Discussion on the Project of Comparing Various Stochastic Gradient Methods
Part 1

- For this part you should get exactly the same result
- With same initial weights and the same operations you should get the same weights in the first several iterations
- For example, here are the weights of running mnist with the following parameters (scripts and results were generated by our TAs)

```python
python3 script.py --optim SGD --bsize 256 --seed 42 --net CNN_4layers --train_set /tmp3/data/mnist.mat --val_set /tmp3/data/mnist.t.mat --dim 28 28 1
```
For simpleNN, first layer of running 11 batches are
batch 1: 0.14049198 -0.03910705 0.18319398 ...
batch 11: 1.36893839e-01 -2.44279262e-02 1.47583246e-01 ...

Results of using Tensorflow
batch 1: 0.14049198 -0.03910705 0.18319398 ...
batch 11: 1.36893839e-01 -2.44279262e-02 1.47583246e-01 ...

For those who did not get the same results, probably you did not check the Tensorflow manual in detail
Part 2

- You must think about how to clearly organize and present your results.
- For example, a table may be better than the following description:

  learning rate ?? gives final accuracy ??, best accuracy ??, learning rate ?? gives final accuracy ??, best accuracy ??, learning rate ?? gives final accuracy ??, best accuracy ??,

- You can see that “learning rate,” “final accuracy,” etc. appear many times.
If the method fails to converge and get bad accuracy, from our discussion, you may decrease the learning rate.

For example, some tried Adam with learning rates 0.01, 0.1, 0.5 on cifar10, and all failed.

In this situation you could try for example 0.005 or 0.001.
Please respect the page limit. We would like to see how you can summarize things in two pages.