

PYTORCH INTERVIEW QUESTIONS

Q1: What is PyTorch?

Answer: PyTorch is a machine learning library for the programming language Python, based on Torch library, used for application such as natural language processing. It is free and open source software release under one of the BSD licenses. It has been released in October 2016 written in Python, C++, and CUDA.

Q2: Who is the founder of PyTorch?

Answer: Soumith Chintala and Hugh parkins (AI researcher in Facebook) are the founders of PyTorch.

Q3: What are the significant features of PyTorch?

Answer: The features of PyTorch are as follows:

Easy interface: PyTorch offers easy to use API, and it is straightforward to operate and run on Python. The code execution is smooth. Python usage: This library is considered to be Pythonic, which smoothly integrates the Python data science stack. Computational Graphs: PyTorch provides an excellent platform which offers dynamic computational graphs. So that a user can change them during runtime, this is more useful when a developer has no idea of how much memory is required for creating a neural network model. Imperative Programming: PyTorch performs computations through each line of the written code. This is similar to Python program execution.

Q4: What are the three levels of abstraction?

Answer: Levels of abstraction are as follows:

Tensor- Imperative n-dimensional Array which runs on GPU. Variable- Node in the computational graph. This stores data and gradient. Module- Neural network layer will store state otherwise learnable weights.

Q5: What are the advantages of PyTorch?

Answer: The following are the advantages of PyTorch:

It is easy to debug. It includes many layers as Torch. It can be considered as NumPy extension to GPU (Graphical Processing Units). It allows building network whose structure is dependent on computation itself. It includes a lot of loss functions.

Q7: What is Artificial Intelligence?

Answer: Artificial intelligence is an excellent area of computer science that highlighted the creation of intelligent machines that work and reacts like humans. It can perform the task typically requiring human knowledge, such as visual perception, speech recognition, decision-making, etc.

Q8: What is Machine learning?

Answer: Machine learning is an application of artificial intelligence (AI) that provides that systems automatically learn and improve from experience without being programmed. Machine learning points on the development of computer programs that can access data and use its trend for themselves.

Q9: What is Deep learning?

Answer: Deep learning is a critical subset of machine learning that imitates the working of the human brain in processing data and creating patterns for use in decision making. Deep

learning has network learning capability of learning unsupervised from data that is unstructured or unlabeled. It uses the concept of neural networks to solve complex problems.

Q11: Explain the following three variants of gradient descent: batch, stochastic, and mini-batch?

Answer:

Stochastic Gradient Descent: Here, we use only a single training example for calculation of gradient and parameters. Batch Gradient Descent: We calculate the gradient for the whole dataset and perform the update at each iteration. Mini-batch Gradient Descent: It's a variant of Stochastic Gradient Descent and here instead of single training example, mini-batch of samples is used.

Q12: What is Backpropagation?

Answer: Backpropagation is a training algorithm used for multiple users for a many layer neural network. In this method, we move the error into the end of the net to all weights inside the system and allowing efficient calculation of the gradient.

It is divided into several steps as follows:-

Forward propagation of training data to generate output. Then by using target value and output value error derivative can be computed concerning output activation. Then we back produce for computing derivative of the error concerning output activation on previous and continue this for all the hidden layers. Using previously solved derivatives for output and all the hidden layers, we calculate error derivatives. And then we update the weights.

Q13: What are the benefits of mini-batch gradient descent?

Answer: The benefits of mini-batch gradient descent are as follow:

It is more efficient compared to stochastic gradient descent. The generalization is maintained by discovering the flat minima. Mini-batches allow help to approximate the gradient of the entire training set, which helps us to avoid local minima.

Q14: What is auto-encoder?

Answer: An auto-encoder is a self-government machine learning algorithm that uses the backpropagation principle, where the target values are equal to the inputs provided.

Internally, it has a hidden layer that manages a code used to represent the input.

pytorch interview question 2

Q15: What is gradient descent?

Answer: Gradient descent is an optimization algorithm, which is used to learn the value of parameters that controls the cost function. It is a repetitive algorithm which moves in the direction of vertical descent as defined by the negative of the gradient.

Q16: What are the essential elements of Pytorch?

Answer: Main elements of PyTorch are as follows:

PyTorch tensors PyTorch NumPy Mathematical operations Autograd Module Optim Module nn Module

Q17: What are tensors in PyTorch?

Answer: A Tensor is a multi-dimensional matrix containing elements of a single data type.

Tensors in PyTorch are same as NumPy array. Its manually compute the forward pass, loss, and backward pass. The most significant difference between the PyTorch Tensors and NumPy Array is that Pytorch can run either in CPU or GPU. To run operations on GPU, just cast the sensor in the file system.

```
# import PyTorch
```

```
import torch
```

```
# define a tensor
```

```
torch.FloatTensor([2])
```

```
2
```

Torch. float tensor of size 1

```
# import PyTorch
```

```
import torch
```

```
# define a tensor
```

```
torch.FloatTensor([2])
```

```
2
```

Torch. float tensor of size 1

Q18: What are the mathematical building blocks of Neural Networks?

Answer: Mathematics is a vital role in any machine learning algorithm and includes many core concepts of mathematics to get the right algorithm.

The essential elements of machine learning and data science are as follows:

Vectors: Vector is considered to be an array of numbers which is continuous or discrete, and space which consists of vectors is called a vector space.

Scalars: Scalars are termed to have zero dimensions containing only one value. When it comes to PyTorch, it does not include a particular tensor with zero dimensions.

Matrices: In matrices, most of the structured data is usually represented in the form of tables or a specific model.

Q19: What is Autograd module in PyTorch?

Answer: There is an automatic differentiation technique used in PyTorch. This technique is more powerful when we are building a neural network. There is a recorder which records what operations we have performed, and then it replays it backs to compute our gradient.

Q20: What is the Optim Module in PyTorch?

Answer: Torch.optim is a module that implements various optimization algorithm used for building neural networks. Most of the commonly used syntax is already supported.

Below is the code of Adam optimizer

```
Optimizer = torch.optim.Adam(model, parameters(), lr=learning rate
```

```
1
```

```
2
```

```
3
```

```
Optimizer = torch.optim.Adam(model, parameters(), lr=learning rate
```

Q21: What is nn Module in PyTorch?

Answer: nn module: The nn package define a set of modules, which are thought of as a neural network layer that produce output from the input and have some trainable weights.

It is a type of tensor that considers a module parameter. Parameters are tensors subclasses. A fully connected ReLU networks where one hidden layer, trained to predict y from x to minimizing the square distance.

Example:

```
Import torch
```

```
# define model
```

```
model= torch.nn.Sequential(
```

```
torch.nn.Linear(hidden_num_units, hidden_num_units),
```

```
torch.nn.ReLU(),
```

```
torch.nn.Linear(hidden_num_units, output_num_units),
```

```
)
```

```
loss_fn= torch.nn.crossEntropyLoss()
```

```
Import torch
```

```
# define model
```

```
model= torch.nn.Sequential(
```

```
torch.nn.Linear(hidden_num_units, hidden_num_units),
```

```
torch.nn.ReLU(),
```

```
torch.nn.Linear(hidden_num_units, output_num_units),
)
```

```
loss_fn= torch.nn.crossEntropyLoss( )
```

Q22: What are the advantages of PyTorch?

Answer: Advantages of PyTorch are:

It is easy to debug. It includes many layers as Torch. It includes many layers as Torch. It allows building networks whose network is dependent upon computation itself. It has a simple interface and easily usable API.

Q23: Write the installation steps of PyTorch?

Answer: Installing PyTorch with Anaconda and Conda

Download Anaconda and install (Go with the latest Python version). Go to the Getting Started section on the PyTorch website through pytorch.org. Generate the appropriate configuration options for your particular environment. For example: OS: Windows Package Manager: conda Python: 3.6 CUDA: 9.0 Run the below command in the terminal(CMD) to install PyTorch. For example, the configuration we specified in step(3), we have the following command:

```
> conda install PyTorch -c PyTorch
```

```
> pip3 install torchvision
```

```
1
```

```
2
```

```
3
```

```
4
```

```
> conda install PyTorch -c PyTorch
```

```
> pip3 install torchvision
```

Q24: How to build a neural network through PyTorch?

Answer: A PyTorch implementation of the neural network looks the same as a NumPy implementation. The motive of this section is to showcase the similar nature of PyTorch and NumPy. For example: create a three-layered network having five nodes in the input layer, three in the hidden layer, and one in the output layer.

Q25: Why use PyTorch for Deep learning?

Answer: PyTorch is the essential part of deep learning tool. Deep learning is a subset of machine learning, which algorithm works on the human brain. These algorithms are known as artificial neural networks. That neural network is used for image classification, Artificial Neural Networks, and Recurrent Neural Networks. Unlike other libraries like TensorFlow where you have first to define an entire computation graph before you can run your model.

Q26: What are the differences between an array and a linked-list?

Answer: Collection of the object in a well-defined order is known as Array. And the linked list is also a set of objects but they are not in a well-defined form or remain in sequence, and also they have a pointer which is not in case of Array.

Q27: Name the two standard regularizations in machine learning approach?

Answer: Two common regularizations are L1 and L2. Both have their distinct functions. L1 contains many variables which are in binary numbers. L2 are meant for error handling, and both of them are related to the Gaussian concept.

Q28: What do you mean by Supervised and Unsupervised learning?

Answer: Supervised learning is a type of education where we teach or trained the machine using data which is well labeled, and some information is already tagged with the correct answer. So that supervised learning algorithm analyses the training data.

Unsupervised learning is an essential type of machine learning where there is neither classified nor labeled and allow to act on that information without guidance.

Q28: what is the difference between Type I and Type II error?

Answer: Type I error is the false positive value. And Type 1 error is a false negative value. Type I error represent something is happening when. Type II errors are describing that there nothing is wrong where something is not right.

Q29: What do you know about Recall and precision?

Answer: Recall is known as an exact positive rate. Precision is generally a predictive value, which is positive.

Q30: Are tensor and matrix the same?

We can't say that tensor and matrix are the same. Tensor has some properties through which we can say both have some similarities such as we can perform all the mathematical operation of the matrix in tensor.

A tensor is a mathematical entity which lives in a structure and interacts with other mathematical entity. If we transform the other entities in the structure in a regular way, then the tensor will obey a related transformation rule. This dynamical property of tensor makes it different from the matrix.

Q31: What is the use of torch.from_numpy()?

The torch.from_numpy() is one of the important property of torch which places an important role in tensor programming. It is used to create a tensor from numpy.ndarray. The ndarray and return tensor share the same memory. If we do any changes in the returned tensor, then it will reflect the ndarray also.

Q32: What is variable and autograd.variable?

Variable is a package which is used to wrap a tensor. The autograd.variable is the central class for the package. The torch.autograd provides classes and functions for implementing automatic differentiation of arbitrary scalar-valued functions. It needs minimal changes to the existing code. We only need to declare tensor for which gradients should be computed with the requires_grad=True keyword.

Q33: How do we find the derivatives of the function in PyTorch? The derivatives of the function are calculated with the help of the Gradient. There are four simple steps through which we can calculate derivative easily.

These steps are as follows:

Initialization of the function for which we will calculate the derivatives. Set the value of the variable which is used in the function. Compute the derivative of the function by using the backward () method. Print the value of the derivative using grad.

Q34: What do you mean by Linear Regression?

Linear Regression is a technique or way to find the linear relation between the dependent variable and the independent variable by minimizing the distance. It is a supervised machine learning approach which is used for classification of order discrete category.

Q35: What is Loss Function?

The loss function is bread and butter for machine learning. It is quite simple to understand and used to evaluate how well our algorithm models our dataset. If our prediction is completely off, then the function will output a higher number else it will output a lower number.

Q36: What is the use of MSELoss, CTCLoss, and BCELoss function?

MSE stands for Mean Squared Error, which is used to create a criterion the measures the mean squared error between each element in an input x and target y. The CTCLoss stands for

Connectionist Temporal Classification Loss, which is used to calculate the loss between continuous time series and target sequence. The BCELoss(Binary Cross Entropy) is used to create a criterion to measures the Binary Cross Entropy between the target and the output.

Q37: Give any one difference between torch.nn and torch.nn.functional?

The torch.nn provide us many more classes and modules to implement and train the neural network. The torch.nn.functional contains some useful function like activation function and convolution operation, which we can use. However, these are not full layers, so if we want to define a layer of any kind, we have to use torch.nn.

Q38:What do you mean by Mean Squared Error?

The mean squared error tells us about how close a regression line to a set of points. Mean squared error done this by taking the distance from the points to the regression line and squaring them. Squaring is required to remove any negative signs.

Q39: What is perceptron?

Perceptron is a single layer neural network, or we can say a neural network is a multi-layer perceptron. Perceptron is a binary classifier, and it is used in supervised learning. A simple model of a biological neuron in an artificial neural network is known as Perceptron

Q40: What is Activation function?

A neuron should be activated or not, is determined by an activation function. The Activation function calculates a weighted sum and further adding bias with it to give the result. The Neural Network is based on the Perceptron, so if we want to know the working of the neural network, then we have to learn how perceptron work.

Q41: How Neural Network differs from Deep Neural Network?

Neural Network and Deep Neural Network both are similar and do the same thing. The difference between NN and DNN is that there can be only one hidden layer in the neural network, but in a deep neural network, there is more than one hidden layer. Hidden layers play an important role to make an accurate prediction.

Q42: Why it is difficult for the network is showing the problem?

Ann works with numerical information, and the problems are translated into numeric values before being introduced to ANN. This is the reason by which it is difficult to show the problem to the network.

Q43: Why we used activation function in Neural Network

To determine the output of the neural network, we use the Activation Function. Its main task is to do mapping of resulting values in between 0 to 1 or -1 to 1 etc. The activation functions are basically divided into two types:

Linear Activation Function Non-linear Activation Function

Q44: Why we prefer the sigmoid activation function rather than other function?

The Sigmoid Function curve looks like S-shape and the reason why we prefer sigmoid rather than other is the sigmoid function exists between 0 to 1. This is especially used for the models where we have to predict the probability as an output.

Q45: What do you mean by Feed-Forward?

"Feed-Forward" is a process through which we receive an input to produce some kind of output to make some kind of prediction. It is the core of many other important neural networks such as convolution neural network and deep neural network./p>

In the feed-forward neural network, there are no feedback loops or connections in the network. Here is simply an input layer, a hidden layer, and an output layer.

Q46: What is the difference between Conv1d, Conv2d, and Conv3d?

There is no big difference between the three of them. The Conv1d and Conv2D is used to apply 1D and 2D convolution. The Conv3D is used to apply 3D convolution over an input signal composed of several input planes.

Q47: What do you understand from the word Backpropagation?

"Backpropagation" is a set of algorithm which is used to calculate the gradient of the error function. This algorithm can be written as a function of the neural network. These algorithms are a set of methods which are used to efficiently train artificial neural networks following a gradient descent approach which exploits the chain rule.

Q48: What is Convolutional Neural Network?

Convolutional Neural Network is the category to do image classification and image recognition in neural networks. Face recognition, scene labeling, objects detections, etc., are the areas where convolutional neural networks are widely used. The CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc.

Q49: What is the difference between DNN and CNN?

The deep neural network is a kind of neural network with many layers. "Deep" means that the neural network has a lot of layers which looks deep stuck of layers in the network. The convolutional neural network is another kind of deep neural network. The Convolutional Neural Network has a convolution layer, which is used filters to convolve an area in input data to a smaller area, detecting important or specific part within the area. The convolution can be used on the image as well as text.

Q50: What are the advantages of PyTorch?

There are the following advantages of Pytorch:

PyTorch is very easy to debug. It is a dynamic approach for graph computation. It is very fast deep learning training than TensorFlow. It increased developer productivity. It is very easy to learn and simpler to code.

Q51: What is the MNIST dataset?

The MNIST dataset is used in Image Recognition. It is a database of various handwritten digits. The MNIST dataset has a large amount of data which is commonly used to demonstrate the true power of deep neural networks.

Q52: What is the CIFAR-10 dataset?

It is a collection of the color image which is commonly used to train machine learning and computer vision algorithms. The CIFAR 10 dataset contains 50000 training images and 10000 validation images such that the images can be classified between 10 different classes.

Q53: What is the difference between CIFAR-10 and CIFAR-100 dataset?

The CIFAR 10 dataset contains 50000 training images and 10000 validation images such that the images can be classified between 10 different classes. On the other hand, CIFAR-100 has 100 classes, which contain 600 images per class. There are 100 testing images and 500 training images per class.

Q54: What do you mean by convolution layer?

Convolution layer is the first layer in Convolutional Neural Network. It is the layer to extract the features from an input image. It is a mathematical operation which takes two inputs such as image matrix and a kernel or filter.

Q55: What do you mean by Stride?

Stride is the number of pixels which are shift over the input matrix. We move the filters to 1 pixel at a time when the stride is equal to 1.

Q56: What do you mean by Padding?

"Padding is an additional layer which can add to the border of an image." It is used to overcome the

Shrinking outputs Losing information on the corner of the image.

Q57: What is pooling layer.

Pooling layer plays a crucial role in pre-processing of an image. Pooling layer reduces the number of parameters when the images are too large. Pooling is "downscaling" of the image which is obtained from the previous layers.

Q58: What is Max Pooling?

Max pooling is a sample-based discrete process whose main objective is to reduce its dimensionality, downscale an input representation. And allow for the assumption to be made about features contained in the sub-region binned.

Q59: What is Average Pooling?

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and will compute the average values of each region.

Q60: What is Sum Pooling?

The sub-region for sum pooling or mean pooling will set the same as for max-pooling but instead of using the max function we use sum or mean.

Q61: What do you mean by fully connected layer?

The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.

Q62: What is the Softmax activation function?

The Softmax function is a wonderful activation function which turns numbers aka logits into probabilities which sum to one. Softmax function outputs a vector which represents the probability distributions of a list of potential outcomes.

Q63: How do you import PyTorch into Anaconda?

Here are the steps: 1. Download and install Anaconda (Go with the latest Python version). 2. Go to the Getting Started section on the PyTorch website. 3. Specify the appropriate configuration options for your particular environment. 4. Run the presented command in the terminal to install PyTorch.

Q64: Can PyTorch run on Windows?

Yes, PyTorch 0.4.0 supports Windows

Q65: What is Cuda in PyTorch?

torch.cuda is used to set up and run CUDA operations. It keeps track of the currently selected GPU, and all CUDA tensors you allocate will by default be created on that device.

Q66: What is the difference between Anaconda and Miniconda?

Anaconda is a set of about a hundred packages including conda, numpy, scipy, ipython notebook, and so on. Miniconda is a smaller alternative to Anaconda.

Q67: How do you check GPU usage?

Use the Windows key + R keyboard shortcut to open the Run command. 2. Type the following command to open DirectX Diagnostic Tool and press Enter: dxdiag.exe. 3. Click the Display tab. 4. On the right, under "Drivers," check the Driver Model information.

Q68: What is the difference between a Perceptron and Logistic Regression?

A Multi-Layer Perceptron (MLP) is one of the most basic neural networks that we use for classification. For a binary classification problem, we know that the output can be either 0 or 1. This is just like our simple logistic regression, where we use a logit function to generate a probability between 0 and 1.

So, what's the difference between the two?

Simply put, it is just the difference in the threshold function! When we restrict the logistic regression model to give us either exactly 1 or exactly 0, we get a Perceptron model

Q69: Can we have the same bias for all neurons of a hidden layer?

Essentially, you can have a different bias value at each layer or at each neuron as well.

However, it is best if we have a bias matrix for all the neurons in the hidden layers as well.

A point to note is that both these strategies would give you very different results.

Q70: What if we do not use any activation function(s) in a neural network?

The main aim of this question is to understand why we need activation functions in a neural network. You can start off by giving a simple explanation of how neural networks are built:

Step 1: Calculate the sum of all the inputs (X) according to their weights and include the bias term:

$$Z = (\text{weights} * X) + \text{bias}$$

Step 2: Apply an activation function to calculate the expected output:

$$Y = \text{Activation}(Z)$$

Steps 1 and 2 are performed at each layer. If you recollect, this is nothing but forward propagation! Now, what if there is no activation function?

Our equation for Y essentially becomes:

$$Y = Z = (\text{weights} * X) + \text{bias}$$

Wait – isn't this just a simple linear equation? Yes – and that is why we need activation functions. A linear equation will not be able to capture the complex patterns in the data – this is even more evident in the case of deep learning problems.

In order to capture non-linear relationships, we use activation functions, and that is why a neural network without an activation function is just a linear regression model.

Q70: In a neural network, what if all the weights are initialized with the same value?

In simplest terms, if all the neurons have the same value of weights, each hidden unit will get exactly the same signal. While this might work during forward propagation, the derivative of the cost function during backward propagation would be the same every time.

In short, there is no learning happening by the network! What do you call the phenomenon of the model being unable to learn any patterns from the data? Yes, underfitting.

Therefore, if all weights have the same initial value, this would lead to underfitting.

Note: This question might further lead to questions on exploding and vanishing gradients, which I have covered below.

Q71: List the supervised and unsupervised tasks in Deep Learning.

Now, this can be one tricky question. There might be a misconception that deep learning can only solve unsupervised learning problems. This is not the case. Some example of Supervised Learning and Deep learning include:

Image classification Text classification Sequence tagging

On the other hand, there are some unsupervised deep learning techniques as well

Word embeddings (like Skip-gram and Continuous Bag of Words) Autoencoders

Q72: What is the role of weights and bias in a neural network?

This is a question best explained with a real-life example. Consider that you want to go out today to play a cricket match with your friends. Now, a number of factors can affect your decision-making, like:

How many of your friends can make it to the game? How much equipment can all of you bring? What is the temperature outside? And so on. These factors can change your decision greatly or not too much. For example, if it is raining outside, then you cannot go out to play at all. Or if you have only one bat, you can share it while playing as well. The magnitude by which these factors can affect the game is called the weight of that factor.

Factors like the weather or temperature might have a higher weight, and other factors like equipment would have a lower weight.

However, does this mean that we can play a cricket match with only one bat? No – we would need 1 ball and 6 wickets as well. This is where bias comes into the picture. Bias lets you assign some threshold which helps you activate a decision-point (or a neuron) only when that threshold is crossed.

Q73: How does forward propagation and backpropagation work in deep learning?

Now, this can be answered in two ways. If you are on a phone interview, you cannot perform all the calculus in writing and show the interviewer. In such cases, it best to explain it as such:

Forward propagation: The inputs are provided with weights to the hidden layer. At each hidden layer, we calculate the output of the activation at each node and this further propagates to the next layer till the final output layer is reached. Since we start from the inputs to the final output layer, we move forward and it is called forward propagation

Backpropagation: We minimize the cost function by its understanding of how it changes with changing the weights and biases in a neural network. This change is obtained by calculating the gradient at each hidden layer (and using the chain rule). Since we start from the final cost function and go back each hidden layer, we move backward and thus it is called backward propagation

Q74: What are the common data structures used in Deep Learning?

List: An ordered sequence of elements (You can also mention NumPy ndarrays here) **Matrix:**

An ordered sequence of elements with rows and columns **Dataframe:** A dataframe is just like a matrix, but it holds actual data with the column names and rows denoting each datapoint in your dataset. If marks of 100 students, their grades, and their details are stored in a dataframe, their details are stored as columns. Each row will represent the data of each of the 100 students

Tensors: You will work with them on a daily basis if you have ventured into deep learning. Used both in PyTorch and TensorFlow, tensors are like the basic programming unit of deep learning. Just like multidimensional arrays, we can perform numerous mathematical operations on them. Read more about tensors here **Computation Graphs:** Since deep learning involves multiple layers and often hundreds, if not thousands of parameters, it is important to understand the flow of computation. A computation graph is just that. A computation graph gives us the sequence of operations performed with each node denoting an operation or a component in the neural network

Q75: Why should we use Batch Normalization?

Once the interviewer has asked you about the fundamentals of deep learning architectures, they would move on to the key topic of improving your deep learning model's performance. Batch Normalization is one of the techniques used for reducing the training time of our deep learning algorithm. Just like normalizing our input helps improve our logistic regression

model, we can normalize the activations of the hidden layers in our deep learning model as well

Q75: List the activation functions you have used so far in your projects and how you would choose one.

The most common activation functions are:

Sigmoid Tanh ReLU Softmax

Q76. Why does a Convolutional Neural Network (CNN) work better with image data?

The key to this question lies in the Convolution operation. Unlike humans, the machine sees the image as a matrix of pixel values. Instead of interpreting a shape like a petal or an ear, it just identifies curves and edges.

Thus, instead of looking at the entire image, it helps to just read the image in parts. Doing this for a 300 x 300 pixel image would mean dividing the matrix into smaller 3 x 3 matrices and dealing with them one by one. This is convolution.

Mathematically, we just perform a small operation on the matrix to help us detect features in the image – like boundaries, colors, etc.

$$Z = X * f$$

Here, we are convolving (* operation – not multiplication) the input matrix X with another small matrix f, called the kernel/filter to create a new matrix Z. This matrix is then passed on to the other layers.

Q77: Why do RNNs work better with text data?

The main component that differentiates Recurrent Neural Networks (RNN) from the other models is the addition of a loop at each node. This loop brings the recurrence mechanism in RNNs. In a basic Artificial Neural Network (ANN), each input is given the same weight and fed to the network at the same time. So, for a sentence like “I saw the movie and hated it”, it would be difficult to capture the information which associates “it” with the “movie”.

The addition of a loop is to denote preserving the previous node’s information for the next node, and so on. This is why RNNs are much better for sequential data, and since text data also is sequential in nature, they are an improvement over ANNs.

Q78: In a CNN, if the input size 5 X 5 and the filter size is 7 X 7, then what would be the size of the output?

This is a pretty intuitive answer. As we saw above, we perform the convolution on ‘x’ one step at a time, to the right, and in the end, we got Z with dimensions 2 X 2, for X with dimensions 3 X 3.

Thus, to make the input size similar to the filter size, we make use of padding – adding 0s to the input matrix such that its new size becomes at least 7 X 7. Thus, the output size would be using the formula:

Dimension of image = (n, n) = 5 X 5

Dimension of filter = (f,f) = 7 X 7

Padding = 1 (adding 1 pixel with value 0 all around the edges)

Dimension of output will be (n+2p-f+1) X (n+2p-f+1) = 1 X 1

Q79: What’s the difference between valid and same padding in a CNN?

This question has more chances of being a follow-up question to the previous one. Or if you have explained how you used CNNs in a computer vision task, the interviewer might ask this question along with the details of the padding parameters.

Valid Padding: When we do not use any padding. The resultant matrix after convolution will have dimensions (n – f + 1) X (n – f + 1) Same padding: Adding padded elements all around the edges such that the output matrix will have the same dimensions as that of the input matrix

Q80: What do you mean by exploding and vanishing gradients?

The key here is to make the explanation as simple as possible. As we know, the gradient descent algorithm tries to minimize the error by taking small steps towards the minimum value. These steps are used to update the weights and biases in a neural network.

However, at times, the steps become too large and this results in larger updates to weights and bias terms – so much so as to cause an overflow (or a NaN) value in the weights. This leads to an unstable algorithm and is called an exploding gradient.

On the other hand, the steps are too small and this leads to minimal changes in the weights and bias terms – even negligible changes at times. We thus might end up training a deep learning model with almost the same weights and biases each time and never reach the minimum error function. This is called the vanishing gradient.

A point to note is that both these issues are specifically evident in Recurrent Neural Networks – so be prepared for follow-up questions on RNN!

Q81: What are the applications of transfer learning in Deep Learning?

I am sure you would have a doubt as to why a relatively simple question was included in the Intermediate Level. The reason is the sheer volume of subsequent questions it can generate! The use of transfer learning has been one of the key milestones in deep learning. Training a large model on a huge dataset, and then using the final parameters on smaller simpler datasets has led to defining breakthroughs in the form of Pretrained Models. Be it Computer Vision or NLP, pretrained models have become the norm in research and in the industry.

Some popular examples include BERT, ResNet, GPT-2, VGG-16, etc and many more.

It is here that you can earn brownie points by pointing out specific examples/projects where you used these models and how you used them as well.

Q82: How backpropagation is different in RNN compared to ANN?

In Recurrent Neural Networks, we have an additional loop at each node. This loop essentially includes a time component into the network as well. This helps in capturing sequential information from the data, which could not be possible in a generic artificial neural network. This is why the backpropagation in RNN is called Backpropagation through Time, as in backpropagation at each time step.

Q83: How does LSTM solve the vanishing gradient challenge?

The LSTM model is considered a special case of RNNs. The problems of vanishing gradients and exploding gradients we saw earlier are a disadvantage while using the plain RNN model. In LSTMs, we add a forget gate, which is basically a memory unit that retains information that is retained across timesteps and discards the other information that is not needed. This also necessitates the need for input and output gates to include the results of the forget gate as well.

84: Why is GRU faster as compared to LSTM?

As you can see, the LSTM model can become quite complex. In order to still retain the functionality of retaining information across time and yet not make a too complex model, we need GRUs.

Basically, in GRUs, instead of having an additional Forget gate, we combine the input and Forget gates into a single Update Gate. It is this reduction in the number of gates that makes GRU less complex and faster than LSTM.

Q85: How is the transformer architecture better than RNN?

Advancements in deep learning have made it possible to solve many tasks in Natural Language Processing. Networks/Sequence models like RNNs, LSTMs, etc. are specifically used for this purpose – so as to capture all possible information from a given sentence, or a paragraph. However, sequential processing comes with its caveats:

It requires high processing power It is difficult to execute in parallel because of its sequential nature This gave rise to the Transformer architecture. Transformers use what is called the attention mechanism. This basically means mapping dependencies between all the parts of a sentence.