Class Assignment 3-1

Convolutional Neural Networks on CIFAR Dataset

Link to dataset: [CIFAR-10 and CIFAR-100 datasets (toronto.edu)]

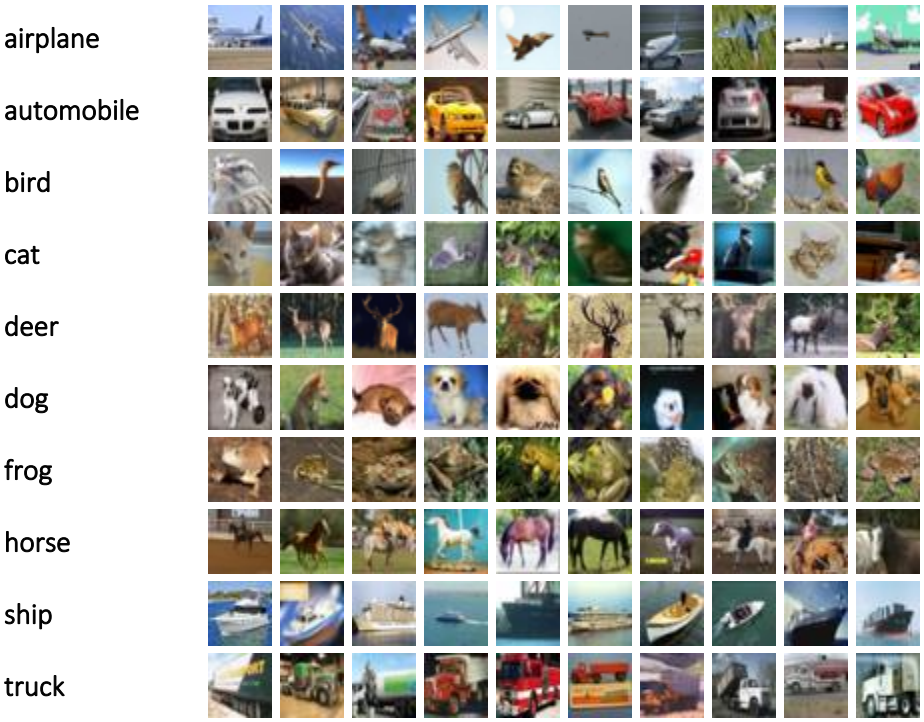
In this exercise we will train a simple Convolutional Neural Network (CNN) to classify CIFAR Images. This exercise will be an example of utilizing Keras Sequential API in creating and training a model.

About dataset:

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

Here are the classes in the dataset, as well as 10 random images from each:



How to import data:

```
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()

# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0
```

Steps to Follow:

1. Very the data to confirm the data is loaded correctly and plot the first 25 images
2. Create a base sequential convolutional model with following architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 max_pooling2d (MaxPooling2D  (None, 15, 15, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 4, 4, 64)          36928
```

3. Add a dense layer to classify based on available classes
4. Compile and train the model
5. Evaluate model by training and validation accuracy
6. Discuss ways to improve the model