CSC 492/592: Special Topics (Deep Learning)
University of Rhode Island
Homework Assignment #2 (Due on Nov 17th at 11:59p)

This assignment is entirely based on one of the assignments from Stanford's CS231n (`http://cs231n.stanford.edu/`). You will implement vectorized backpropagation, update rules, batch normalization, dropout, and also train Neural Networks and Convolutional Neural Networks.

The assignment is individual and you can work in one of two ways: locally on your own machine, or on a virtual machine on Google Cloud. Google Cloud is recommended for anyone that would like to use better GPU resources (will only be useful on the last question). A Google Cloud set-up tutorial and credits for using GPU instances will be released on Piazza.

To start working on this assignment you will need Python 3 (Python 2 is not supported). You can download the starter code from:

```
http://cs231n.stanford.edu/assignments/2017/spring1617_assignment2.zip
```

After downloading the archive, unzip it and enter the `assignment2` directory. You will now remove (erase) the following files:

```
README.md
frameworkpython
start_ipython_osx.sh
```

Before starting to work on your assignment ensure that you have installed all requirements listed in the following file:

```
requirements.txt
```

You will now download the CIFAR-10 dataset, using the following commands from the `assignment2` directory:

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

After downloading CIFAR-10, you will start the IPython notebook server with the `jupyter notebook` command. If you are unfamiliar with IPython, you can also refer to this IPython tutorial:

```
https://github.com/cs231n/cs231n.github.io/blob/master/ipython-tutorial
```

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 `assignment2.zip`. Please submit this file on Gradescope.

Questions

Q1 **Fully-connected Neural Network (20 pt)** 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 (20 pt)** In the IPython notebook `BatchNormalization.ipynb` you will implement batch normalization, and use it to train deep fully-connected networks.

Q3 **Dropout (10 pt)** The IPython notebook `Dropout.ipynb` will help you implement Dropout and explore its effects on model generalization.

Q4 **Convolutional Networks (25 pt)** 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 (25 pt)** For this last part, you will be working in either TensorFlow or PyTorch. You only need to complete ONE of these two notebooks. 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. You should feel free to implement anything that you want to get better performance. A Kaggle competition will be open for CIFAR-10, where you can submit your predictions. More information about this submission will be posted on Piazza.