Problem Set 2
Image Classification
Applying CNNs For Image Classification

In this problem set we want to use PyTorch to train convolutional neural network (CNN) based image classifier. You will be using CIFAR-10 dataset to train and evaluate your model. You should try different values for the CNN hyper-parameters and study the effect of each of them on the model performance.

1 CIFAR-10 Dataset

The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6000 images per class. The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks. You can download the dataset from here.

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

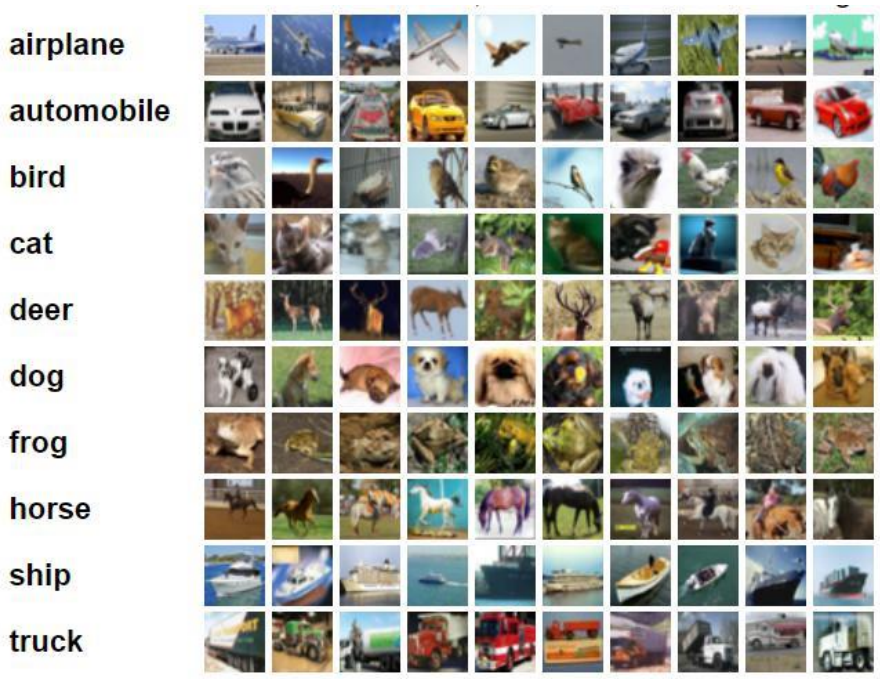

Figure 1: Sample Images from CIFAR-10 dataset.

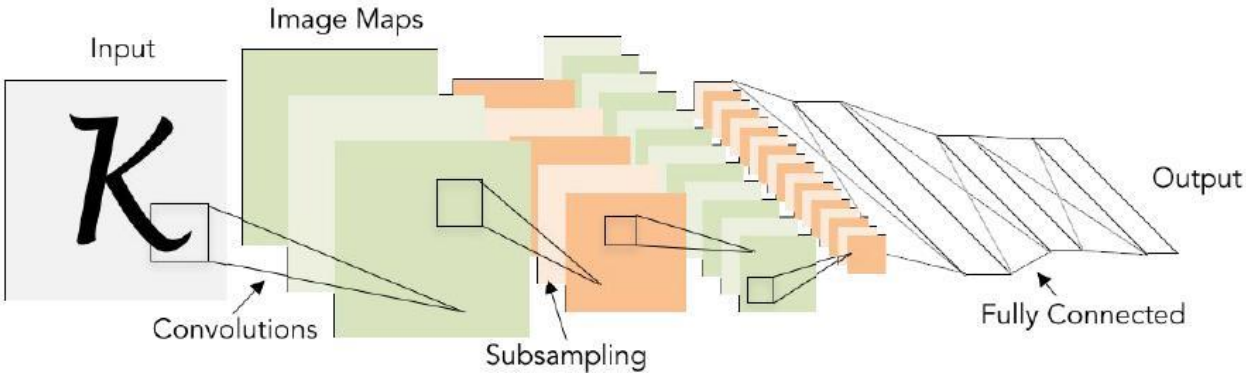

Figure 2: The general structure of convolutional neural network applied for character recognition.

2 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) are a special kind of neural networks for processing data that has a known grid-like topology such as time-series data and image data. The general structure of a CNN consists of one or more convolutional layers followed by one or more fully connected layers as in a standard multi-layer neural network as illustrated in Figure 2. This constrained architecture allows the CNN to leverage the spatial and temporal structure of the input and allows the network to learn more complex features from different parts of the input.

Typically, a convolutional layer consists of three stages:

1. In the first stage, the layer applies a kernel to its input by performing several convolutions in parallel to produce a set of linear activations.

2. In the second stage, each linear activation is run through a nonlinear activation function, such as the recti ed linear activation function.

3. In the third stage, a pooling function is used to further modify the output of the layer. Pooling functions replace the output of the net at a certain location with a summary statistic of the nearby outputs. For example, the max pooling operation reports the maximum output within a rectangular neighborhood. Other popular pooling functions include the average of a rectangular neighborhood or a weighted average based on the distance from the central pixel.

3 Requirements

You are required to train different CNN classifiers on CIFAR-10 dataset.

You should change the size of the kernels used and the number of lters and study the effect of their values on the model performance (accuracy and training time). Possible sizes for the kernel may include: 1 × 1, 3 × 3, ..., 11 × 11, ..., etc.

You should train the inception module you studied in lectures, train its both versions and compare their performance.

Hint: You can refer to Keras implementation of common CNN-based classifiers to guide you in building the models architecture. But you are not allowed to use any pre-trained models. Transfer Learning is not allowed.

4 Deliverables

You are required to deliver the following:

Your code.

Output for some test images.

Report including explanation of your code and evaluation results of the different models you trained.