CS 410/510: Deep Learning
Winter 2021
Assignment 1
Due Jan 20th (11:59pm)

Useful Links:

https://pytorch.org/tutorials/beginner/pytorch_with_examples.html

Implementation Notes: I suggest using Google colab for the assignment since it is much easier to import the necessary libraries. Here is a link to get started: https://colab.research.google.com/notebooks/intro.ipynb - scrollTo=GJBs_flRovLc In colab you will be implementing the assignments in pytorch. If you take code from the above pytorch tutorial and copy it into code cells in colab, it ought to work just fine So I urge you to explore and really exploit the cool features of colab.

Assignment Description:

The goal here is to implement a fully connected NN to classify images and basically learn how its performance is affected by choices of different parameters and data.

1. You will use FashionMNIST data for this project. The dataloader from the pytorch tutorial will load the image dataset for you.

```
# datasets
trainset = torchvision.datasets.FashionMNIST('./data',
    download=True,
    train=True,
    transform=transform)
testset = torchvision.datasets.FashionMNIST('./data',
    download=True,
    train=False,
    transform=transform)
# dataloaders
trainloader = torch.utils.data.DataLoader(trainset, batch_size=4,
                                        shuffle=True, num_workers=2)


testloader = torch.utils.data.DataLoader(testset, batch_size=4,
                                        shuffle=False, num_workers=2)
# constant for classes
classes = ('T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
        'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle Boot')
```

2. Next, you need to create a few different NNs and explore how the *accuracy* changes. Initially start with a single FC layer with 1024 neurons. The FashionMNIST images are 28x28 with a single channel. So your input is a vector of size 28*28 = 784 that feeds to the 1024 size layer which then connects to the output layer of size 10. Use learning rates of 0.001 and momentum of 0. Use ReLU and cross entropy loss. Use optim.SGD as the optimizer. Next, repeat with 2 fully-connected layers of 1024 each. For these experiments use a fixed batch size of 30.
    a. *Submit link(s) to your colab notebook(s) (In the submission label this as 1)*
    b. *Submit a one paragraph description of the difference in accuracy of the two networks and why (in the submission label this as 2)*
3. Next, using the second neural network, experiment with these variable settings:
    a. Mini-batch size of 1, 10, 1000
    b. Vary the learning rate as 1.0, 0.1, 0.01, 0.001

    c. Change the activation from ReLU to Sigmoid for both the FC (fully connected) layers
    - _Submit a table containing accuracy for the 3*4*2 = 24 cases. For the best and worst cases, write a one para description that explains the difference (in the submission label this 3)_

4. From 3 above, pick the best set of parameters for the 2 FC layer network. Pick 9 sets of 1% of images from each of the 10 categories and add them to the other 9 categories. This is called _pollution_. Run the best case on this data (training and then test on the unpolluted test data).
    a. _Report your results with an explanation (In the submission label this 4)_

You will submit this one single document on D2L. Be clear and if you like, add sample images and graphs. Anything to explain your results.

DO NOT use a CNN. This has to be a fully-connected network.