**1.What is the difference between TRAINABLE and NON-TRAINABLE PARAMETERS?**

**Trainable parameters are those which value is adjusted/modified during training as per their gradient. In Batch Normalization layer we have below mentioned trainable params: gamma: It's a scaling factor beta: a learned offset factor Non trainable parameters are those which value is not optimized during the training as per their gradient.**

**in keras, non-trainable parameters means the number of weights that are not updated during training with backpropagation.**

**There are mainly two types of non-trainable weights:**

* **The ones that you have chosen to keep constant when training. This means that keras won't update these weights during training at all.**
* **The ones that work like statistics in Batch Normalization layers. They're updated with mean and variance, but they're not "trained with backpropagation".**

**2.In the CNN architecture, where does the DROPOUT LAYER go?**

**We can apply a Dropout layer to the input vector, in which case it nullifies some of its features; but we can also apply it to a hidden layer, in which case it nullifies some hidden neurons. Dropout layers are important in training CNNs because they prevent overfitting on the training data.**

**Dropout layers are important in training CNNs because they prevent overfitting on the training data. If they aren’t present, the first batch of training samples influences the learning in a disproportionately high manner. This, in turn, would prevent the learning of features that appear only in later samples or batches**

**3. What is the optimal number of hidden layers to stack?**

**There are many rule-of-thumb methods for determining an acceptable number of neurons to use in the hidden layers, such as the following:**

1. **The number of hidden neurons should be between the size of the input layer and the size of the output layer.**
2. **The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.**
3. **The number of hidden neurons should be less than twice the size of the input layer.**

**the number of hidden layers shouldn't be too high! Because of the gradient descent when the number of layers is too large,**

1. **In each layer, how many secret units or filters should there be?**

**More than 0 and less than the number of parameters in each filter. For instance, if you have a 5x5 filter, 1 color channel (so, 5x5x1), then you should have less than 25 filters in that layer. The reason being is that if you have 25 or more filters, you have at least 1 filter per pixel. The filter bank should provide some lossy compression of the input, and if there are as many filters as parameters per filter, then it doesn't lose any data, it just massively overfits.**

**The size of the hidden layer is normally between the size of the input and output-.It should be should be 2/3 the size of the input layer plus the size of the o/p layer The number of hidden neurons should be less than twice the size of the input layer.**

1. **What should your initial learning rate be?**

**A traditional default value for the learning rate is 0.1 or 0.01, and this may represent a good starting point on your problem. A default value of 0.01 typically works for standard multi-layer neural networks but it would be foolish to rely exclusively on this default value**

**The learning rate (or step-size) is explained as the magnitude of change/update to model weights during the backpropagation training process. As a configurable hyperparameter, the learning rate is usually specified as a positive value less than 1.0. In back-propagation, model weights are updated to reduce the error estimates of our loss function.**

**6. What do you do with the activation function?**

**The activation function decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.**

**An activation function is a function that is added to an artificial neural network in order to help the network learn complex patterns in the data.**

**7.What is NORMALIZATION OF DATA?**

**Data normalization is generally considered the development of clean data. Diving deeper, however, the meaning or goal of data normalization is twofold: Data normalization is the organization of data to appear similar across all records and fields.**

**8. What is IMAGE AUGMENTATION and how does it work?**

* **Image augmentation is an engineered solution to create a new set of images by applying standard image processing methods to existing images. This solution is mostly useful for neural networks or CNN when the training dataset size is small.**
* **Instead of spending days manually collecting data, we can make use of Image Augmentation techniques. Image Augmentation is the process of generating new images for training our deep learning model. These new images are generated using the existing training images and hence we don’t have to collect them manually.**
* **Instead of spending days manually collecting data, we can make use of Image Augmentation techniques. Image Augmentation is the process of generating new images for training our deep learning model. These new images are generated using the existing training images and hence we don’t have to collect them manually.**

**9. What is DECLINE IN LEARNING RATE?**

* **In the decaying learning rate approach, it decreases with increase in epochs/iterations. The formula used is shown below: In the above equation, o is the initial learning rate, is the decay rate and is the learning rate at a given Epoch number.**
* **In both the methods, the learning rate decreases irrespective of difficulty involved in minimizing the cost function. In this approach, the learning rate increases or decreases based on the gradient value of the cost function.**

**10.What does EARLY STOPPING CRITERIA mean?**

* **In machine learning, early stopping is a form of regularization used to avoid overfitting when training a learner with an iterative method, such as gradient descent. Such methods update the learner so as to make it better fit the training data with each iteration.**

**These early stopping rules work by splitting the original training set into a new training set and a validation set. The error on the validation set is used as a proxy for the generalization error in determining when overfitting has begun. These methods are most commonly employed in the training of neural networks.**