In this assignment you will practice writing backpropagation code, and training Neural Networks and Convolutional Neural Networks. The goals of this assignment are as follows:

- understand **Neural Networks** and how they are arranged in layered architectures
- understand and be able to implement (vectorized) **backpropagation**
- implement various **update rules** used to optimize Neural Networks
- implement **Batch Normalization** and **Layer Normalization** for training deep networks
- implement **Dropout** to regularize networks
- understand the architecture of **Convolutional Neural Networks** and get practice with training these models on data
- gain experience with a major deep learning framework, such as **TensorFlow** or **PyTorch**.

Setup

Get the code as a zip file here.

You can follow the setup instructions here.

NOTE: Our initial release of the assignment did not include the PyTorch and TensorFlow notebooks for Q5. These have now been finalized, and the zip file has been updated with these notebooks.

Download data:

Once you have the starter code, you will need to download the CIFAR-10 dataset. Run the following from the `assignment2` 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 `assignment2` 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 `assignment2` code has been tested to be compatible with python version `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 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`.

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 `assignment2` 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`.

Q1: Fully-connected Neural Network (20 points)

The IPython notebook `FullyConnectedNets.ipynb` will introduce you to our modular layer design, and then use those layers to implement fully-connected networks of arbitrary depth. To optimize these models you will implement several popular update rules.

Q2: Batch Normalization (30 points)

In the IPython notebook `BatchNormalization.ipynb` you will implement batch normalization, and use it to train deep fully-connected networks.

Q3: Dropout (10 points)

The IPython notebook `Dropout.ipynb` will help you implement Dropout and explore its effects on model generalization.

Q4: Convolutional Networks (30 points)

In the IPython Notebook `ConvolutionalNetworks.ipynb` you will implement several new layers that are commonly used in convolutional networks.

Q5: PyTorch / TensorFlow on CIFAR-10 (10 points)

For this last part, you will be working in either TensorFlow or PyTorch, two popular and powerful deep learning frameworks. **You only need to complete ONE of these two notebooks.** You do NOT need to do both, and we will *not* be awarding extra credit to those who do.

Open up either `PyTorch.ipynb` or `TensorFlow.ipynb`. There, you will learn how the framework works, culminating in training a convolutional network of your own design on CIFAR-10 to get the best performance you can.

NOTE: The PyTorch notebook requires PyTorch version 0.4, which was released on 4/24/2018. You can install this version of PyTorch using conda or pip by following the instructions here: http://pytorch.org/

Submitting your work

There are ***two*** steps to submitting your assignment:

1. Submit a pdf of the completed iPython notebooks to Gradescope. If you are enrolled in the course, then you should have already been automatically added to the course on Gradescope.

To produce a pdf of your work, you can first convert each of the .ipynb files to HTML. To do this, simply run from your assignment directory

```
jupyter nbconvert --to html FILE.ipynb
```

for each of the notebooks, where `FILE.ipynb` is the notebook you want to convert. Then you can convert the HTML files to PDFs with your favorite web browser, and then concatenate them all together in your favorite PDF viewer/editor. Submit this final PDF on Gradescope, and be sure to tag the questions correctly!

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

2. Submit a zip file of your assignment on AFS. To do this, run the provided `collectSubmission.sh` script, which will produce a file called `assignment2.zip`. You will then need to SCP this file over to Stanford AFS using the following command (entering your Stanford password if requested):

```
# Run from the assignment directory where the zip file is located
scp assignment2.zip YOUR_SUNET@myth.stanford.edu:~/DEST_PATH
```

`YOUR_SUNET` should be replaced with your SUNetID (e.g. `jdoe`), and `DEST_PATH` should be a path to an existing directory on AFS where you want the zip file to be copied to (you may want to create a CS231N directory for convenience). Once this is done, run the following:

```
# SSH into the Stanford Myth machines
ssh YOUR_SUNET@myth.stanford.edu

# Descend into the directory where the zip file is now located
cd DEST_PATH

# Run the script to actually submit the assignment
/afs/ir/class/cs231n/submit
```

Once you run the submit script, simply follow the on-screen prompts to finish submitting the assignment on AFS. If successful, you should see a "SUBMIT SUCCESS" message output by the script.

---

cs231n
cs231n
karpathy@cs.stanford.edu