Problem 1: CNN for Image Classification

Using your preferred machine learning library, train a small convolutional network (CNN) to classify images from the *CIFAR10* dataset. Note that most libraries have utility functions to download and load this dataset (TensorFlow, PyTorch, keras).

Using the API for loading the dataset will readily divide it into training and testing sets. Randomly sample 20% of the training set and use that as your new training set for the purposes of this problem. Use the test set for validation. Implement the following in ***one*** *.ipynb* with all the output shown (already run).

1- **MLP:** Build a multi-layer perceptron with the following layers:
   - Fully connected layer with 512 units and a sigmoid activation function
   - Fully connected layer with 512 units and a sigmoid activation function
   - Output layer with the suitable activation function and number of neurons for the classification task

2- **CNN1:** Build a Convolutional neural network with the following architecture:
   - 2D Convolutional layer with 64 filters (size of 3x3) and ReLU activation function
   - 2D Convolutional layer with 64 filters (size of 3x3) and ReLU activation function
   - Fully connected (Dense) layer with 512 units and a sigmoid activation function
   - Fully connected layer with 512 units and a sigmoid activation function
   - Output layer with the suitable activation function and number of neurons for the classification task

3- **CNN2:** Build a Convolutional Neural network with the following architecture:
   - 2D Convolutional layer with 64 filters (size of 3x3) and ReLU activation function
   - 2x2 Max pooling layer
   - 2D Convolutional layer with 64 filters (size of 3x3) and ReLU activation function
   - 2x2 Max pooling layer
   - Fully connected layer with 512 units and a sigmoid activation function
   - Dropout layer with 0.2 dropout rate
   - Fully connected layer with 512 units and a sigmoid activation function
   - Dropout layer with 0.2 dropout rate
   - Output layer with the suitable activation function and number of neurons for the classification task

Use a batch size of 32, utilize Adam as the optimizer and choose an appropriate loss function while monitoring the accuracy in both networks. Train each network for 5 epochs.

What you will submit

a) Well commented code and a description of what each part does.
b) Report:
    - Any preprocessing steps you made
    - Description of the output layer used and the loss function (use 1 setting for all 3 networks) and why you made these choices.
    - Change the number of layers and the number of neurons per layer in the MLP, plot/tabulate the training and validation accuracies and comment on the results.
    - Train and test accuracy for all three networks and comment (what happens and why) on the performance of the MLP vs CNNs.
    - Plot the training and validation curves for the two CNNs and comment on the output. How does the training time compare for each of the CNNs? How does the different architectures influence these results? What do you expect the accuracies to be if the networks were trained for more epochs?
    - Recommendations to improve the network. What changes would you make to the architecture and why?
c) Do not upload the dataset but make sure your code is able to download it if we need to run your code.