Deep Learning Systems
(ENGR-E 533)
Homework 1

Instructions

Due date: Sep. 24, 2023, 23:59 PM (Eastern)

- Start early if you're not familiar with the subject, TF or PT programming, and LaTeX.
- Do it yourself. Discussion is fine, but code up on your own
- Late policy
  - If the sum of the late hours (throughout the semester) $<$ seven days (168 hours): no penalty
  - If your total late hours is larger than 168 hours, you'll get only 80% of all the late-submitted homework.
- I ask you to use either PyTorch 1.X or Tensorflow 2.x running on Python 3.
- Submit a `.ipynb` as a consolidated version of your report and code snippets. But, the math should be clear with LaTeX symbols and the explanations should be full by using text cells. In addition, submit an `.html` version of your notebook as well, where you embed your sound clips and images. For example, if you have a graph as a result of your code cell, it should be visible in this `.html` version before we run your code. Ditto for the sound examples.

Problem 1: Softmax MNIST Classification [2 points]

1. Replicate the test accuracy graph on M02-S09.
2. Show me your weight visualization, too.
3. Please do not use any advanced optimization methods (Adam, batch norm, dropout, etc.) or initialization methods (Xavier and so on). Plan SGD should just work.
4. In TF 2.x, you can do something like this to download the MNIST dataset:

```
mnist = tf.keras.datasets.mnist
```

In PT, you can use these lines of commands (don't worry about the batch size and normalization– you can go for your own option for them):

```
import torchvision
mnist_train=torchvision.datasets.MNIST('mnist',
    train=True,
    download=True,
    transform=torchvision.transforms.Compose([
```

```
    torchvision.transforms.ToTensor(),
    torchvision.transforms.Normalize((0.1307,), (0.3081,))
    ]))
mnist_test=torchvision.datasets.MNIST('mnist',
    train=False,
    download=True,
    transform=torchvision.transforms.Compose([
    torchvision.transforms.ToTensor(),
    torchvision.transforms.Normalize((0.1307,), (0.3081,))
    ]))
```

Problem 2: Autoencoders [4 points]

1. Replicate the test accuracy graph on M02-S12.
2. It means, you also want to show the figures in M02-S11.
3. Note that your encoder weights are frozen; you only update the softmax layer weights (the 100×10 matrix and the bias).

Problem 3: A shallow NN [3 points]

1. Replicate the test accuracy graph on M02-S14.
2. I don't have to see the visualization of the first layer. Just show me your graphs.

Problem 4: Full BP on the both layers [6 points]

1. Replicate the test accuracy graph on M02-S17.

Replicate the figures in M03 Adult Optimization, slide 22 using the details as follows:

1. Use the same network architecture and train five different network instances in five different setups. The architecture has to be a fully connected network (a regular network, not a CNN or RNN) with five hidden layers, 512 hidden units per layer.
2. Create five different networks that share the same architecture as follows:
   (a) Activation function: the logistic sigmoid function; initialization: random numbers generated from the normal distribution ($\mu = 0$, $\sigma = 0.01$)
   (b) Activation function: the logistic sigmoid function; initialization: Xavier initializer
   (c) Activation function: ReLU; initialization: random numbers generated from the normal distribution ($\mu = 0$, $\sigma = 0.01$)
   (d) Activation function: ReLU; initialization: Xavier initializer
   (e) Activation function: ReLU; initialization: Kaiming He's initializer
3. You don't have to implement your own initializer. Both TF and PT come with pre-implemented initializers.

4. Train them with the traditional SGD. Do not improve SGD by introducing momentum or any other advanced stuff. Your goal is to replicate the figures in 22. Feel free to use pre-implemented SGD optimizer.

5. In practice, you will need to investigate different learning rate as for SGD, which will give you different convergence behaviors.

6. Don't worry if your graphs are slightly different from mine. We will give a full mark if your graphs show the same trend.