MLDS Homework 1 report

1. **Group Information**

Group name: 怕～了吧～～

Team member 1: B01901121 李律慈

Team member 2: B01901085 楊承翰

Member Contribution

* 楊承翰: Implements basic DNN structure and data preprocessing, normalizing the output.
* 李律慈: More effort in training, implement momentum and try different model.

1. **What we have done**

First we implement the basic DNN using the simple baseline specification without momentum. We also generate smaller training data set by random to reduce the file reading time. It is convenient in the beginning since reading the whole training data takes about 5 minutes, but the program may crash due to some bugs, and the time is wasted.

We had trouble using GPU to accelerate the program, so it took a few hours to generate a result that can pass the simple baseline. After passing the simple baseline, we tried some techniques such as momentum and different structure of neural network. They also give us better results. Besides, we have tried to implement relu as our activation function. However, it seems that there is some bug make cost cannot go down. Moreover, there are some bug in our adagrad, and we do not have enough to debug, so we only try to tune the parameter in momentum.

1. **Experiment results**
2. Line1 mu(learning rate) is 0.01, which is fixed, while the learning rate of line2 and line3 is , where i is the ith iteration.
3. According to above figure, we can observe that although line1 has smaller learning rate at the beginning, its cost is still larger than line2 and line3 after 100 iteration fixed learning.
4. Since eta of line3 is larger than line2, cost of line3 decrease faster than line2 as our expectation.
5. Except momentum, three lines have the same model.
6. We can observe that cost of line1 and line2(with momentum) decrease faster than line3.
7. Also, cost of line1 and line2 are smaller than line3 at the same iteration.
8. Model without momentum gets about 57.3% (public score). Model with momentum gets about 61.1%(public score).
9. Three models use momentum with different layers: 69-256-256-48, 69-256-256-256-48, 69-256-256-256-256-48 respectively.
10. We can observe that cost of 4 layers is a little less than 3 layers and 5 layers. We think that this phenomenon is caused by using sigmoid function as our activation function.