

---

# MLDS HW1-2

TAs  
ntu.mldsta@gmail.com

---

# Outline

- ❖ **Timeline**
- ❖ **Task Descriptions**
- ❖ **Q&A**

# Timeline

# Three Parts in HW1

- (1-1) Deep vs Shallow:
  - Simulate a function.
  - Train on actual task using shallow and deep models.
- (1-2) Optimization
  - Visualize the optimization process.
  - Observe gradient norm during training.
  - What happens when gradient is almost zero?
- (1-3) Generalization

# Schedule

- 3/9 :
  - Release HW1-1
- 3/16 :
  - Release HW1-2
- 3/23:
  - Deadline to team-up by yourselves
  - Release HW1-3
- 3/30:
  - Deadline to team-up by TAs
- 4/6:
  - All HW1 due (including HW1-1, HW1-2 and HW1-3)

# Task Descriptions

# HW1-2: Optimization

- Three subtask
  - Visualize the optimization process.
  - Observe gradient norm during training.
  - What happens when gradient is almost zero?
- Train on designed function, MNIST or CIFAR-10...

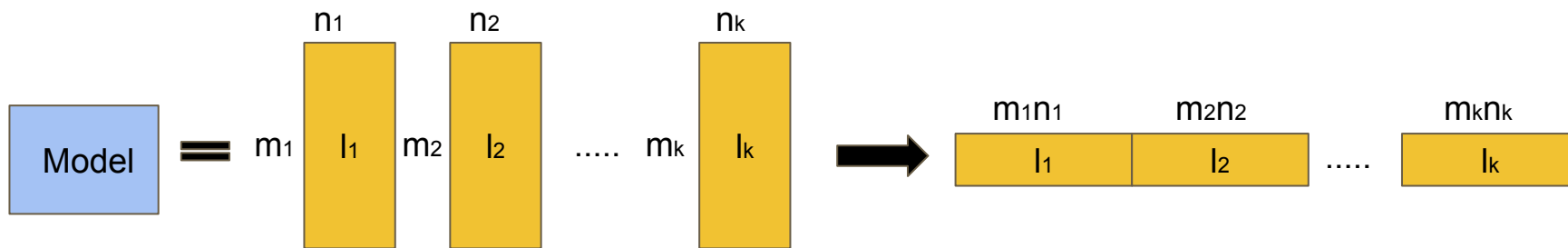
# Visualize the Optimization Process <sup>1/3</sup>

- Requirement
  - Collect weights of the model every  $n$  epochs.
  - Also collect the weights of the model of different training events.
  - Record the accuracy (loss) corresponding to the collected parameters.
  - Plot the above results on a figure.

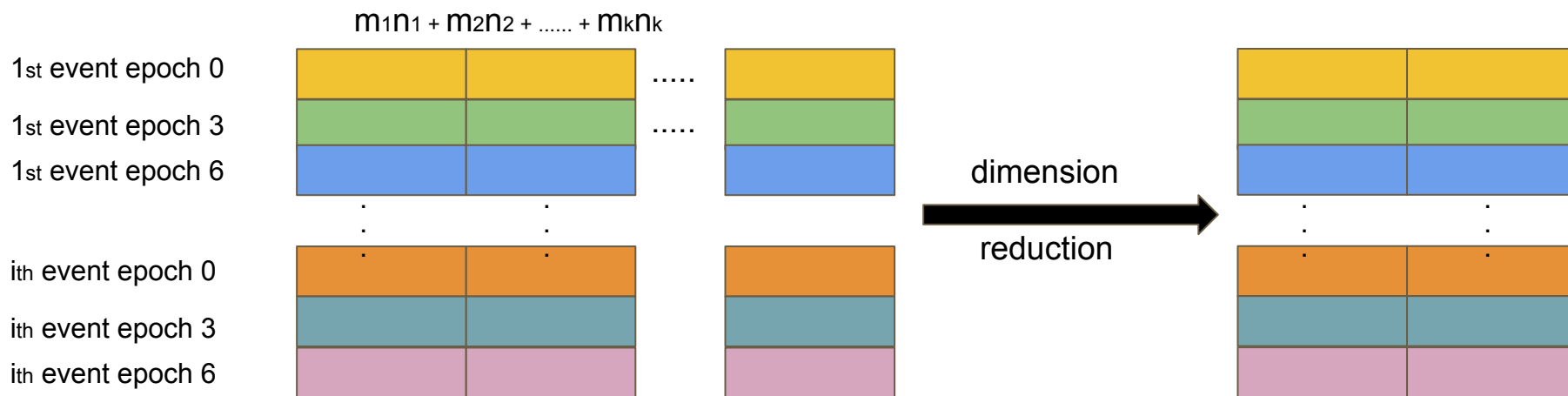


# Visualize the Optimization Process <sup>2/3</sup>

- Collect parameters of the model:

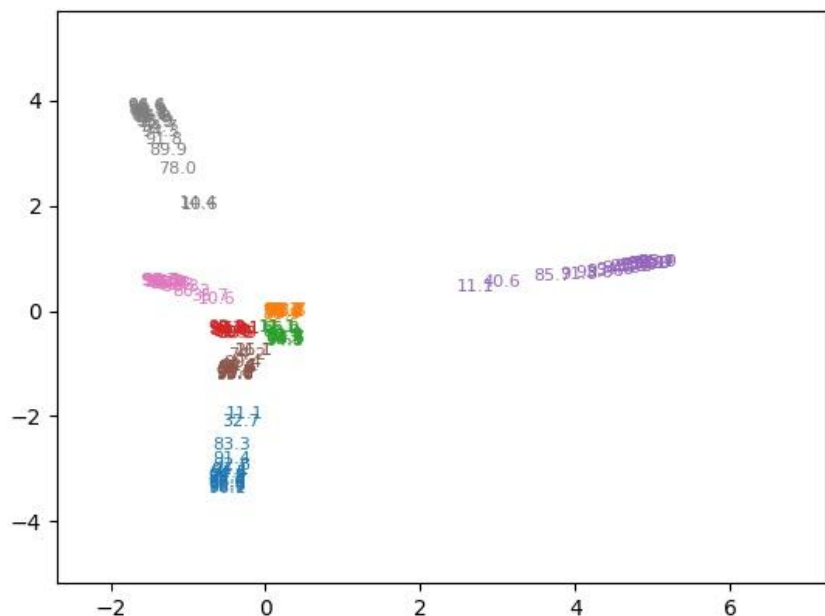


- Reduce the dimension

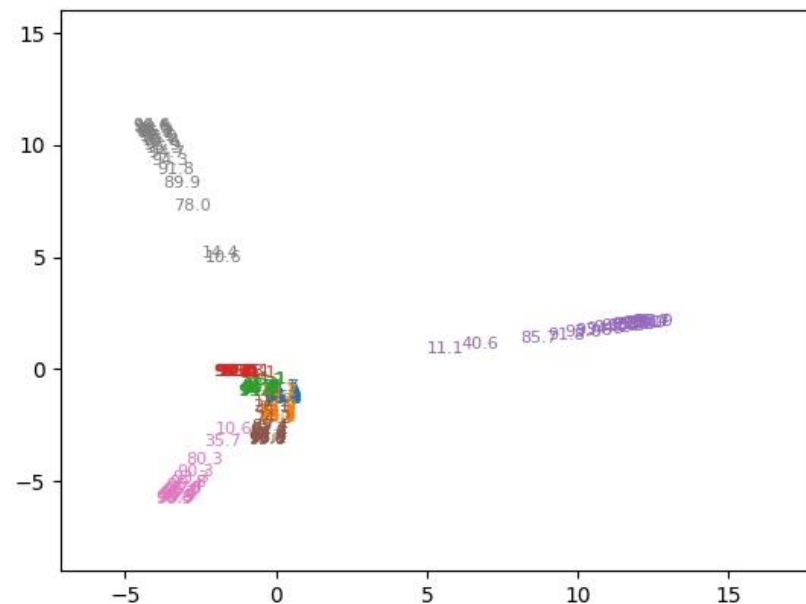


# Visualize the Optimization Process <sup>3/3</sup>

- DNN train on MNIST
- Collect the weights every 3 epochs, and train 8 times. Reduce the dimension of weights to 2 by PCA.



layer 1



whole model

# Observe Gradient Norm During Training <sup>1/2</sup>

- Requirement

- Record the gradient norm and the loss during training.
- Plot them on **one** figure.

- p-norm

- $\|\mathbf{x}\|_p := \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$
- In PyTorch:

```
grad_all = 0.0

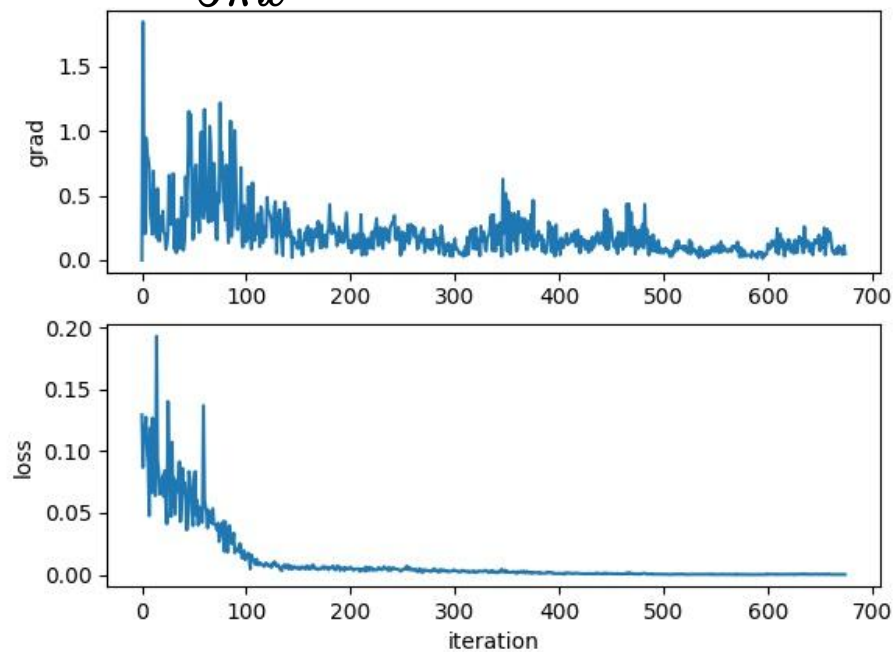
for p in model.parameters():
    grad = 0.0
    if p.grad is not None:
        grad = (p.grad.cpu().data.numpy() ** 2).sum()
        grad_all += grad

grad_norm = grad_all ** 0.5
```

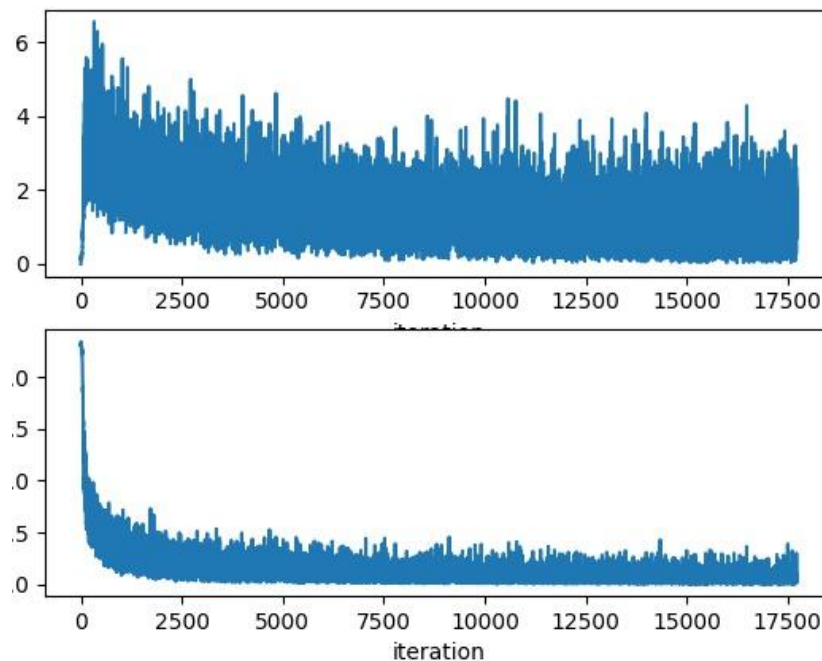
- Other packages: The similar code can be applied.

# Observe Gradient Norm During Training <sup>2/2</sup>

$$\frac{\sin 5\pi x}{5\pi x}$$



MNIST



# What Happened When Gradient is Almost Zero <sup>1/3</sup>

- Requirement
  - Try to find the weights of the model when the gradient norm is zero (as small as possible).
  - Compute the "minimal ratio" of the weights: how likely the weights to be a minima.
  - Plot the figure between minimal ratio and the loss when the gradient is almost zero.
- Tips
  - Train on a small network.

# What Happened When Gradient is Almost Zero <sup>2/3</sup>

## 1. How to reach the point where the gradient norm is zero?

First, train the network with original loss function.

- i. Change the objective function to gradient norm and keep training.
- ii. Or use second order optimization method, such as Newton's method or Levenberg-Marquardt algorithm (more stable)

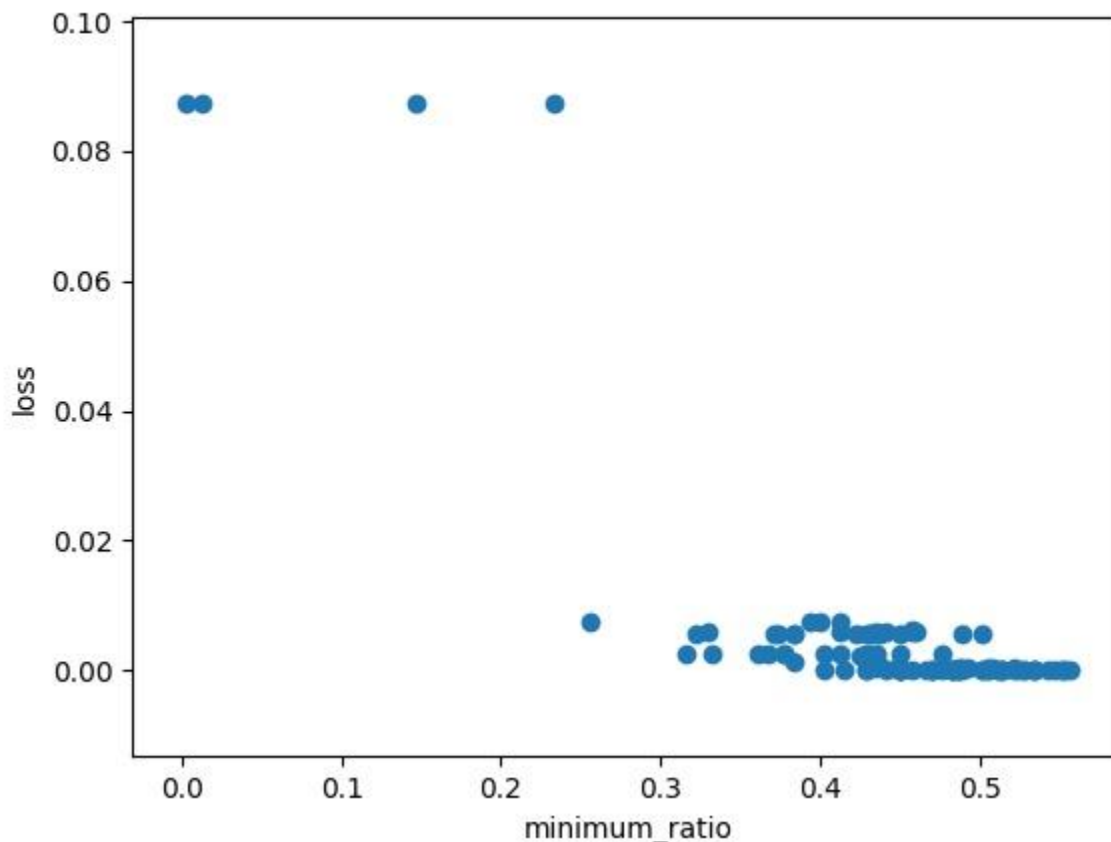
## 2. How to compute minimal ratio?

- i. Compute  $H(L(\theta_{norm=0}))$  (hessian matrix), and then find its eigenvalues. The proportion of the eigenvalues which are greater than zero is the minimal ratio.
- ii. Sample lots of weights around  $\theta_{norm=0}$ , and compute  $L(\theta_{sample})$ . The minimal ratio is the proportion that  $L(\theta_{sample}) > L(\theta_{norm=0})$

# What Happened When Gradient is Almost Zero <sup>3/3</sup>

- $\frac{\sin 5\pi x}{5\pi x}$

- Train 100 times.
- Find gradient norm equal to zero by change objective function.
- Minimal ratio is defined as the proportion of eigenvalues greater than zero.



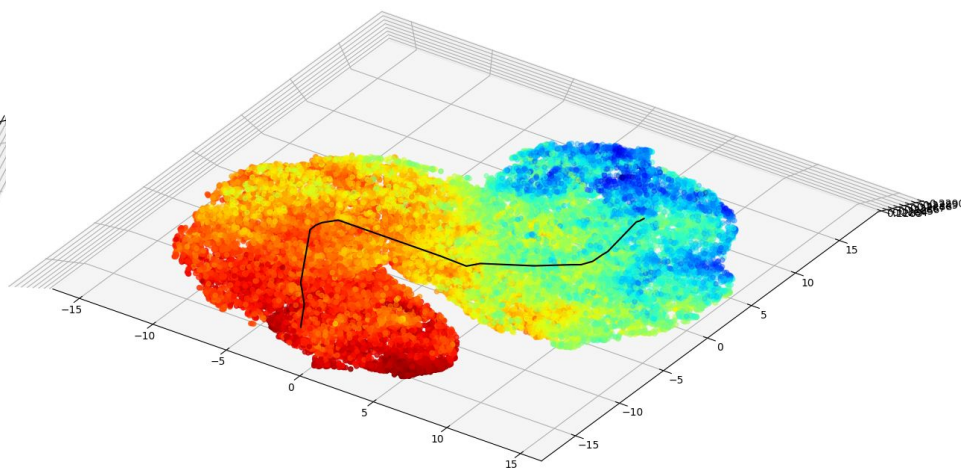
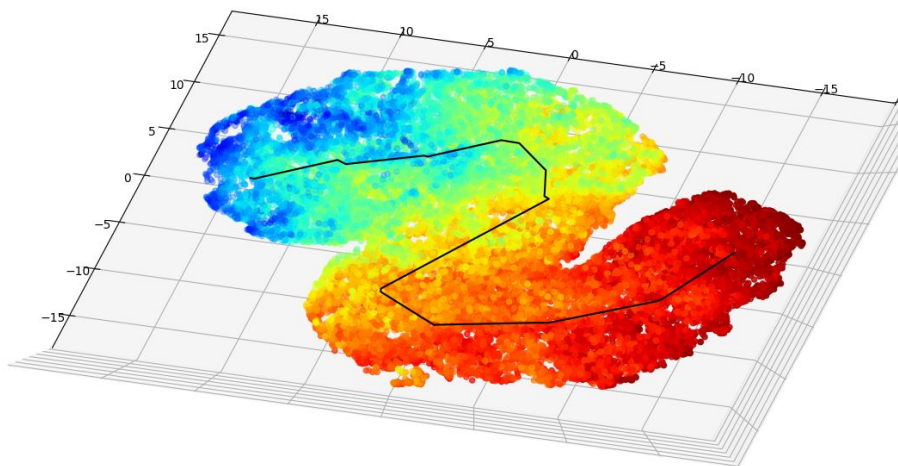
# HW1-2 Report Questions (10%)

- Visualize the optimization process.
  - Describe your experiment settings. (The cycle you record the model parameters, optimizer, dimension reduction method, etc) (1%)
  - Train the model for 8 times, selecting the parameters of any one layer and whole model and plot them on the figures separately. (1%)
  - Comment on your result. (1%)
- Observe gradient norm during training.
  - Plot one figure which contain gradient norm to iterations and the loss to iterations. (1%)
  - Comment your result. (1%)
- What happens when gradient is almost zero?
  - State how you get the weight which gradient norm is zero and how you define the minimal ratio. (2%)
  - Train the model for 100 times. Plot the figure of minimal ratio to the loss. (2%)
  - Comment your result. (1%)
- Bonus (1%)
  - Use any method to visualize the error surface.
  - Concretely describe your method and comment your result.



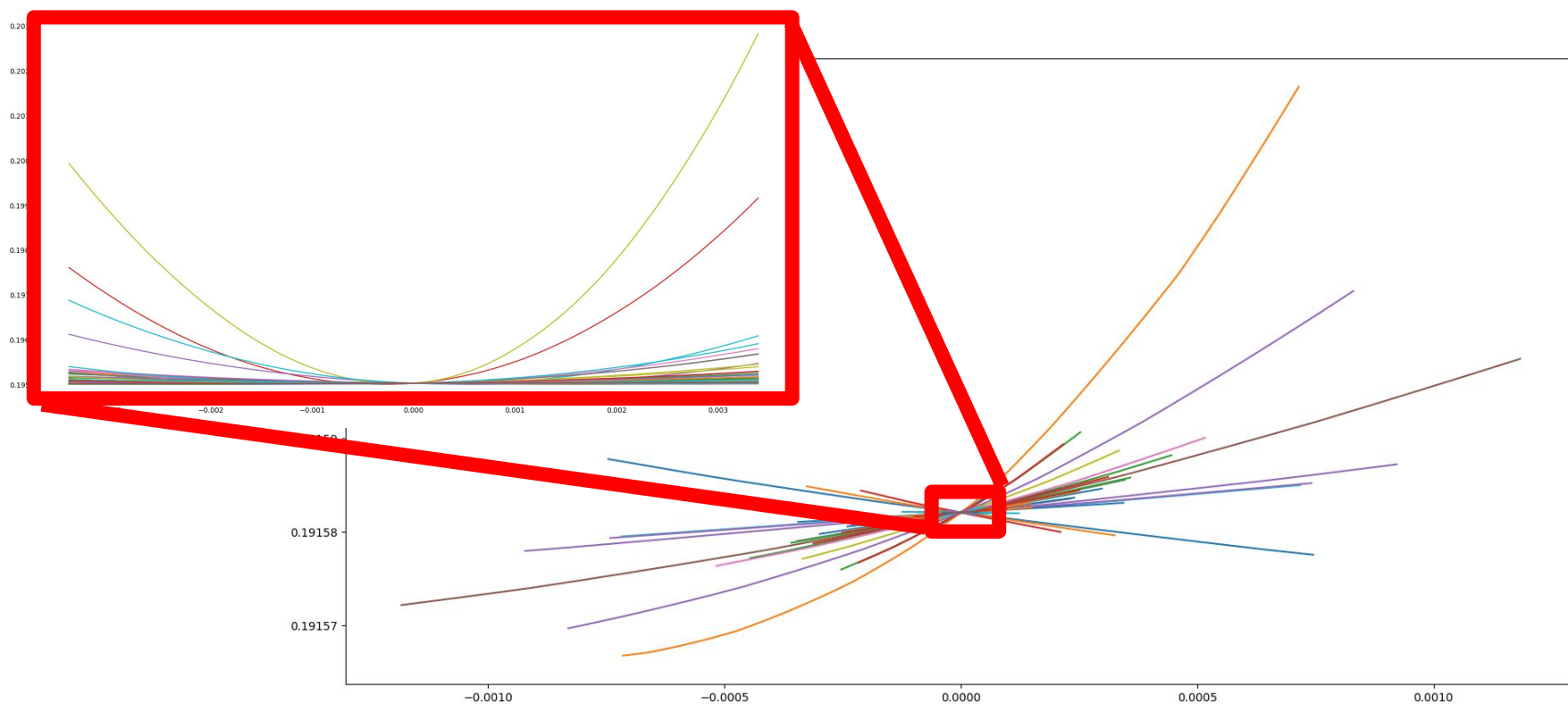
# Example of Bonus <sup>1/3</sup>

- Similar method as [pg.10](#), but use TSNE to reduce dimensionality.
- First train with gradient descent, then use second order optimization, finally train for about 10 epochs further by second order optimization.
- During the 10 epochs, randomly plot nearby parameters.
- Tips:
  - Use small model (less than 50 parameters) on small tasks (simulate function)
  - Fixed number of possible input (thus number of possible output is also fixed)
  - Scale the range of randomness according to each parameters' rate of descending
  - Non-linear coloring



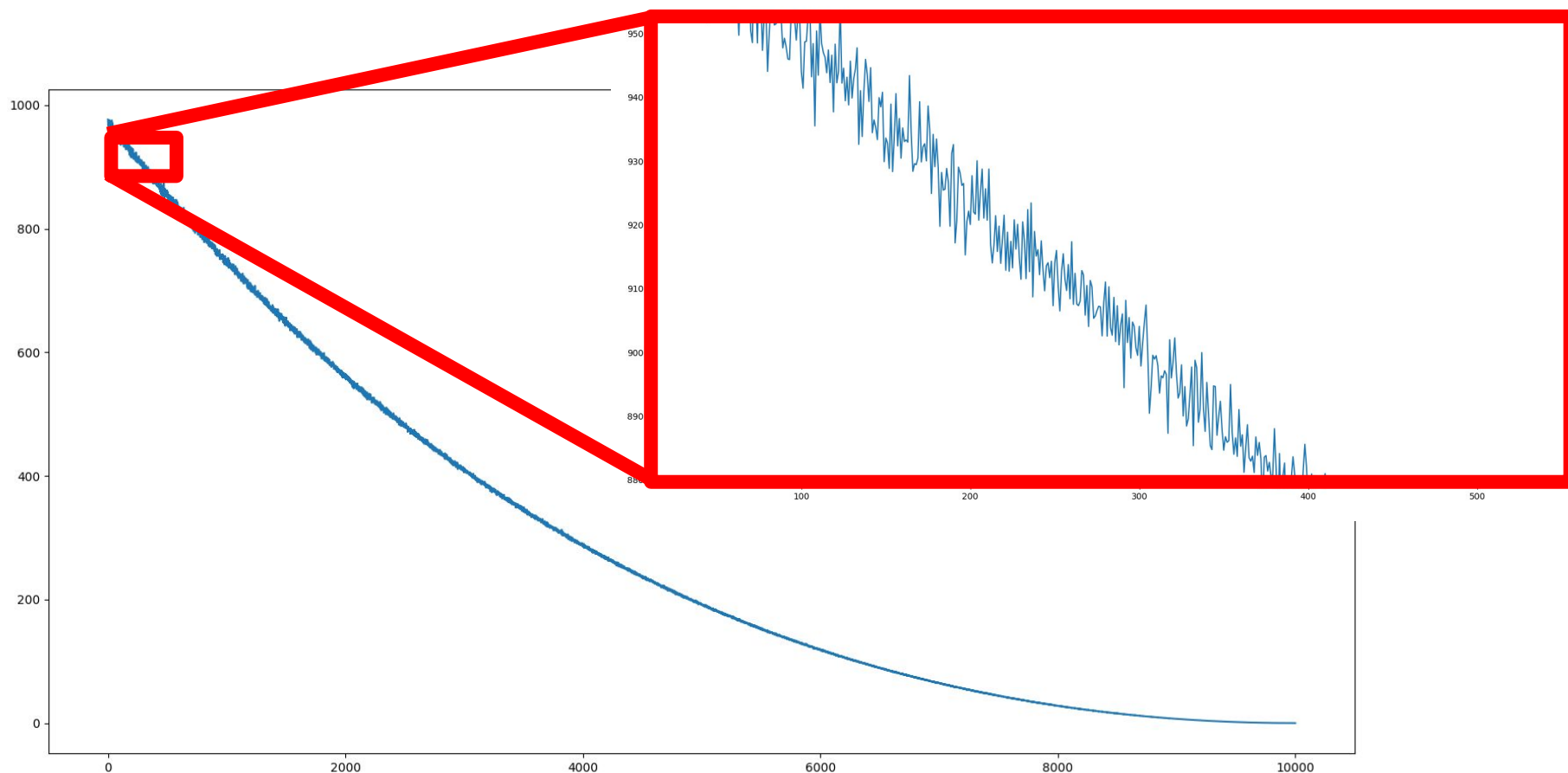
# Example of Bonus <sup>2/3</sup>

- Perturb each parameter randomly within a small range and plot the resulting loss.
- Scale differences between parameters are significant.



# Example of Bonus <sup>3/3</sup>

- Plot the error surface between start and end point.



# Allow Packages

- python 3.6
- TensorFlow r1.6
- PyTorch 0.3 / torchvision
- Keras 2.0.7 (TensorFlow backend only)
- MXNet 1.1.0
- CNTK 2.4
- matplotlib
- scikit-learn 0.19.1
- Python Standard Library
- If you want to use other packages, please ask TAs for permission first!

# Submission

- Deadline: **2018/4/6 23:59 (GMT+8)**
- Write the questions of HW1-1, HW1-2 and HW1-3 in **one** report.
- Chinese unless you are not familiar with Chinese
- At most 10 pages for HW1-1, HW1-2 and HW1-3
- Your github must have several files under directory hw1/
  - Readme.\*
  - Report.pdf
  - other code
- In your Readme, state clearly how to run your program to generate the results in your report.
- Files for training is required.

# Q&A

[ntu.mldsta@gmail.com](mailto:ntu.mldsta@gmail.com)