CSCI 4830/5722 Computer Vision, Spring 2018
Instructor: Fleming
Homework 5 – part II

Q1 – mandatory for UG students, due Sunday, May 6th, by 11:55pm

Q1 + Q2 – mandatory for graduate students, due Wednesday, May 9th, by 11:55pm

Q2 – extra-credit for UG students, due Wednesday, May 9th, by 11:55pm
Q3 – extra-credit for graduate students, due Wednesday, May 9th, by 11:55pm

Part II: Training and Testing

In the second part of this assignment you will train and test a simple image classification pipeline based on the SVM classifier and a simple two-layer neural network classifier.

Provided files:

- Starter code is provided as a zip file on Moodle. Your file tree opens in the *hmwk5* folder
- Download the CIFAR-10 data set for Python here: https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz and save the archive in the *hmwk5_2/datasets* folder.

Working locally

Installing Anaconda: To work locally, we recommend using the free Anaconda Python distribution, which provides an easy way for you to handle package dependencies. Please be sure to download the Python 3 version, which currently installs Python 3.6.

Anaconda Virtual environment: Once you have Anaconda installed, it makes sense to create a virtual environment for this assignment. 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 (in a terminal)

```
conda create -n hmwk5 python=3.6 anaconda
```

to create an environment called `hmwk5`. You can of course choose a different name. Then, to activate and enter the environment, run

```
source activate hmwk5
```

To exit, you can simply close the window, or run

```
source deactivate hmwk5
```

Note that every time you want to work on the assignment, you should run `source activate hmwk5` (change to the name of your virtual env).

You may refer to this page for more detailed instructions on managing virtual environments with Anaconda.

Python virtualenv: Alternatively, you may use python virtualenv for the project. To set up a virtual environment, run the following:

```
cd hmwk5
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 (please note we support 3.6)
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
```

Working remotely on Google Cloud (not supported by our staff)

Another option is to use Google Cloud for this assignment. Google Cloud is free to try for one year. You will definitely have better CPU/GPU resources than you may have locally. Please see the set-up tutorial from Stanford's cs231n here for more details.

Download data:

Once you have the starter code (regardless of which method you choose above), you will need to download the CIFAR-10 dataset. Go to https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz and save the archive in the "datasets" folder. Run the following from the `hmwk5` directory:

```
cd hmwk5_2/datasets
./get_datasets.sh
```

Start your virtual environment:

After you setup your virtual environment, just remember to run

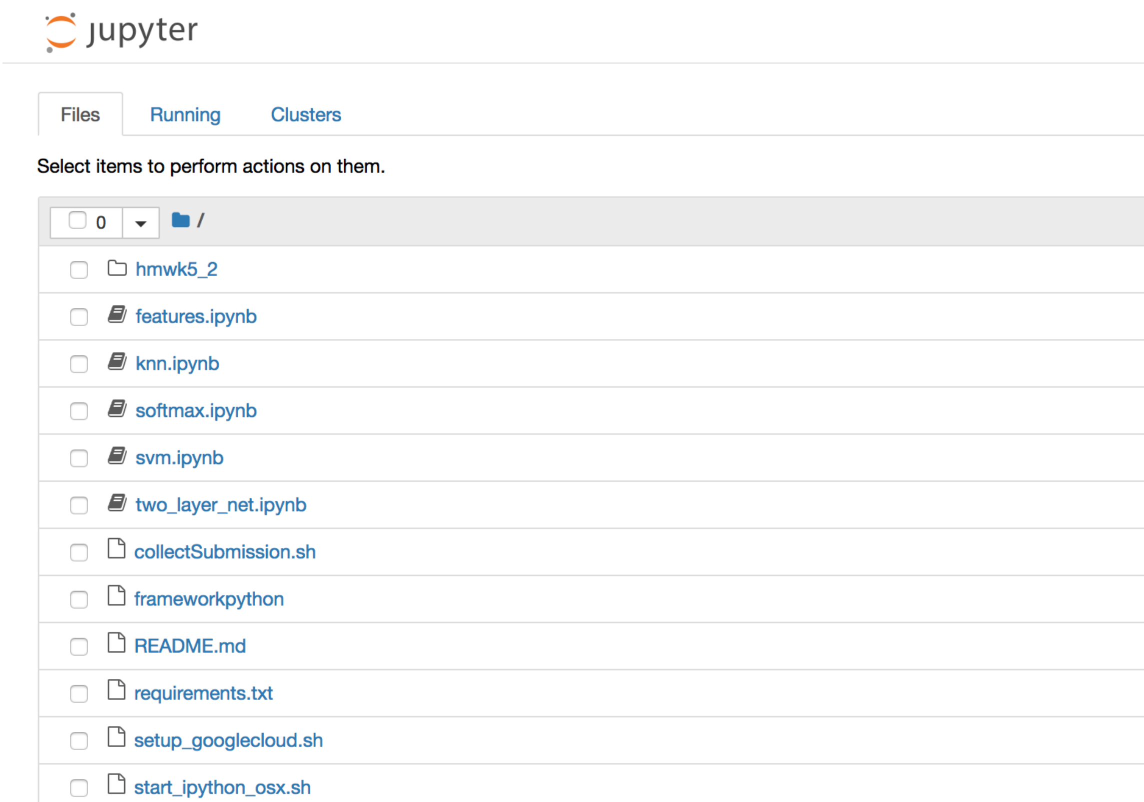
`source activate hmwk5`

Start IPython:

After you have the CIFAR-10 data, you should start the IPython notebook server from the `hmwk5` directory, with this command:

```
jupyter notebook
```

(See the [Google Cloud Tutorial]() for any additional steps you may need to do for setting this up, if you are working remotely).
Once your notebook server is running, point your web browser at http://localhost:8888 to start using your notebooks. If everything worked correctly, you should see a screen like this, showing all available IPython notebooks in the current directory:



Click through one of the notebook files (.ipynb) to work on different parts of the assignment. Open *knn.ipynb* first; the solution is provided.

What You Have to Do:

Q0: k-Nearest Neighbor classifier
The IPython Notebook **knn.ipynb** will walk you through implementing the kNN classifier. The solution for knn is provided. It will help you familiarize yourself with

the pipeline, the initial setup part, splitting the training and the testing data, calling specific class files, doing cross-validation at the end.

Q1: Training a Support Vector Machine (50 points) – **mandatory for all students**
The IPython Notebook **svm.ipynb** will walk you through implementing the SVM classifier. Fill in the missing code parts and answer the questions which require a written answer. At some points, you will need to fill in missing parts in the class files: *hmwk5_2/classifiers/linear_classifier.py* and *hmwk5_2/classifiers/linear_svm.py*.

Q2: Two-Layer Neural Network (100 points) – **mandatory for graduate students and extra-credit for undergraduate students**
The IPython Notebook **two_layer_net.ipynb** will walk you through the implementation of a two-layer neural network classifier. Fill in the missing code parts and answer the questions which require a written answer. At some points, you will need to fill in missing parts in the class files: *hmwk5_2/classifiers/neural_net.py.*

Q3: Implement a Softmax classifier (50 points) - **extra-credit for graduate students**
The IPython Notebook **softmax.ipynb** will walk you through implementing the Softmax classifier. Fill in the missing code parts and answer the questions which require a written answer. At some points, you will need to fill in missing parts in the class files: *hmwk5_2/classifiers/softmax.py.*

Submitting the assignment:
Save each IPython notebook and each class file as a pdf file. Zip your entire file tree (the **hmwk5** folder), plus the pdf files and submit the archive through Moodle, as Hmwk5-partII before the deadline specified at the top.