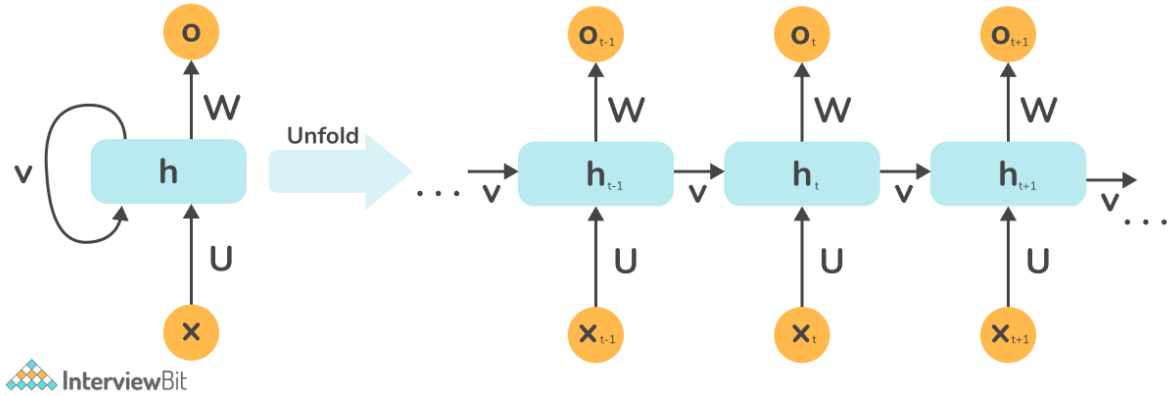
© Copyright by Interviewbit



The model must always aim for a low bias and a low variance in order to achieve the best balance between the two mistakes.

30. How does Recurrent Neural Network backpropagation vary from Artificial Neural Network backpropagation?

Backpropagation in Recurrent Neural Networks differ from that of Artificial Neural Networks in the sense that each node in Recurrent Neural Networks has an additional loop as shown in the following image:



© Copyright by Interviewbit

This loop, in essence, incorporates a temporal component into the network. This allows for the capture of sequential information from data, which is impossible with a generic artificial neural network.

31. What exactly do you mean by exploding and vanishing gradients?

By taking incremental steps towards the minimal value, the gradient descent algorithm aims to minimize the error. The weights and biases in a neural network are updated using these processes.

However, at times, the steps grow excessively large, resulting in increased updates to weights and bias terms — to the point where the weights overflow (or become NaN, that is, Not a Number). An exploding gradient is the result of this, and it is an unstable method.

On the other hand, if the steps are excessively small, it results in minor – even negligible – changes in the weights and bias terms. As a result, we may end up training a deep learning model with nearly identical weights and biases every time, never reaching the least error function. The vanishing gradient is what it's called.

32. What are autoencoders? Explain the different layers of autoencoders.

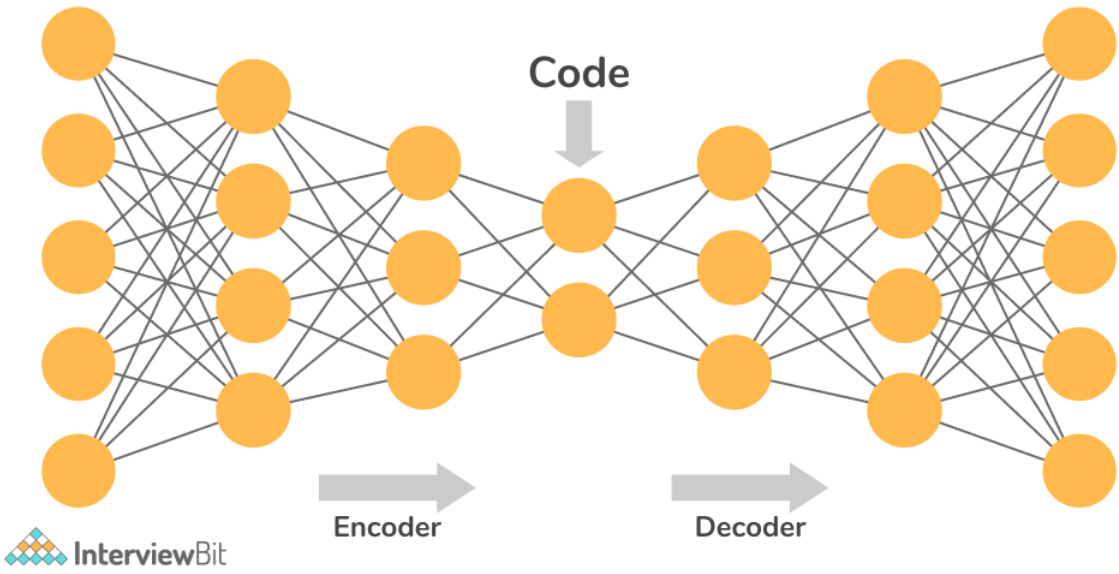
An autoencoder is a type of neural network with the condition that the output layer has the same dimension as that of the input layer. In other words, the number of output units in the output layer is equal to the number of input units in the input layer. An autoencoder is also known as a replicator neural network since it duplicates data from the input to the output in an unsupervised way.

© Copyright by Interviewbit

By sending the input through the network, the autoencoders rebuild each dimension of the input. It may appear simple to use a neural network to replicate an input, however, the size of the input is reduced during the replication process, resulting in a smaller representation. In comparison to the input and output layers, the middle layers of the neural network have fewer units. As a result, the reduced representation of the input is stored in the middle layers. This reduced representation of the input is used to recreate the output.

Following are the different layers in the architecture of autoencoders :

- **Encoder:** An encoder is a fully connected, feedforward neural network that compresses the input image into a latent space representation and encodes it as a compressed representation in a lower dimension. The deformed representation of the original image is the compressed image.
- **Code:** The reduced representation of the input that is supplied into the decoder is stored in this section of the network.
- **Decoder:** Like the encoder, the decoder is a feedforward network with a structure identical to the encoder. This network is in charge of reassembling the input from the code to its original dimensions.



© Copyright by Interviewbit

As we can see in the above image, the input is compressed in the encoder, then stored in the Code, and then the original input is decompressed from the code by the decoder. The autoencoder's principal goal is to provide an output that is identical to the input.

33. Mention the applications of autoencoders.

Following are the applications of autoencoders:-

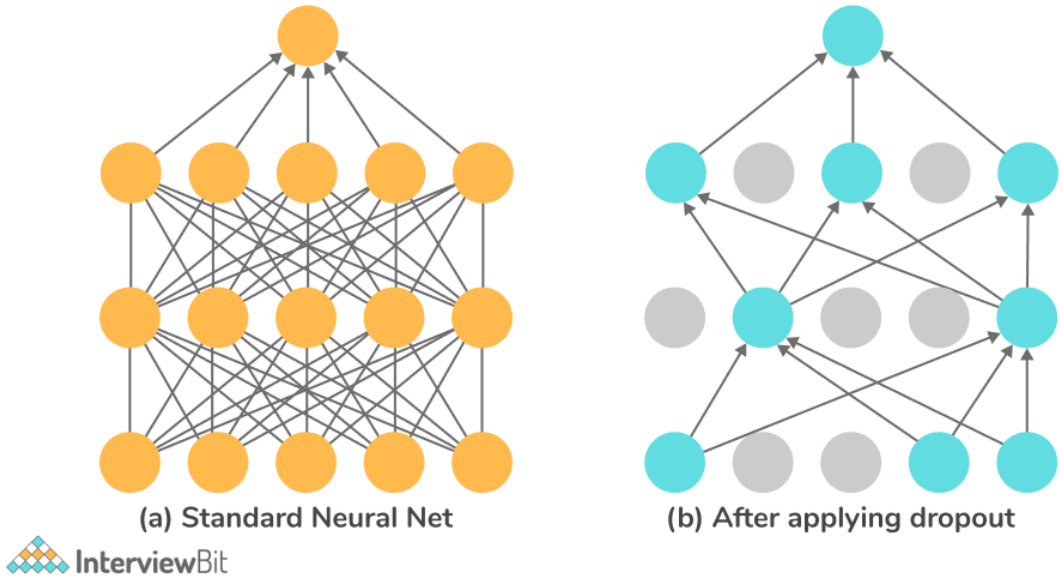
© Copyright by Interviewbit

- **Image Denoising:** Denoising images is a skill that autoencoders excel at. A noisy image is one that has been corrupted or has a little amount of noise (that is, random variation of brightness or color information in images) in it. Image denoising is used to gain accurate information about the image's content.
- **Dimensionality Reduction:** The input is converted into a reduced representation by the autoencoders, which is stored in the middle layer called code. This is where the information from the input has been compressed, and each node may now be treated as a variable by extracting this layer from the model. As a result, we can deduce that by removing the decoder, an autoencoder can be utilised for dimensionality reduction, with the coding layer as the output.
- **Feature Extraction:** The encoding section of Autoencoders aids in the learning of crucial hidden features present in the input data, lowering the reconstruction error. During encoding, a new collection of original feature combinations is created.
- **Image Colorization:** Converting a black-and-white image to a coloured one is one of the applications of autoencoders. We can also convert a colourful image to grayscale.
- **Data Compression:** Autoencoders can be used for data compression. Yet they are rarely used for data compression because of the following reasons:
    - **Lossy compression:** The autoencoder's output is not identical to the input, but it is a near but degraded representation. They are not the best option for lossless compression.
    - **Data-specific:** Autoencoders can only compress data that is identical to the data on which they were trained. They differ from traditional data compression algorithms like jpeg or gzip in that they learn features relevant to the provided training data. As a result, we can't anticipate a landscape photo to be compressed by an autoencoder trained on handwritten digits.

34. What do you know about Dropout?

© Copyright by Interviewbit

Dropout is a regularization approach that helps to avoid overfitting and hence improves generalizability (that is, the model predicts correct output for most of the inputs in general, rather than only being limited to the training data set). In general, we should utilize a low dropout value of 20 percent to 50 percent of neurons, with 20% being a decent starting point. A probability that is too low has no effect, whereas a number that is too high causes the network to under-learn.



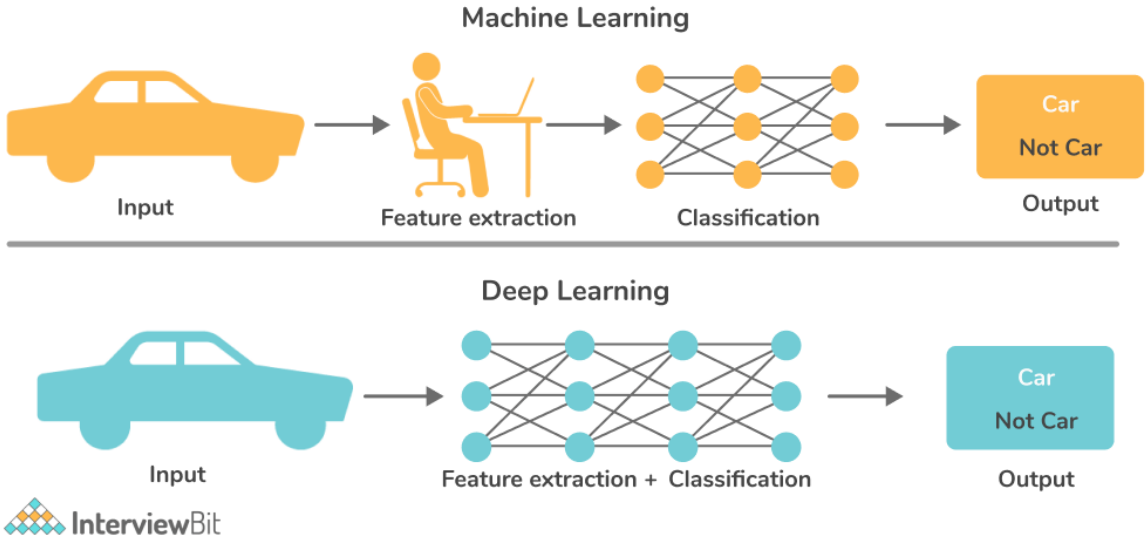
When you employ dropout on a larger network, you're more likely to achieve better results because the model has more opportunities to learn independent representations.

35. Differentiate between Deep Learning and Machine Learning.

© Copyright by Interviewbit

Deep Learning: Deep Learning is a subclass of Machine Learning in which a recurrent neural network and an artificial neural network are linked. The algorithms are constructed in the same way as machine learning algorithms are, however, there are many more levels of algorithms. The artificial neural network refers to all of the algorithm's networks put together. In much simpler terms, it mimics the human brain by connecting all of the neural networks in the brain, which is the concept of deep learning. It uses algorithms and a technique to tackle all types of complex problems.

Machine Learning: Machine learning is a subset of Artificial Intelligence (AI) that allows a system to learn and grow from its experiences without having to be programmed to that level. Data is used by Machine Learning to learn and get accurate outcomes. Machine learning algorithms have the ability to learn and improve their performance by gaining more data. Machine learning is currently employed in self-driving cars, cyber fraud detection, face recognition, and Facebook friend suggestion, among other applications. Learn More.



The following table illustrates the difference between them:

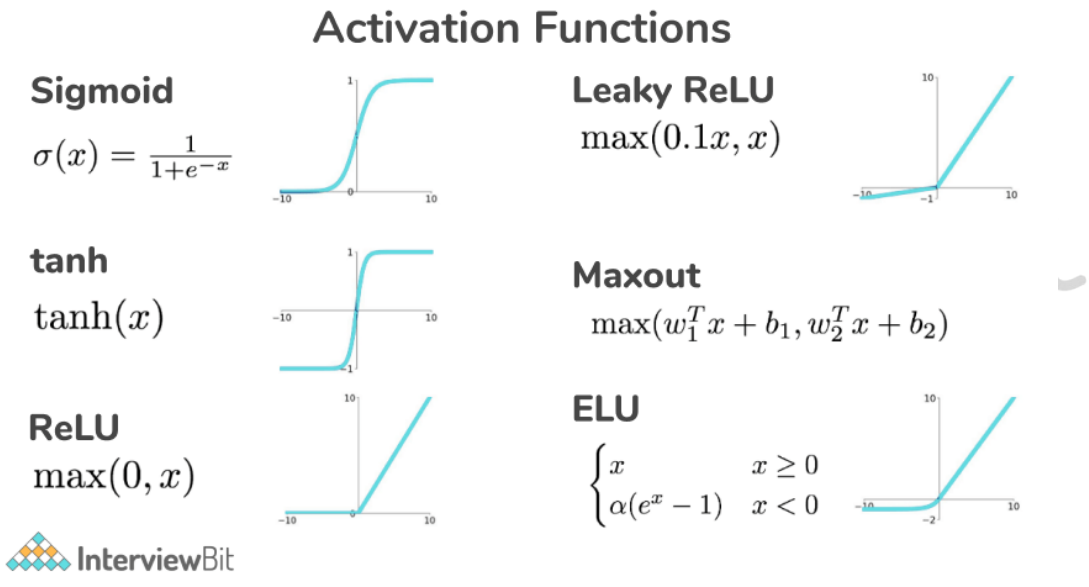
© Copyright by Interviewbit

Deep Learning	Machine Learning
Deep Learning is a subclass of Machine Learning.	Machine Learning is a super-class of Deep Learning.
Deep Learning employs neural networks to represent data, which is a very distinct data representation (ANN).	Machine Learning represents data in a different way than Deep Learning since it uses structured data.
In this, the output ranges from numerical values to free-form elements such as text or sound.	In this, the output consists of numerical values
It uses a neural network to evaluate data features and relationships by passing data through processing layers.	It uses a variety of automated techniques to convert input into model functions and forecast future actions.
It usually deals with millions of data points.	It usually deals with thousands of data points.
Machine Learning has evolved into Deep Learning. Essentially, it refers to the depth of machine learning.	Artificial Intelligence has evolved into Machine Learning.

36. Explain the different types of activation functions.

Following are the different types of activation functions:

© Copyright by Interviewbit



Sigmoid function: The sigmoid function is a non-linear activation function in an ANN that is mostly utilised in feedforward neural networks. It's a differentiable real function with positive derivatives everywhere and a certain degree of smoothness, defined for real input values. The sigmoid function is found in the deep learning models' output layer and is used to anticipate probability-based outputs. The sigmoid function is written as follows:

{"detectHand":false}

Hyperbolic Tangent Function (Tanh): The Tanh function is a smoother and zero-centered function having a range of -1 to 1. The output of the tanh function is represented by:

{"detectHand":false}

Because it provides higher training performance for multilayer neural networks, the tanh function is considerably more widely utilised than the sigmoid function. The tanh function's primary advantage is that it gives a zero-centered output, which helps with backpropagation.

© Copyright by Interviewbit

Softmax function: The softmax function is another type of activation function used in neural networks to generate probability distribution from a vector of real numbers. This function returns a number between 0 and 1, with the sum of the probabilities equal to 1. The softmax function is written like this:

{"detectHand":false}
This function is most commonly used in multi-class models, returning probabilities for each class, with the target class having the highest probability. It can be found in practically all of the output layers of the DL architecture.

Softsign function: This is most commonly used in regression computation issues and text-to-speech applications based on deep learning. It's a quadratic polynomial with the following representation:

{"detectHand":false}

Rectified Linear Unit Function: The rectified linear unit (ReLU) function is a fast-learning artificial intelligence (AI) that promises to give cutting-edge performance and outstanding results. In deep learning, the ReLU function outperforms other AFs like the sigmoid and tanh functions in terms of performance and generalisation. The function is a roughly linear function that preserves the features of linear models, making gradient-descent approaches easier to optimise.

On each input element, the ReLU function performs a threshold operation, setting all values less than zero to zero. As a result, the ReLU is written as:

{"detectHand":false}

Exponential Linear Unit Function: The exponential linear units (ELUs) function is a type of AF that can be used to speed up neural network training (just like ReLU function). The ELU function's major advantage is that it can solve the vanishing gradient problem by employing identity for positive values and boosting the model's learning properties. The exponential linear unit function has the following representation:

{"detectHand":false}

© Copyright by Interviewbit

Links to More Interview Questions

C Interview Questions	Php Interview Questions	C Sharp Interview Questions
Web Api Interview Questions	Hibernate Interview Questions	Node Js Interview Questions
Cpp Interview Questions	Oops Interview Questions	Devops Interview Questions
Machine Learning Interview Questions	Docker Interview Questions	Mysql Interview Questions
Css Interview Questions	Laravel Interview Questions	Asp Net Interview Questions
Django Interview Questions	Dot Net Interview Questions	Kubernetes Interview Questions
Operating System Interview Questions	React Native Interview Questions	Aws Interview Questions
Git Interview Questions	Java 8 Interview Questions	Mongodb Interview Questions
Dbms Interview Questions	Spring Boot Interview Questions	Power Bi Interview Questions
Pl Sql Interview Questions	Tableau Interview Questions	Linux Interview Questions
Ansible Interview Questions	Java Interview Questions	Jenkins Interview Questions

© Copyright by Interviewbit