Note: this is the 2017 version of this assignment.

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)

Setup

You can work on the assignment in one of two ways: locally on your own machine, or on a virtual machine on Google Cloud.

Working remotely on Google Cloud (Recommended)

Note: after following these instructions, make sure you go to **Download data** below (you can skip the **Working locally** section).
As part of this course, you can use Google Cloud for your assignments. We recommend this route for anyone who is having trouble with installation set-up, or if you would like to use better CPU/GPU resources than you may have locally. Please see the set-up tutorial here for more details. :)

Working locally

Get the code as a zip file here. As for the dependencies:
Installing Python 3.5+: To use python3, make sure to install version 3.5 or 3.6 on your local machine. If you are on Mac OS X, you can do this using Homebrew with `brew install python3`. You can find instructions for Ubuntu here.

Virtual environment: If you decide to work locally, we recommend using virtual environment for the project. If you choose not to use a virtual environment, it is up to you to make sure that all dependencies for the code are installed globally on your machine. To set up a virtual environment, run the following:

```
cd assignment1
sudo pip install virtualenv      # This may already be installed
virtualenv -p python3 .env       # Create a virtual environment
(python3)
# Note: you can also use "virtualenv .env" to use your default
python (usually python 2.7)
source .env/bin/activate         # Activate the virtual environment
pip install -r requirements.txt  # Install dependencies
# Work on the assignment for a while ...
deactivate                       # Exit the virtual environment
```

Note that every time you want to work on the assignment, you should run `source .env/bin/activate` (from within your `assignment1` folder) to re-activate the virtual environment, and `deactivate` again whenever you are done.

Download data:

Once you have the starter code (regardless of which method you choose above), you will need to download the CIFAR-10 dataset. Run the following from the `assignment1` directory:

```
cd cs231n/datasets
./get_datasets.sh
```

Start IPython:

After you have the CIFAR-10 data, you should start the IPython notebook server from the `assignment1` 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.

Some Notes

NOTE 1: This year, the `assignment1` code has been tested to be compatible with python versions `2.7`, `3.5`, `3.6` (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 `virtualenv` 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`.

NOTE 2: If you are working in a virtual environment on OSX, you may *potentially* encounter errors with matplotlib due to the issues described here. In our testing, it seems that this issue is no longer present with the most recent version of matplotlib, but if you do end up running into this issue you may have to use the `start_ipython_osx.sh` script from the `assignment1` directory (instead of `jupyter notebook` above) to launch your IPython notebook server. Note that you may have to modify some variables within the script to match your version of python/installation directory. The script assumes that your virtual environment is named `.env`.

Submitting your work:

Whether you work on the assignment locally or using Google Cloud, once you are done working run the `collectSubmission.sh` script; this will produce a file called `assignment1.zip`. Please submit this file on Canvas.

Q1: k-Nearest Neighbor classifier (20 points)

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

Q2: Training a Support Vector Machine (25 points)

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

Q3: Implement a Softmax classifier (20 points)

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

Q4: Two-Layer Neural Network (25 points)

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 (10 points)

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.

Q6: Cool Bonus: Do something extra! (+10 points)

Implement, investigate or analyze something extra surrounding the topics in this assignment, and using the code you developed. For example, is there some other interesting question we could have asked? Is there any insightful visualization you can plot? Or anything fun to look at? Or maybe you can experiment with a spin on the loss function? If you try out something cool we'll give you up to 10 extra points and may feature your results in the lecture.