CMPUT 328 Fall 2020

Assignment 2

Worth 10% of the total weight

Part 1: Logistic Regression [Worth 5% of the total weight]:

Implement Logistic Regression in PyTorch:

a) Train and test on both MNIST and CIFAR10 datasets.
   - Define your own data pipeline for training, validation and testing using PyTorch dataloader.
     - Use the last 12,000 samples of the train set as validation set for both datasets.
     - Test the trained model on the validation set every few epochs to prevent overfitting
     - **Do not use the test set for training**.
   - Take note of the differences between MNIST and CIFAR10 to make your code work for both. For example:
     - MNIST images are grayscale, while CIFAR10 images are RGB.
     - MNIST and CIFAR10 images have different spatial dimensions (height and width).

b) Add a regularization term to improve your model (L1 or L2 regularization, whichever gives better accuracy)

Expected Performance: A correctly implemented, and somewhat well-tuned version of this algorithm will have an accuracy of **92-94%** on MNIST and **38-40%** on CIFAR10 for both test and validation sets

Using existing code: Some of the functionality required for this part might be present lecture notebooks. You are advised to avoid any copy-pasting from there but if you do, make sure to document it in enough detail to make it absolutely clear to the marking TA that you understand what is going on.

This applies to copying from any source in general but this particular case is important since any instance of undocumented copying from lecture notebooks will **lead to heavy penalty (beyond the one third reserved for documentation) and might even lead to zero marks**.

Part 2: Hyperparameter Search [Worth 5% of the total weight]

Find optimal hyperparameters for Adaptive Moment Estimation (Adam) and Stochastic Gradient Descent (SGD) on CIFAR10 dataset including the regularization method you used in part 1.

- You should perform grid search or random search for finding the optimal hyper-parameters using accuracy on the validation set and select the best configuration.
- You can also use more advanced search strategies like evolutionary search, but you are **not** allowed to use any automatic parameter search methods like *scorch.*
- You should not use the test set during this process.
- You only need to complete the function ***tune_hyper_parameter*** in *A2_submission.py* for this part.

Template Code: You are provided with template code in the form of 3 files: *A2_submission.py*, *A2_part1.py*, and *A2_part2.py*. *A2_part1.py* and *A2_part2.py* are independent of each other but both use *A2_submission.py*.

You need to complete all the functions in *A2_submission.py*.

The code can be run using *python3 A2_part1.py* for part 1 and *python3 A2_part2.py* for part 2 on your own machine or *!python3 "<full path to A2_part1.py or A2_part2.py>"* from a code cell in Colab.

You can also create notebooks on Colab and copy all code from the template files to cells.

Submission: **You need to submit only the completed *A2_submission.py***. Make sure to import any additional libraries you need so it can be used as a standalone Python module from both *A2_part.py* and *A2_part2.py*

Marking: A third of the marks will be for documentation and remaining two-thirds will be for the code:

Part 1: Code marks will depend on correctness of the implementation and following metrics:

- **Runtime**: The total runtime of your submission (including training and testing) **should not exceed 300 seconds** for either dataset using GPU on Colab.
    - One trick to improve your run time is to grid search the hyperparameters first but only put in the best hyperparameters you found in your submission.
- **Accuracy**: Each dataset will give you 50% of the total score in this assignment.
    - **MNIST**: score scales linearly from **82 - 92%** on the test set
    - **CIFAR10**: score scales linearly from **28 - 38%** on the test set

Part 2: Code marks will depend only on the correctness of your search implementation.

- **Runtime**: The runtime of your submission **should not exceed 1500 seconds** using GPU on Colab.
- **Accuracy**: There are no specific accuracy requirements except there should be improvement in loss / accuracy compared to part 1.