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

A1. In a neural network, parameters are variables that are learned during the training process to optimize the model's performance. The difference between trainable and non-trainable parameters is whether they are adjusted during training or not.

Trainable parameters are the variables that the optimizer adjusts to minimize the loss function during training. These parameters include the weights and biases of each layer of the neural network. The optimizer updates these parameters at each iteration of the training process, using backpropagation to calculate the gradients of the loss function with respect to each weight and bias.

Non-trainable parameters are the variables that do not change during training. These parameters are usually set beforehand or fixed and do not affect the loss function. Examples of non-trainable parameters include hyperparameters such as learning rate, batch size, and number of epochs, as well as the parameters of some layers that do not require training, such as pooling and activation layers.

In summary, trainable parameters are adjusted during training to optimize the model's performance, while non-trainable parameters are set beforehand or fixed and do not change during training.

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

A2. The Dropout layer can be added after one or more fully connected layers in a CNN architecture. It is typically used to prevent overfitting by randomly dropping out (setting to zero) some of the output values of the layer during training. By doing this, the network is forced to learn more robust features and prevents it from relying too heavily on a specific set of neurons or features.

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

A3. There is no universally optimal number of hidden layers to stack in a neural network as it depends on the specific problem being solved, the size and complexity of the dataset, and the computational resources available. Generally, it is recommended to start with a small number of hidden layers and gradually increase the number until the desired performance is achieved, monitoring for signs of overfitting. It's also important to note that deeper networks can be more difficult to train due to issues such as vanishing gradients, and require more advanced techniques such as skip connections, residual networks, and layer normalization.

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

A4. The number of neurons or filters in each layer of a neural network is a hyperparameter that needs to be tuned for each specific problem. There is no one-size-fits-all answer to this question. The optimal number of units or filters can depend on various factors such as the complexity of the problem, the amount and quality of the training data, the network architecture, the regularization techniques used, etc. In practice, a common strategy is to start with a small number of units or filters in the first layers and gradually increase the number of units or filters in deeper layers. This approach allows the network to learn simple features in the early layers and more complex features in the deeper layers. However, the exact number of units or filters to use at each layer is typically determined through experimentation and tuning of hyperparameters.

5. What should your initial learning rate be?

A5.   
The choice of initial learning rate depends on several factors, including the problem being solved, the model architecture, and the optimizer being used. Generally, a good starting point for the learning rate is to use a value between 0.1 and 0.001, but the optimal learning rate may need to be determined through experimentation and validation. It is often useful to use a learning rate schedule, such as reducing the learning rate by a factor of 10 when the validation loss plateaus, to fine-tune the learning rate during training.

6. What do you do with the activation function?

A6. The activation function is a crucial component in neural networks as it introduces non-linearity into the model, allowing it to learn complex relationships between input and output. Choosing the right activation function can have a significant impact on the performance of the model. Typically, the ReLU (Rectified Linear Unit) activation function is used in most layers of a neural network, as it is computationally efficient and has been found to work well in many applications. However, depending on the problem domain and the nature of the data, other activation functions such as sigmoid, tanh, or softmax may be more appropriate. It is important to experiment with different activation functions to determine the optimal one for a particular problem.

7. What is NORMALIZATION OF DATA?

A7. Normalization of data is the process of scaling numerical data to a standard range to facilitate better model training and improve model performance. This involves transforming the values of the features in a dataset to a similar scale, typically between 0 and 1 or -1 and 1.

Normalization helps to mitigate issues such as vanishing or exploding gradients in neural networks, which can occur when the input features have very different scales. By bringing all features to a similar scale, normalization can improve the convergence of the optimization algorithm and reduce the training time required for a model.

Different techniques for normalization include min-max scaling, z-score normalization, and log transformation, among others. The choice of normalization technique depends on the type of data and the specific requirements of the model being trained.

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

A8. Image augmentation is a technique used in deep learning to artificially increase the size of a training dataset by applying random transformations to the existing images. This technique is particularly useful when the available training data is limited.

Image augmentation involves applying a variety of operations to the images, such as rotation, scaling, shearing, flipping, and cropping. These operations are performed randomly on each image in the training dataset, creating new variations of the original images.

For example, if we have an image of a cat, we can apply various transformations to generate new images such as flipping the image horizontally, cropping a part of the image, changing the brightness or contrast, or adding random noise to the image. By applying these transformations, we create more images that can be used for training our neural network, leading to a more robust and generalized model.

Image augmentation can be implemented in various deep learning libraries, such as TensorFlow and PyTorch, by using built-in functions or custom code. It is a powerful tool for improving the accuracy of deep learning models, especially when the dataset is small.

9. What is DECLINE IN LEARNING RATE?

A9. A decline in learning rate refers to the process of reducing the learning rate during the training process of a neural network. The learning rate determines the step size of the updates made to the model parameters during training. If the learning rate is too high, the optimization algorithm may overshoot the optimal solution and lead to divergence. On the other hand, if the learning rate is too low, the optimization process may take longer to converge to the optimal solution.

To overcome this issue, the learning rate is often decreased over time during the training process. This can be done through different techniques, such as step decay, where the learning rate is decreased by a certain factor after a fixed number of epochs, or adaptive learning rate methods such as Adam, where the learning rate is dynamically adjusted based on the past gradients. The goal is to balance the speed of convergence and the stability of the optimization process.

What does EARLY STOPPING CRITERIA mean?

A10.   
Early stopping criteria is a technique used in machine learning to avoid overfitting and improve the generalization ability of the model. It involves monitoring the performance of the model on a validation set during training and stopping the training process when the performance on the validation set no longer improves.

The criteria for early stopping can vary depending on the problem and the dataset, but typically involve monitoring a metric such as validation loss or accuracy. The training process is halted when the performance on the validation set stops improving or starts to worsen.

By using early stopping, the model is prevented from overfitting to the training data and is instead encouraged to generalize better to new, unseen data.