MLDS HW 1-1 Deep v.s. Shallow

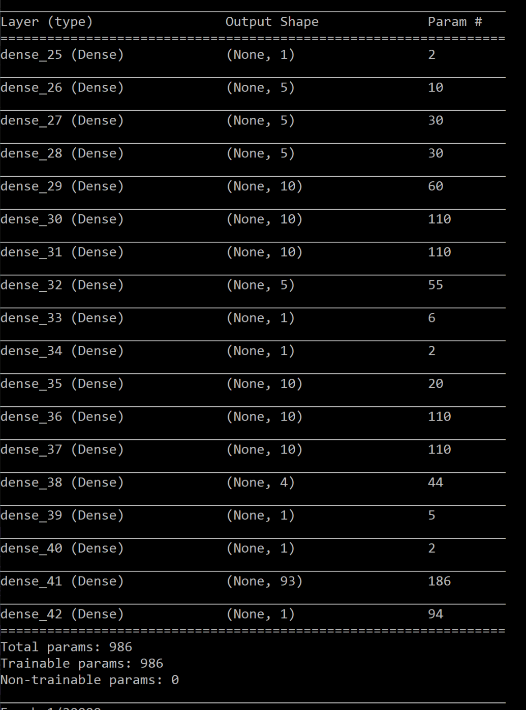
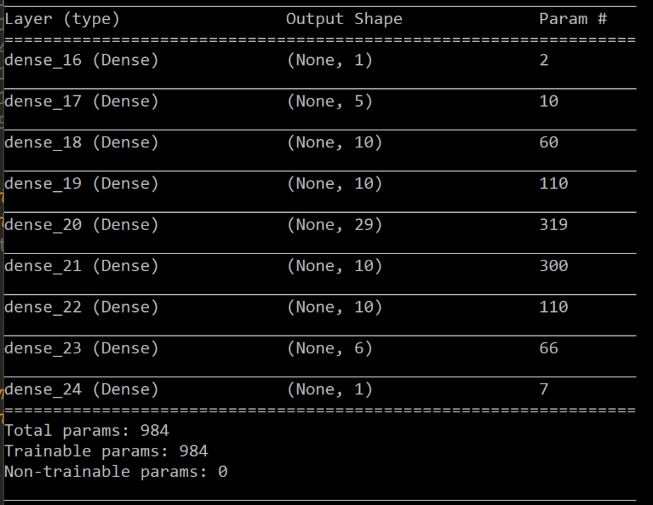
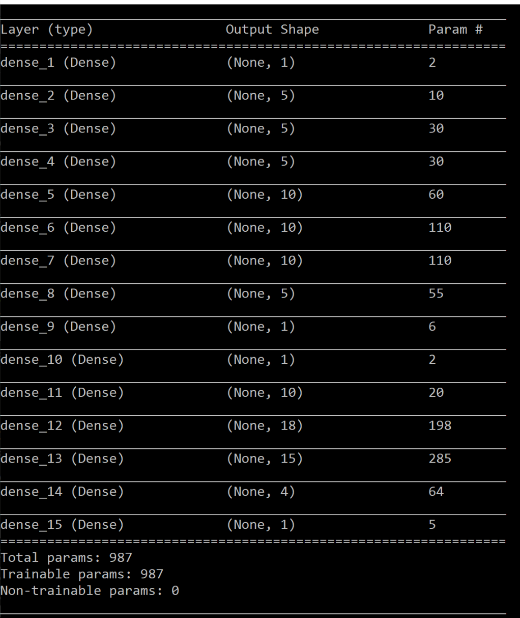
1. Simulate a Function:

* Describe the models you use, including the number of parameters (at least two models) and the function you use.

Function 1: / Function 2:

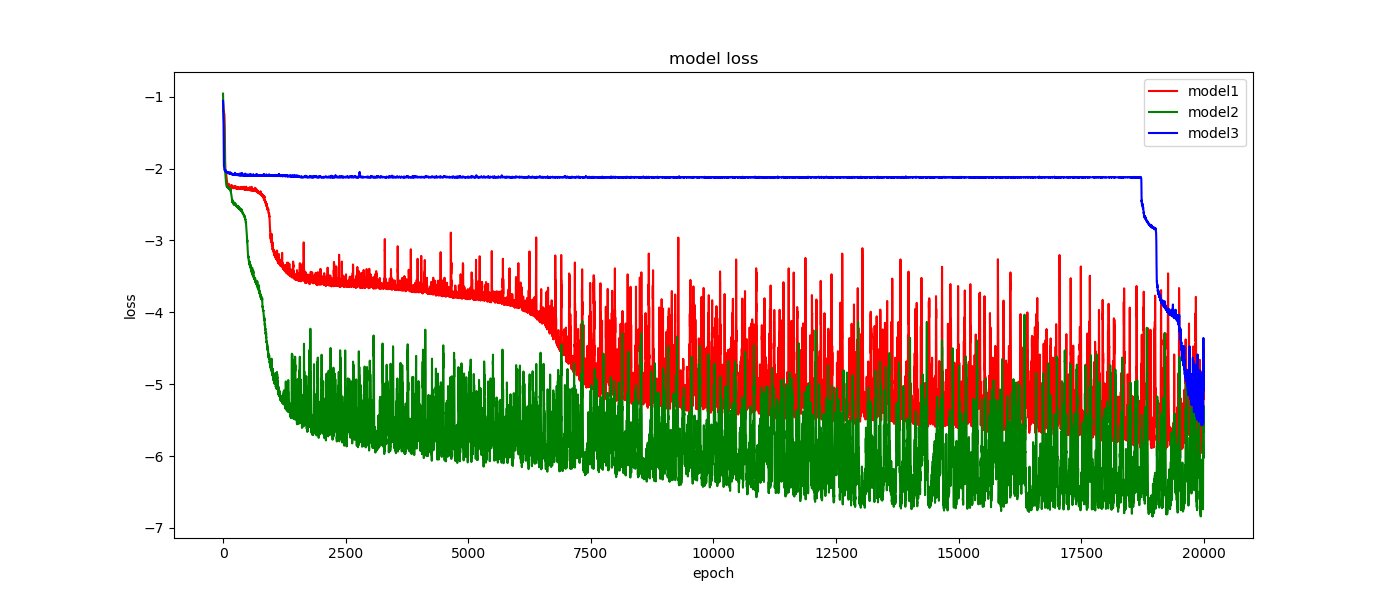
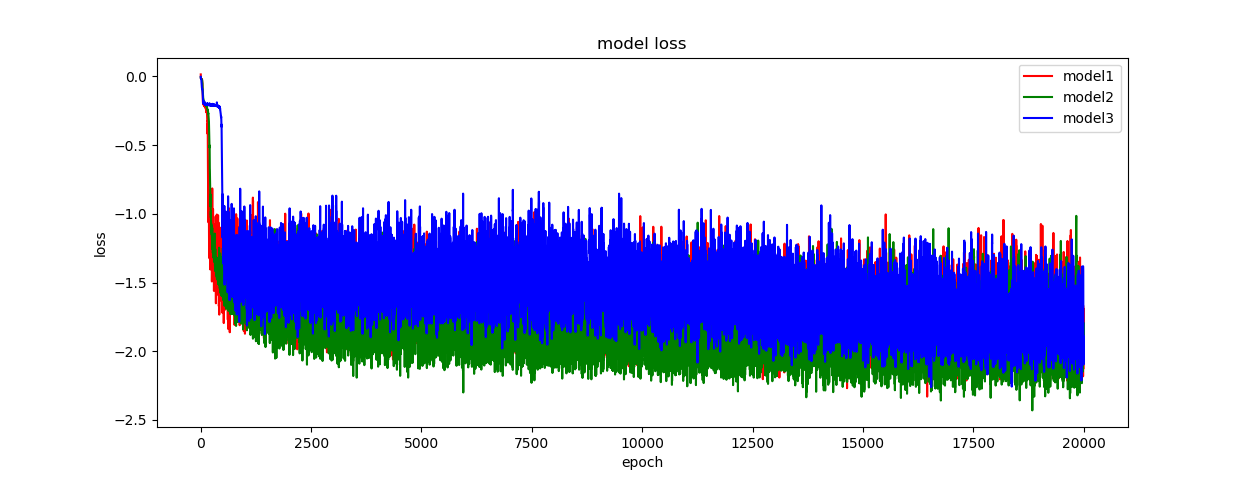
Each layer of these three models is **tanh** function.

Model 1(#parameters: 987) Model 2 (#parameters: 986) Model 3 (#parameters:984)

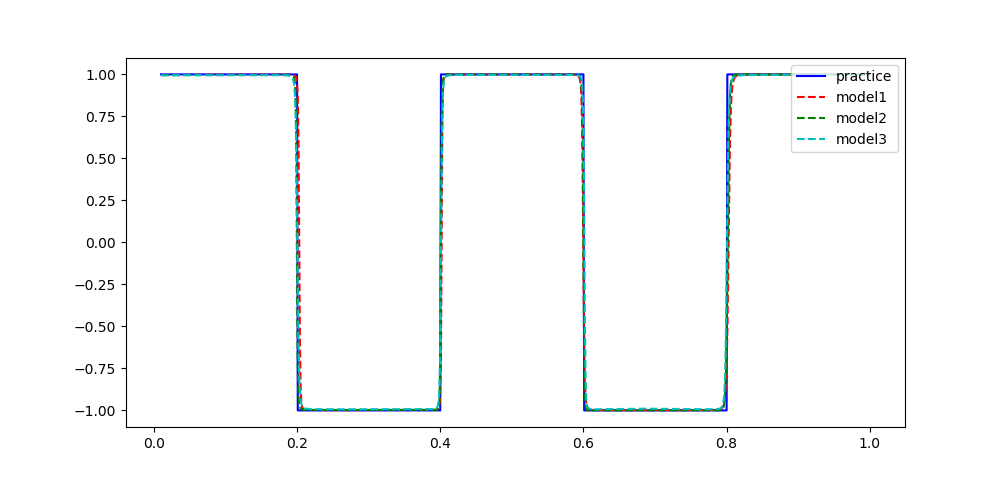


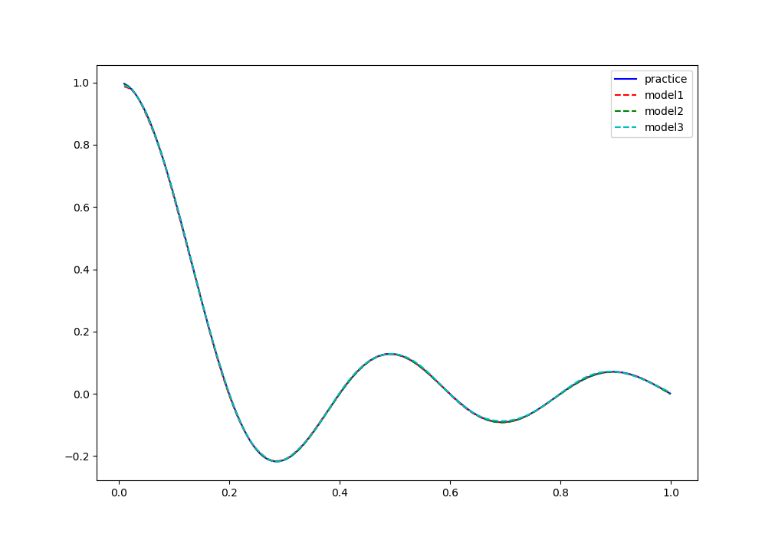
* In one chart, plot the training loss of all models.

Function 1: Function 2:



* In one graph, plot the predicted function curve of all models and the ground-truth function curve.

Function 1: Function 2:



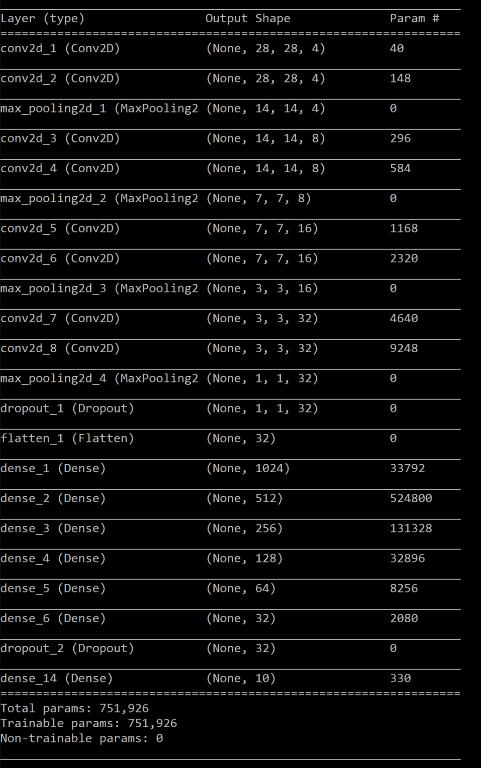
* Comment on your results.

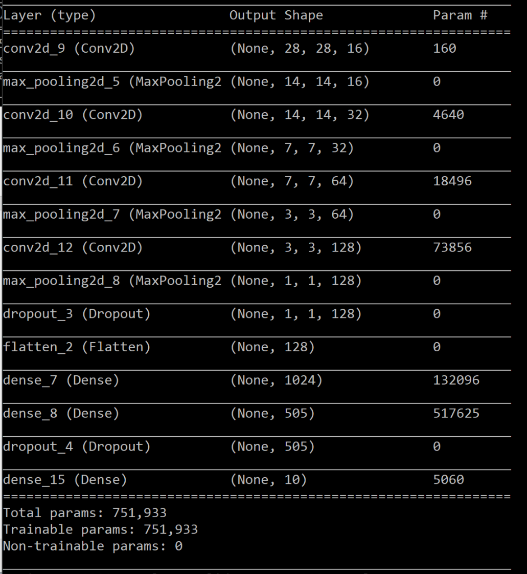
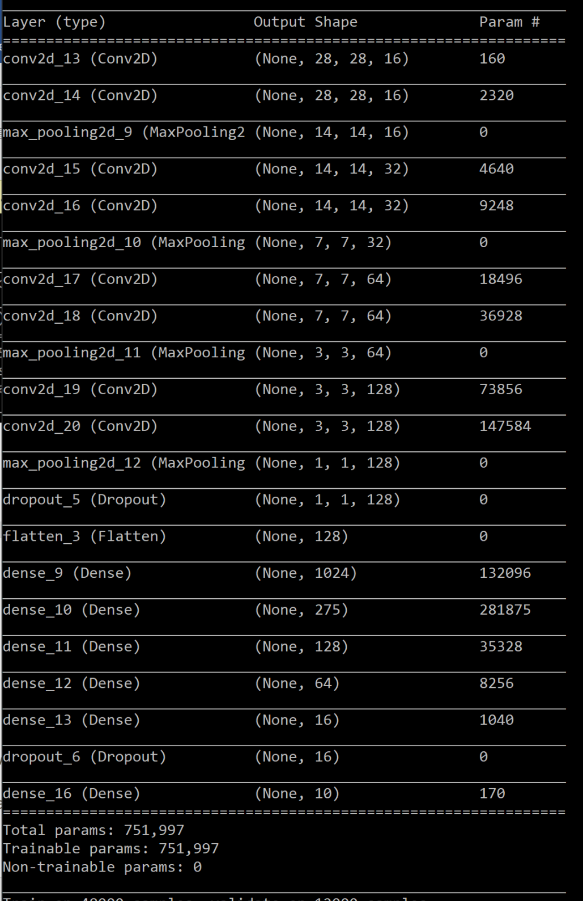
From the above experiment results, it can be observed that when the model gets deeper and more complicated, the loss of the model cannot always decrease faster than the shallow one. I supposed that one deep model may stuck in the saddle point (local minimum) if the initialization of the model is near around the original point, which may be the reason why Model 3 (the deepest one) has flatter loss curve in Function 1.

1. Train on Actual Tasks:

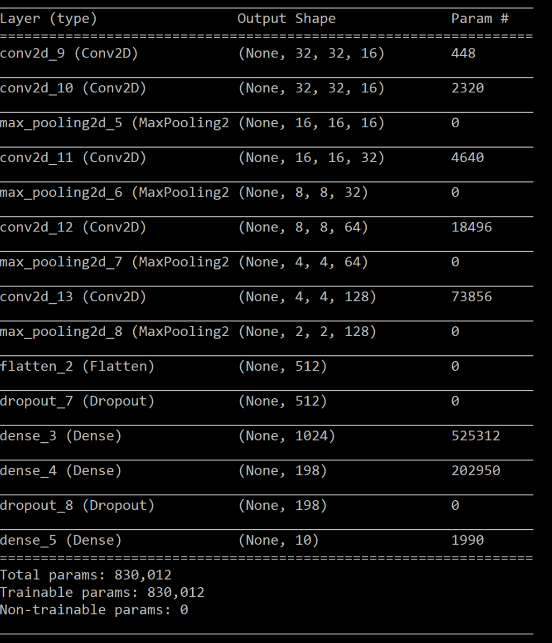
* Describe the models you use and the task you chose.

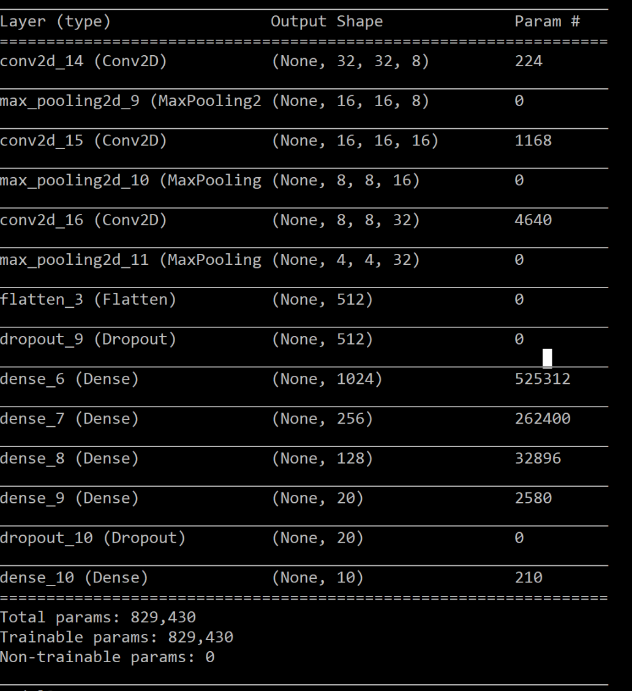
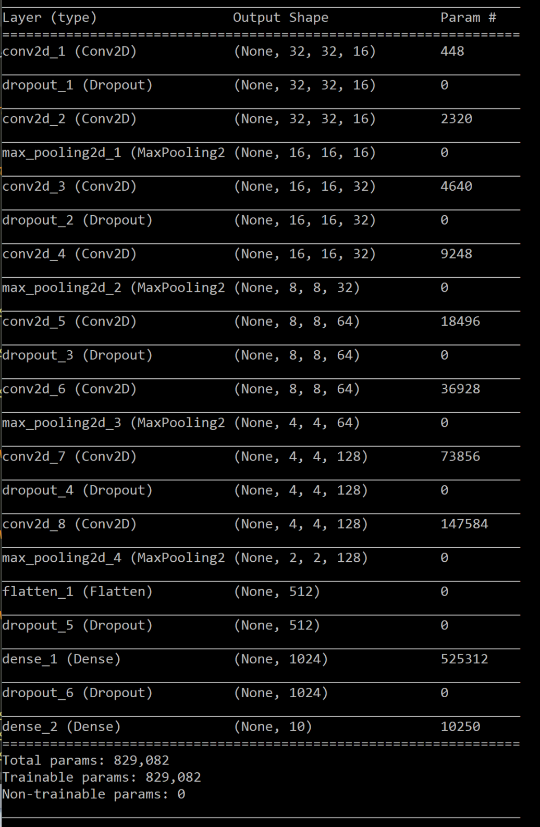
Task 1: **MNIST**

Model 1(#parameters: 751933) Model 2 (#parameters: 751997) Model 3 (#parameters: 751926)



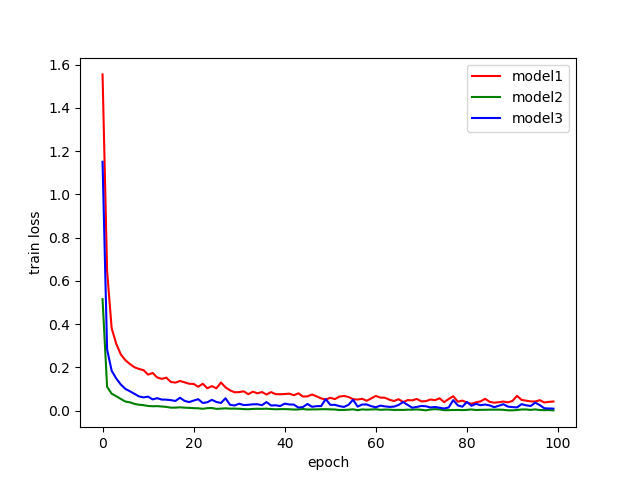
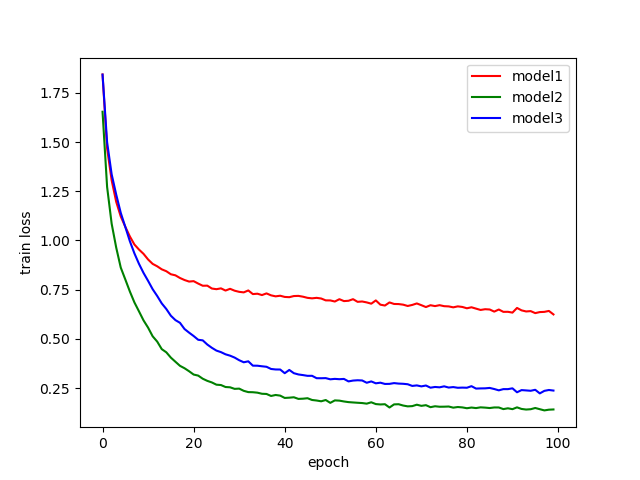
Task 2: **CIFAR-10**

Model 1(#parameters:830012) Model 2 (#parameters: 829082) Model 3 (#parameters: 829430)



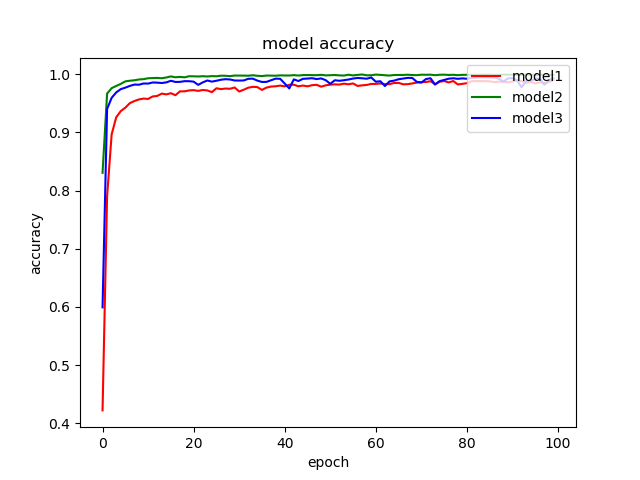
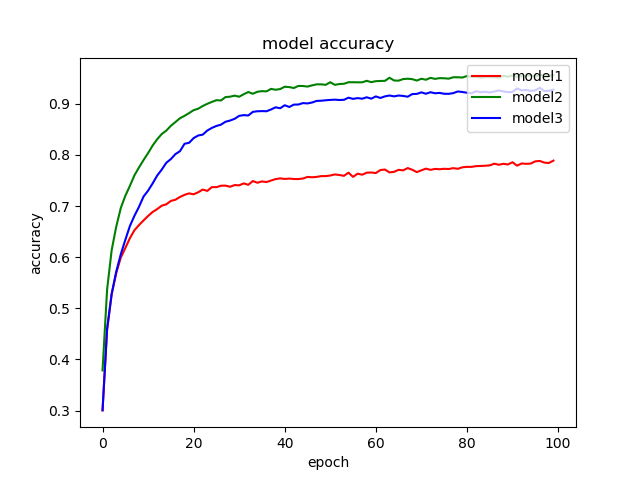
* In one chart, plot the training loss of all models.

Task 1: **MNIST** Task 2: **CIFAR-10**



* In one chart, plot the training accuracy.

Task 1: **MNIST** Task 2: **CIFAR-10**



Cifar10

* Comment on your results.
* Comment on your results.

From the above experiment results, it can be observed that when the model gets deeper and more complicated, the loss value decreases faster and the accuracy value increases faster than the shallower ones. Since the task of MNIST is simpler, the final performance of the there models are similar; however, it can be observed that in the task of CIFAR-10, the final performances of the three models are different among one another.

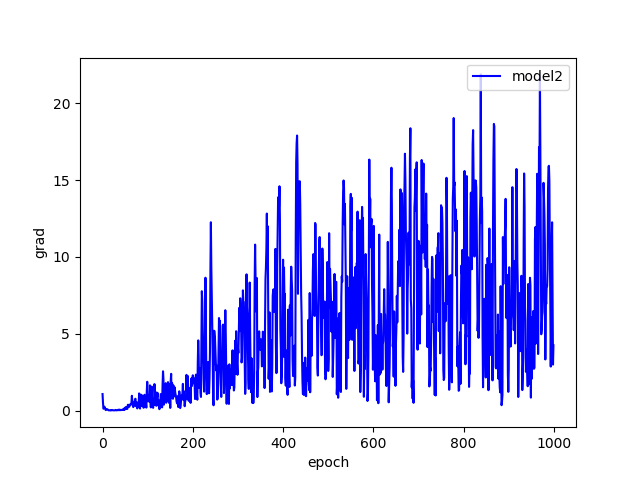
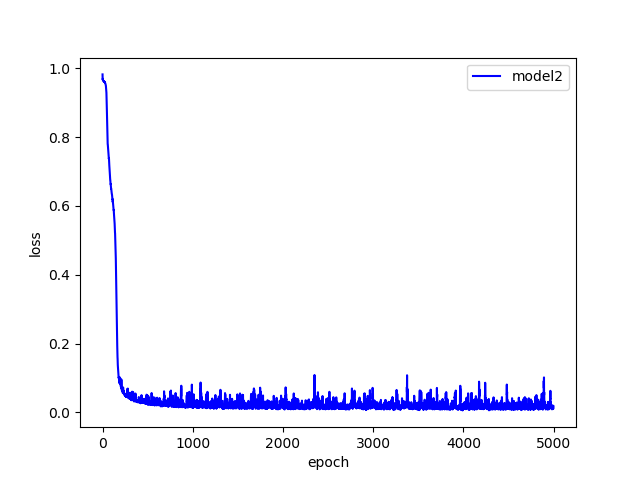
MLDS HW 1-2 Optimization

1. **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%)

1. **Observe gradient norm during training.**

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



* Comment your result.

From the above result, it can be observed that the gradient norm changes slowly at the beginning, but shows substantial changing when the epoch is up to 200. I think it may because task MNIST has a flatten plan near the original point in the loss surface.

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%)

MLDS HW 1-3 Generalization

1. **Can network fit random variables?**

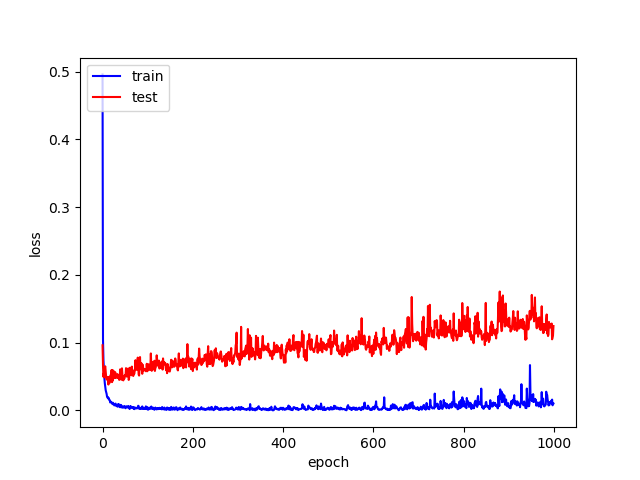
* Describe your settings of the experiments. (e.g. which task, learning rate, optimizer)

Task: MNIST

Learning rate = 0.005

Optimizer: Adam

* Plot the figure of the relationship between training and testing, loss and epochs.

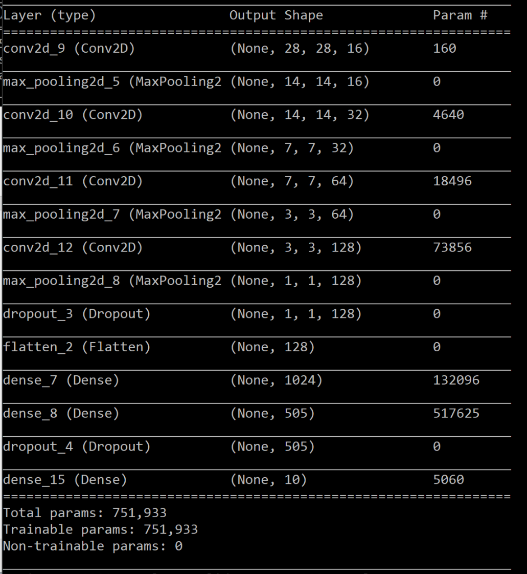


1. **Number of parameters v.s. Generalization**

* Describe your settings of the experiments. (e.g. which task, the 10 or more structures you choose)

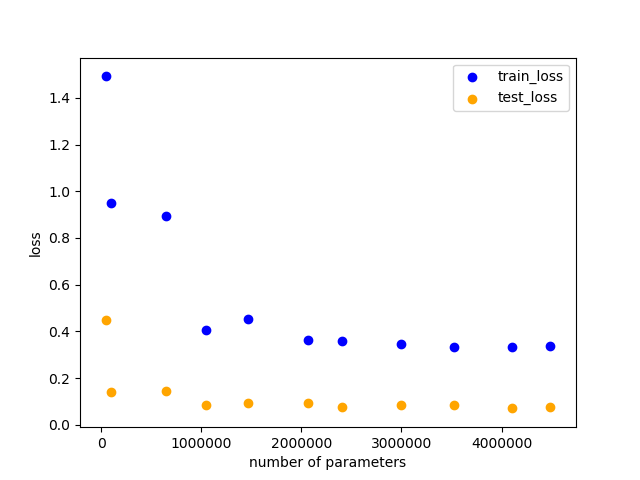
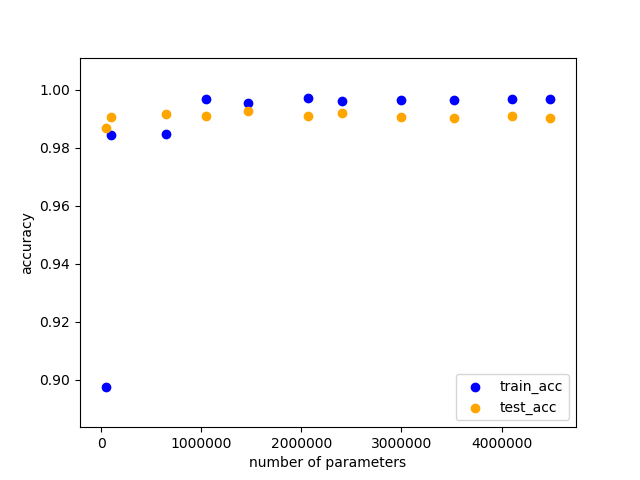
Task: MNIST

Structure: The basic model is listed on the below:



Setting: Change the basic model to the ones with different amounts of parameters

* Plot the figures of both training and testing, loss and accuracy to the number of parameters.



* Comment your result.

From the above results, it can be observed that when the number of parameters increases, the loss value indeed decreases the same as our expected thoughts; however, the accuracy value seems to be less different when the number of parameters are more than around 750000.

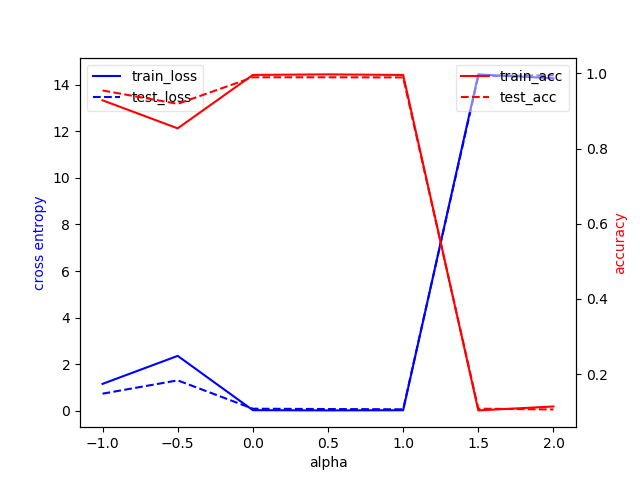
1. **Flatness v.s. Generalization**

* Part 1:
  + Describe the settings of the experiments (e.g. which task, what training approaches)

Task: MNIST

Training approaches: batch size 64 v.s. batch size 1024

* + Plot the figures of both training and testing, loss and accuracy to the number of interpolation ratio.



* + Comment your result.

It can be observed from the above result that when the alpha value increases, which has indicated that the batch size is larger, the accuracy value is decreasing and the loss value is increasing, and vise versa.

* Part 2 :
  + Describe the settings of the experiments (e.g. which task, what training approaches) (0.5%)
  + Plot the figures of both training and testing, loss and accuracy, sensitivity to your chosen variable. (1%)
  + Comment your result. (1%)

分工:

|  |  |  |
| --- | --- | --- |
| r06944046 | r06922141 | b03901156 |
| 33.3% | 33.3% | 33.3% |