Assignment 3

In this assignment, you will implement Fully-Connected Neural Networks and Convolutional Neural Networks for image classification models. 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 modular backpropagation
- Implement various update rules used to optimize Neural Networks
- Implement Batch 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

This assignment is due on **Friday, October 9th at 11:59pm EDT**.

Q1: Fully-Connected Neural Network (40 points)

The notebook **fully_connected_networks.ipynb** will walk you through implementing Fully-Connected Neural Networks.

Q2: Convolutional Neural Network (60 points)

The notebook **convolutional_networks.ipynb** will walk you through implementing Convolutional Neural Networks.

Steps

1. Download the zipped assignment file

- Click here to download the starter code

2. Unzip all and open the Colab file from the Drive

Once you unzip the downloaded content, please upload the folder to your Google Drive. Then, open each `*.ipynb` notebook file with Google Colab by right-clicking the *.ipynb file. We recommend editing your `*.py` file on Google Colab, set the ipython notebook and the code side by side. For more information on using Colab, please see our Colab tutorial.

3. Work on the assignment

Work through the notebook, executing cells and writing code in `*.py,` as indicated. You can save your work, both `*.ipynb` and `*.py,` in Google Drive (click "File" -> "Save") and resume later if you don't want to complete it all at once.

While working on the assignment, keep the following in mind:

- The notebook and the python file have clearly marked blocks where you are expected to write code. **Do not write or modify any code outside of these blocks**.
- **Do not add or delete cells from the notebook**. You may add new cells to perform scratch computations, but you should delete them before submitting your work.
- **Run all cells, and do not clear out the outputs, before submitting.** You will only get credit for code that has been run.

4. Evaluate your implementation on Autograder

4. Evaluate your implementation on Autograder

Once you want to evaluate your implementation, please submit the `*.py` and `*.ipynb` files to Autograder for grading your implementations in the middle or after implementing everything. You can

partially grade some of the files in the middle, but please make sure that this also reduces the daily submission quota. Please check our Autograder tutorial for details.

5. Download .zip file

Once you have completed a notebook, download the completed `uniqueid_umid_A3.zip` file, which is generated from your last cell of the `convolutional_networks.ipynb` file. Before executing the last cell in `convolutional_networks.ipynb`, please manually save your results so that the zip file includes all updates.

Make sure your downloaded zip file includes your most up-to-date edits; the zip file should include *fully_connected_networks.ipynb*, *convolutional_networks.ipynb*, *fully_connected_networks.py*, *convolutional_networks.py* , *best_overfit_five_layer_net.pth*, *best_two_layer_net.pth*, *one_minute_deepconvnet.pth*, *overfit_deepconvnet.pth* for this assignment.

6. Submit your python and ipython notebook files to Autograder

When you are done, please upload your work to Autograder (UMich enrolled students only). Your `*.ipynb` files *SHOULD include* all the outputs. Please check your outputs up to date before submitting yours to Autograder.
