Assignment#3

In this assignment you will practice putting together a simple image classification pipeline, based on the k-Nearest Neighbor or the SVM/Softmax classifier. The goals of this assignment are as follows:

- understand the basic **Image Classification pipeline** and the data-driven approach (train/predict stages)
- understand the train/val/test **splits** and the use of validation data for **hyperparameter tuning**.
- develop proficiency in writing efficient **vectorized** code with numpy
- implement and apply a k-Nearest Neighbor (**kNN**) classifier
- implement and apply a Multiclass Support Vector Machine (**SVM**) classifier
- implement and apply a **Softmax** classifier
- implement and apply a **Two layer neural network** classifier
- understand the differences and tradeoffs between these classifiers
- get a basic understanding of performance improvements from using **higher-level representations** than raw pixels (e.g. color histograms, Histogram of Gradient (HOG) features).

You Need to Download data:

Once you have the starter code (regardless of which method you choose above), you will need to download the CIFAR-10 dataset.

Start IPython:

After you have the CIFAR-10 data, you should start the IPython notebook server from the **assignment3** directory, with the **jupyter notebook** command. (See the **Google Cloud Tutorial** for any additional steps you may need to do for setting this up, if you are working remotely)

If you are unfamiliar with IPython, you can also refer to our **IPython tutorial.**

NOTE 1: The assignment#3 code has been tested to be compatible with python version **3.7** (it may work with other versions of **3.x**, but we won't be officially supporting them). You will need to make sure that during your virtual environment setup that the correct version of **python** is used. You can confirm your python version by (1) activating your virtualenv and (2) running **which python**.

Feel free to pick any others ways to implement this assignment#3 !!!!

Q1: k-Nearest Neighbor classifier

The IPython Notebook **knn.ipynb** will walk you through implementing the kNN classifier.

Q2: Training a Support Vector Machine

The IPython Notebook **svm.ipynb** will walk you through implementing the SVM classifier.

Q3: Implement a Softmax classifier

The IPython Notebook **softmax.ipynb** will walk you through implementing the Softmax classifier.

Q4: Two-Layer Neural Network

The IPython Notebook **two_layer_net.ipynb** will walk you through the implementation of a two-layer neural network classifier.

Q5: Higher Level Representations: Image Features

The IPython Notebook **features.ipynb** will walk you through this exercise, in which you will examine the improvements gained by using higher-level representations as opposed to using raw pixel values.

Submitting your work

Important: *Please make sure that the submitted notebooks have been run and the cell outputs are visible.*