# CV Assignment 6

**Difference between TRAINABLE and NON-TRAINABLE PARAMETERS:**

- Trainable parameters are the parameters of a model that are updated during the training process to minimize the loss function. These parameters include weights and biases in the layers of the neural network.
- Non-trainable parameters are parameters that are not updated during training and remain fixed. These parameters are often part of pre-trained models or layers that are frozen to prevent further updates. Examples include parameters in pre-trained embeddings or fixed convolutional filters in transfer learning.

**Placement of DROPOUT LAYER in CNN Architecture:**

- The dropout layer in a CNN architecture is typically added after one or more convolutional layers and before the fully connected layers (or dense layers) of the network.
- By applying dropout after convolutional layers, the dropout layer helps regularize the model by preventing overfitting and improving generalization performance.
- The dropout rate is often set higher in the fully connected layers compared to convolutional layers to account for the larger number of parameters and prevent overfitting.

**Optimal Number of Hidden Layers to Stack:**

- The optimal number of hidden layers in a neural network depends on various factors, including the complexity of the task, the size of the dataset, and the availability of computational resources.
- While there is no one-size-fits-all answer, empirical evidence suggests that adding more layers can improve the representational capacity of the model and potentially enhance performance.
- However, adding too many layers can lead to overfitting, increased training time, and computational overhead. Therefore, it's essential to balance model complexity with generalization performance through experimentation and validation.

**Number of Secret Units or Filters in Each Layer:**

- The number of units or filters in each layer of a neural network is a hyperparameter that needs to be tuned based on the specific task and dataset.

- The number of filters in convolutional layers determines the richness of features extracted from the input data. It is often determined based on the complexity of the input data and the desired level of abstraction.
- The number of units in fully connected layers affects the capacity of the model to learn complex patterns and representations. It is typically chosen based on the complexity of the task and the size of the dataset, with larger layers providing more expressive power but also increasing the risk of overfitting.

**Initial Learning Rate:**

- The initial learning rate is a hyperparameter that determines the step size of parameter updates during training.
- Choosing an appropriate initial learning rate is crucial for achieving stable training and convergence to an optimal solution.
- The initial learning rate is often chosen based on heuristics, such as starting with a small value (e.g., 0.001) and adjusting it based on the performance of the model on a validation set.
- Techniques like learning rate schedules, learning rate warm-up, and adaptive learning rate methods can also be used to dynamically adjust the learning rate during training to improve convergence and stability.

**Activation Function:**

- The activation function introduces non-linearity into the network, allowing it to learn complex relationships and mappings between input and output.
- Common activation functions include ReLU (Rectified Linear Unit), sigmoid, tanh, and softmax.
- The choice of activation function depends on the task and network architecture. ReLU is commonly used in hidden layers due to its simplicity and effectiveness in training deep networks, while sigmoid and softmax are often used in output layers for binary or multi-class classification tasks.
- It's important to choose an activation function that avoids issues like vanishing gradients and dead neurons and promotes faster convergence during training.

**Normalization of Data:**

- Normalization of data is a preprocessing step that scales the input features to a similar range to facilitate training and improve convergence.
- Common normalization techniques include min-max scaling, z-score normalization, and batch normalization.

- Normalizing the input data helps prevent issues like vanishing or exploding gradients, improves the stability of training, and speeds up convergence.
- For image data, normalization often involves scaling pixel values to the range [0, 1] or [-1, 1] to ensure numerical stability and improve the performance of the model.

**Image Augmentation:**

- Image augmentation is a data augmentation technique commonly used in computer vision tasks to artificially increase the diversity of the training dataset by applying transformations to the input images.
- Common image augmentation techniques include random rotations, translations, flips, zooms, and changes in brightness and contrast.
- Image augmentation helps the model generalize better to unseen data, improves robustness to variations in lighting, orientation, and perspective, and reduces overfitting.
- During training, random variations are applied to the input images, creating new samples that are similar but not identical to the original images.

**Decline in Learning Rate:**

- The decline in learning rate, also known as learning rate decay or scheduling, is a technique used to gradually reduce the learning rate during training to improve convergence and prevent oscillations.
- Learning rate decay can be applied using various strategies, such as step decay, exponential decay, polynomial decay, or cosine annealing.
- Declining the learning rate over time helps the model navigate the loss landscape more effectively, allowing it to fine-tune the parameters and converge to a better solution.
- Learning rate decay is often combined with other techniques like learning rate warm-up and adaptive learning rate methods to improve training stability and convergence speed.