Assignment 2

This assignment is due on **Wednesday, May 6 2020** at 11:59pm PDT.

▶ Handy Download Links

Goals

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

You can work on the assignment in one of two ways: **remotely** on Google Colaboratory or **locally** on your own machine.

Regardless of the method chosen, ensure you have followed the setup instructions before proceeding.

Option A: Google Colaboratory (Recommended)

Download. Starter code containing Colab notebooks can be downloaded here.

If you choose to work with Google Colab, please familiarize yourself with the recommended workflow.

Note 1. Please make sure that you work on the Colab notebooks in the order of the questions (see below). The reason is that the code cells that get executed *at the end* of the notebooks save the modified files back to your drive and some notebooks may require code from previous notebook.

Note 2. Related to above, ensure you are periodically saving your notebook (`File -> Save`), and any edited `.py` files relevant to that notebook (i.e. **by executing the last code cell**) so that you don't lose your progress if you step away from the assignment and the Colab VM disconnects.

Once you have completed all Colab notebooks **except** `collect_submission.ipynb`, proceed to the submission instructions.

Option B: Local Development

Download. Starter code containing jupyter notebooks can be downloaded here.

Install Packages. Once you have the starter code, activate your environment (the one you installed in the Software Setup page) and run `pip install -r requirements.txt`.

Download CIFAR-10. Next, you will need to download the CIFAR-10 dataset. Run the following from the `assignment2` directory:

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

Start Jupyter Server. After you have the CIFAR-10 data, you should start the Jupyter server from the `assignment2` directory by executing `jupyter notebook` in your terminal.

Complete each notebook, then once you are done, go to the submission instructions.

Q1: Fully-connected Neural Network (20 points)

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

Q3: Dropout (10 points)

The 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.

Submitting your work

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

Once you have completed all notebooks and filled out the necessary code, there are ***two*** steps you must follow to submit your assignment:

1. If you selected Option A and worked on the assignment in Colab, open `collect_submission.ipynb` in Colab and execute the notebook cells. If you selected Option B and worked on the assignment locally, run the bash script in `assignment2` by executing `bash collectSubmission.sh`.

This notebook/script will:

- Generate a zip file of your code (`.py` and `.ipynb`) called `a2.zip`.
- Convert all notebooks into a single PDF file.

Note for Option B users. You must have (a) `nbconvert` installed with Pandoc and Tex support and (b) `PyPDF2` installed to successfully convert your notebooks to a PDF file. Please follow these installation instructions to install (a) and run `pip install PyPDF2` to install (b). If you are, for some inexplicable reason, unable to successfully install the above dependencies, you can manually convert each jupyter notebook to HTML (`File -> Download as -> HTML (.html)`), save the HTML page as a PDF, then concatenate all the PDFs into a single PDF submission using your favorite PDF viewer.

If your submission for this step was successful, you should see the following display message:

```
### Done! Please submit a2.zip and the pdfs to Gradescope. ###
```

2. Submit the PDF and the zip file to Gradescope.

Note for Option A users. Remember to download `a2.zip` and `assignment.pdf` locally before submitting to Gradescope.

cs231n
cs231n