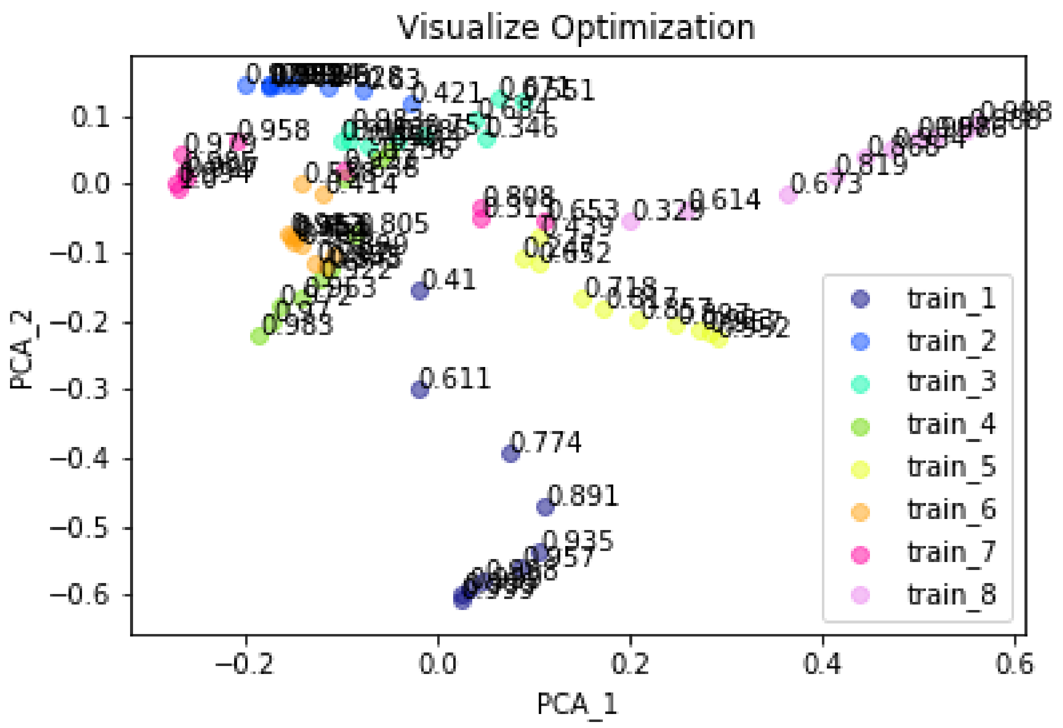
HW 1-2 What Happened When Gradient is Almost Zero

1. Visualize the optimization process

- **Describe the experiment settings:**

  I train the classification task on MNIST dataset with a simple DNN model, which epoch is 30. The training data size is 1000 and batch size is 200. It can let the model to fitting the training data (Overfitting) and easily visualize the optimization process. After training, I use PCA to project all the trainable variables into 2-dimension.

- **Train the model for 8 times:**

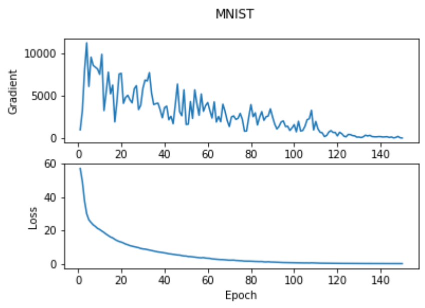


- **Comment the result:**

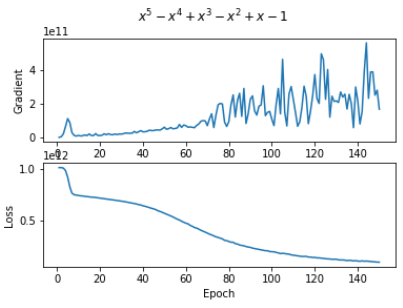
  The activation function is non-linear(Relu). As the result, I found that each time training would lead to difference optimization solution. However, the final performance is similar (98%-99%) for each training. Each of the solution is the same (i.e. same weights with different order) or totally different remains unknown.

2. Observe gradient norm during training

- **Plot one figure which contain gradient norm to iterations and the loss the iterations:**



MNIST
(parameters: 16750)



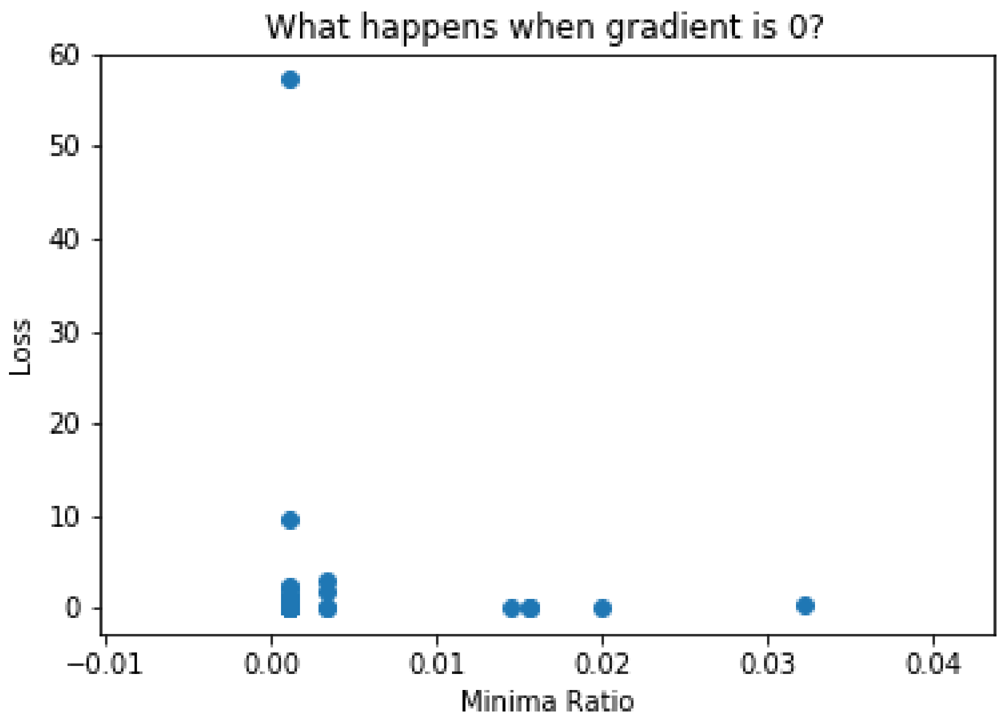
Simulate a function
(parameters:1090)

- **Comment the result:**

First of all, I found that when the gradient norm getting larger, the loss value decrease fast. Both of two tasks (MNIST and Simulate a function) shows the same.result. However, the MNIST dataset shows substantial changing but the Simulate a function task change slowly at the beginning. I think it is because that the MNIST task doesn’t have a flatten plan near original point in the loss surface, but the Simulate function task has. It is weird because that it would be difficult to train if the parameters are too many. However, our result shows that it could be easily training if the parameters getting larger.

3. What happens when gradient is almost zero

- **Plot one figure which contain gradient norm to iterations and the loss the iterations:**



- **Comment the result:**

I don’t think I did a great job in this task. The result shows that loss value and minima ratio don’t have a relationship. No matter the loss value getting larger or smaller, the minima ratio shows similar in a small value. Maybe the way I calculate minima ratio is wrong. Since Tensorflow cannot calculate Hessian matrix with multi-dimension, I modify each layer into flatten first, then reshape to the specific size. And I assume that the Hessian matrix is blocking diagonal, which means that each layer doesn’t have relationship. Then the minima ratio is calculated as the ratio of Eigen values which are larger then zero. This result is based on Simulative Function task. With fewer parameters, the Hessian matrix and SVD could be calculated quickly.