CS 639: Deep Learning for Computer Vision, Spring 2023
Problem Set 1

Instructor: Yong Jae Lee (yongjaelee@cs.wisc.edu)
TA: Utkarsh Ojha (uojha@wisc.edu)
TA: Mu Cai (mucai@cs.wisc.edu)

Due: Thursday, February 23rd, 11:59 PM

Instructions

1. Download the zipped assignment file from Canvas.

2. 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. No installation or setup is required. For more information, please see this tutorial on using Colab.

3. Next, we recommend editing your `*.py` file on Google Colab, set the ipython notebook and the code side by side. Work through the notebook, executing cells and implementing the codes in the `*.py` file as indicated. You can save your work, both `*.ipynb` and `*.py`, in Google Drive (click "File" -> "Save" or press "Ctrl/Cmd + s") and resume later if you don't want to complete it all at once.

4. 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.

5. The assignment has three parts, where you will be implementing (i) k-nearest neighbors, (ii) linear (softmax) classifier and (iii) fully connected neural networks. Work on them in the mentioned order.

6. Once you have completed all the parts, download the completed `uniqueid_PS1.zip` file, which is generated from your last cell of the `fully_connected_networks.ipynb` file. Submit this to Canvas. Note that only one person from the group will need to do this.

7. You may complete the assignment individually or with a partner (i.e., maximum group of 2 people). If you worked with a partner, provide the name of your partner in the `*.ipynb` file. **We will be using MOSS to check instances of plagiarism/cheating.**

The goal of this assignment is for you to learn how to use KNN classifier, linear layer, softmax function, and fully connected layers so that you can

- Develop proficiency with training and evaluation pipeline in machine learning.
- Gain experience using forward function and backward propagation.

The notebooks `knn.ipynb`, `linear_classifier.ipynb`, `fully_connected_networks.ipynb` will walk you through those pipelines in PyTorch. You are required to write code on `knn.py`, `linear_classifier.py` and `fully_connected_networks.py`.[1]

[OPTIONAL] Extra credit short answer problems [up to 0.5% each added to final class score]

Save your answers into a PDF, and submit it as a separate file on Canvas.

1. A deep neural network has multiple layers with non-linear activation functions (e.g., ReLU) in between each layer, which allows it to learn a complex non-linear function. Suppose instead we had a deep neural network without any non-linear activation functions. Concisely describe what effect this would have on the network. (Hint: can it still be considered a deep network?)

2. Why is the bias term needed in linear classifiers?

[1] The coding assignment is adapted from Stanford CS 231n.