MLDS HW1 Report

電機三 張家銘、張承洋、張景程

1-1

- Simulate a Function:
  - Describe the models you use, including the number of parameters (at least two models) and the function you use. (0.5%)

    # of parameters : 385

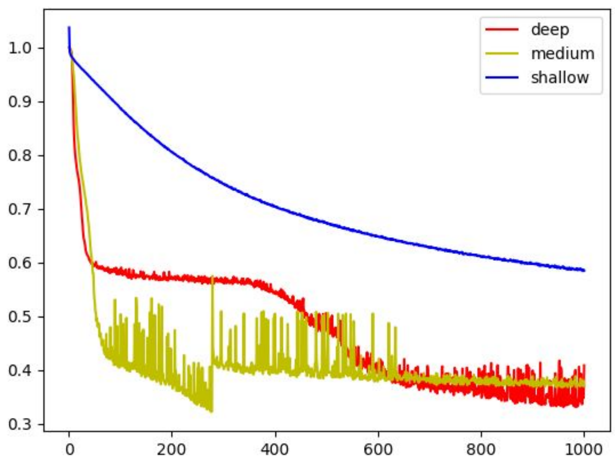
    Deep　: DNN七層，units : 8, 8, 8, 8, 8, 8, 1

    Medium：DNN五層，units : 16, 8, 16, 4, 1

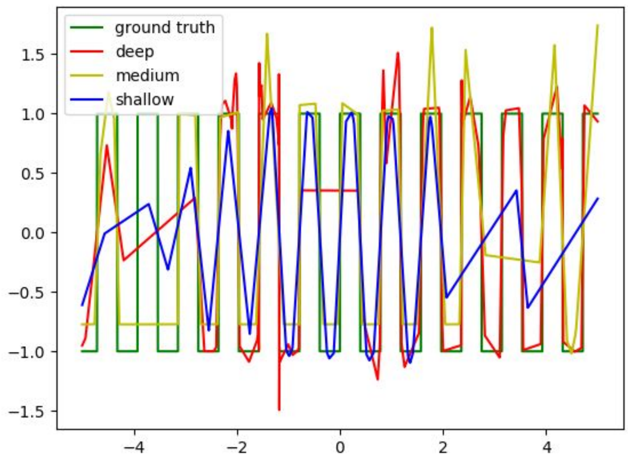
    Shallow：DNN兩層，units : 128, 1
  - In one chart, plot the training loss of all models. (0.5%)
  - In one graph, plot the predicted function curve of all models and the ground-truth function curve. (0.5%)

ground truth : **sign(sin(8*X))**

training loss



predicted function curve

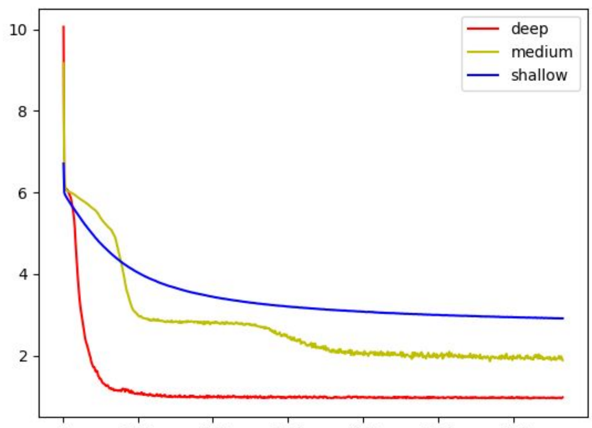


  - Comment on your results. (1%)

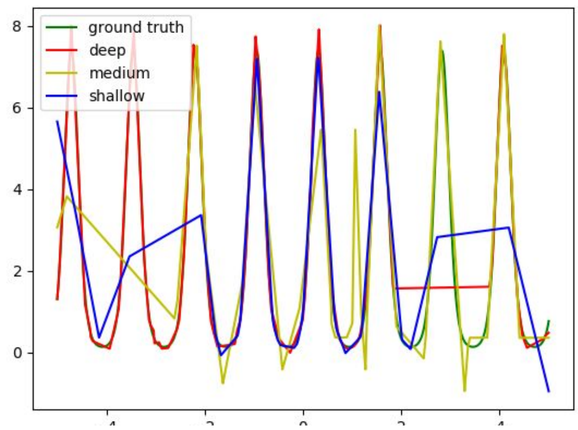
    Deep model的training loss最低，且可以更貼近函數，雖然可能會train比較久，而Shallow model的loss下降最緩慢，也比較難fit到原先設定的target function.
  - Use more than two models in all previous questions. (bonus 0.25%)
  - Use more than one function. (bonus 0.25%)

ground truth : **exp(2*sin(5*X))**

training loss



predicted function curve



- Train on Actual Tasks:
  - Describe the models you use and the task you chose. (0.5%)

    Task : MNIST

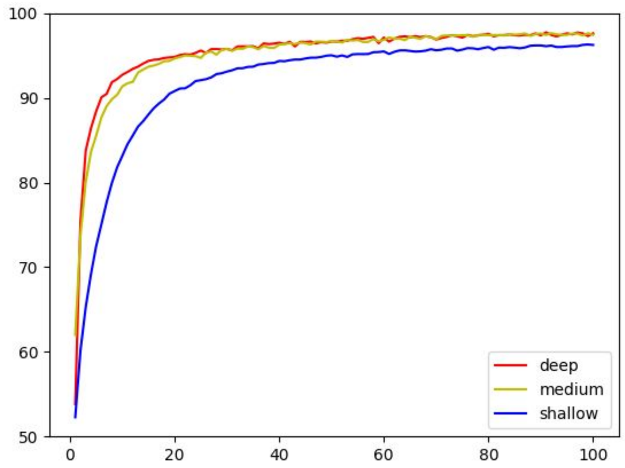
    Models : 三種model皆為CNN+DNN，固定DNN層數為三層並調整CNN層數來創建不同深度的model

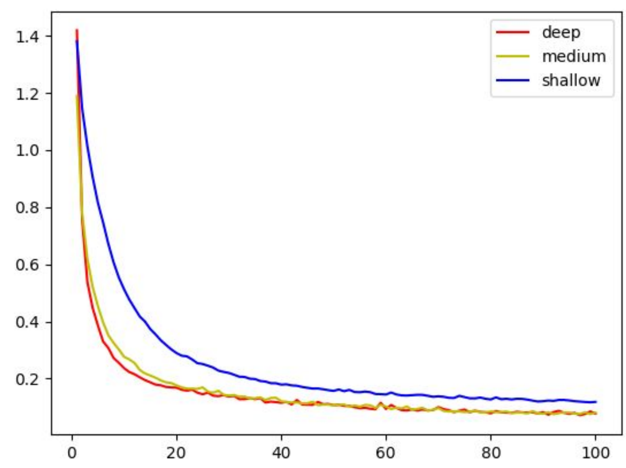
    Deep　　: CNN units : 5, 5, 5　　　　#params:2568

    Medium : CNN units : 2, 4　　　　　#params:2666

    Shallow : CNN units : 4　　　　　　#params:2610
  - In one chart, plot the training loss of all models. (0.5%)
  - In one chart, plot the training accuracy. (0.5%)

    training accuracy　　　　　　　　　training loss





  - Comment on your results. (1%)

    可能是mnist太好train，所以Deep, medium 的差別不大，但在training初期epoch數小的時候就看得出兩者的差距，Deep表現比較好。
  - Use more than two models in all previous questions. (bonus 0.25%)
  - Train on more than one task. (bonus 0.25%)

    Task : CIFAR-10

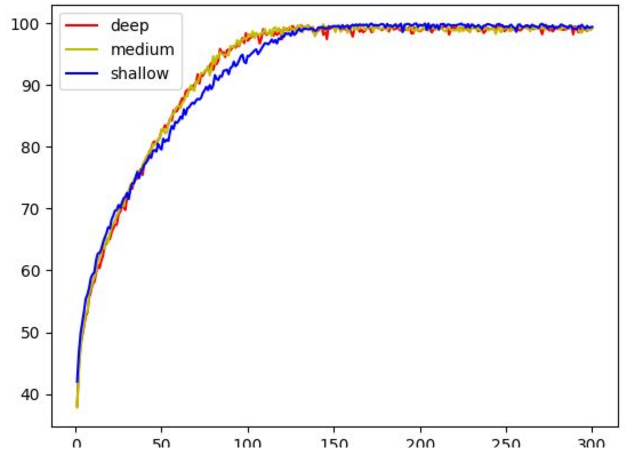
    Models : 仿照mnist的model，三種model皆為CNN+DNN，固定DNN層數為三層並調整CNN層數來創建不同深度的model

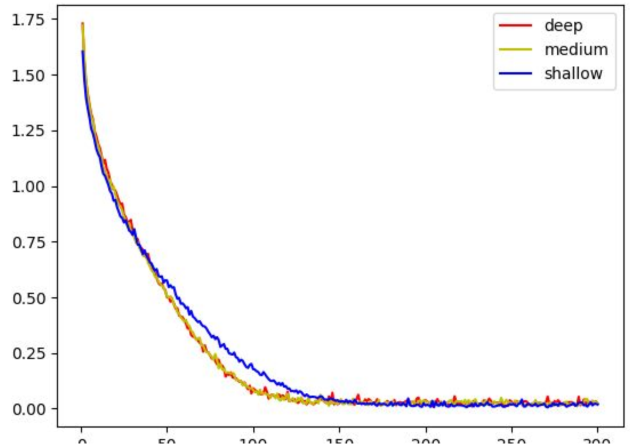
    Deep　　: CNN units : 8, 8, 8　　　　#params:104186

    Medium : CNN units :10, 8　　　　　#params:104154

    Shallow : CNN units : 15　　　　　　#params:104318

    training accuracy　　　　　　　　　training loss





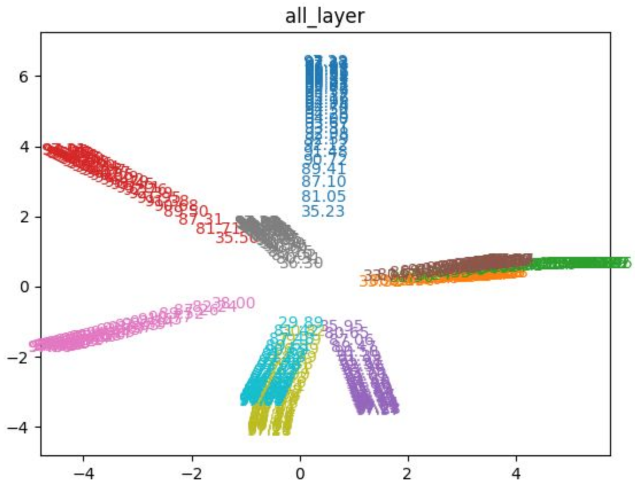
1-2

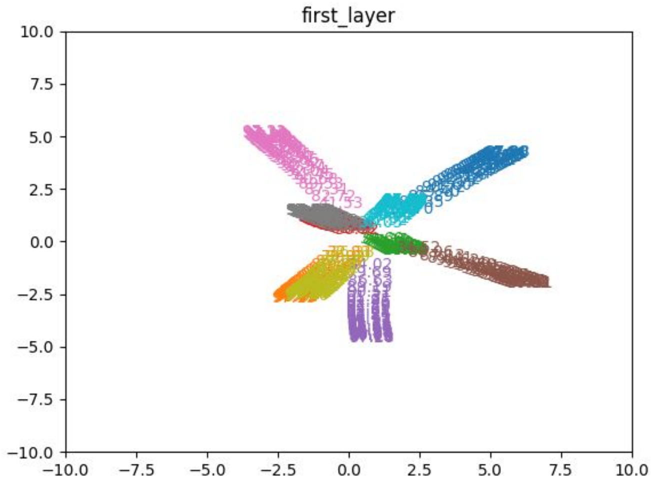
- Visualize the optimization process.
  - Describe your experiment settings. (The cycle you record the model parameters, optimizer, dimension reduction method, etc) (1%)

    Cycle: 每3個epoch 紀錄一次parameter

    Optimizer: Adam

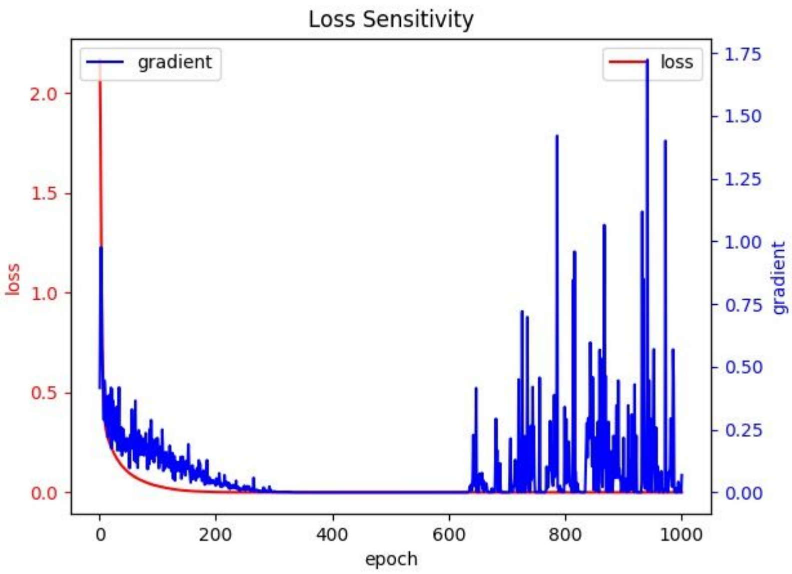
    Dimension reduction: PCA
  - Train the model for 8 times, selecting the parameters of any one layer and whole model and plot them on the figures separately.(1%)





  - Comment on your result. (1%)

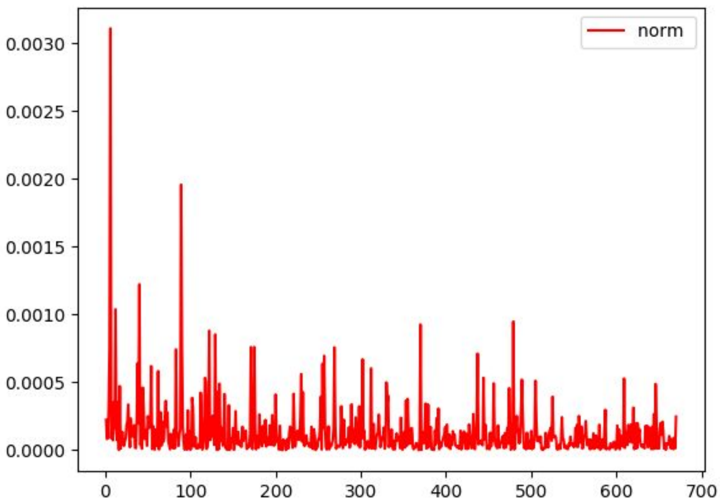
    根據上課內容，每次train出來的結果很有可能會落在不同的流域，而實際實驗的結果也跟上課內容相同，不過有可很特別的情況，就是每次initial的參數都initial到接近的位置，然後會都是往外跑，這個結果很讓人驚訝。
- Observe gradient norm during training.
  - Plot one figure which contain gradient norm to iterations and the loss to iterations. (1%)

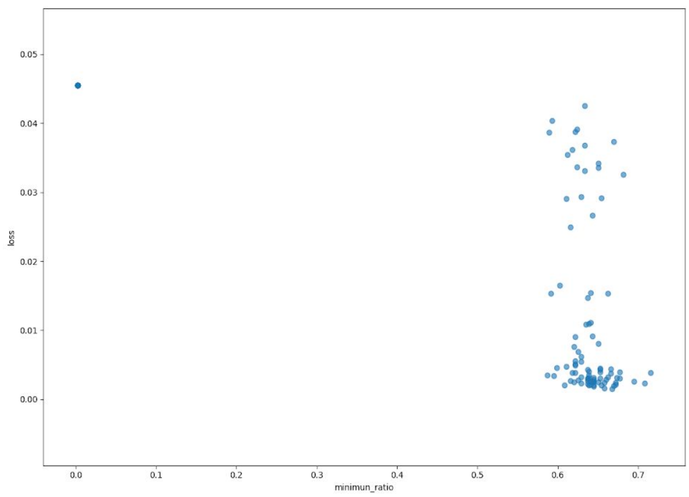


- Comment your result. (1%)
  由圖可觀察train到500個epoch的時候gradient會幾乎降為0，可是到了六百多epoch之後gradient會突然暴增，和Ian　Goodfellow的實驗結果相符。然而卻十分難以理解，原本train到500多個epoch時我們覺得已經不太可能再改變參數了，然後後來的gradient卻非常非常的大，可是loss卻依舊穩穩的不再改變，這實在難以理解。

- 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%)
    我們的 task 是 simulate 一個 sinc function （128個點、參數量 521），作法是先以 mse 作為 loss function ，train 10000 個 epochs 之後換成以 gradient norm (two-norm) 作為 loss function ，然後繼續 train 10000 epochs ，之所以會選擇 10000 epochs 是因為我們觀察到通常到 10000 epochs 之後， loss 會趨於穩定。
    我們定義的 minimal ration 是 hessian matrix 的正的特徵值的比例。

  - Train the model for 100 times. Plot the figure of minimal ratio to the loss. (2%)

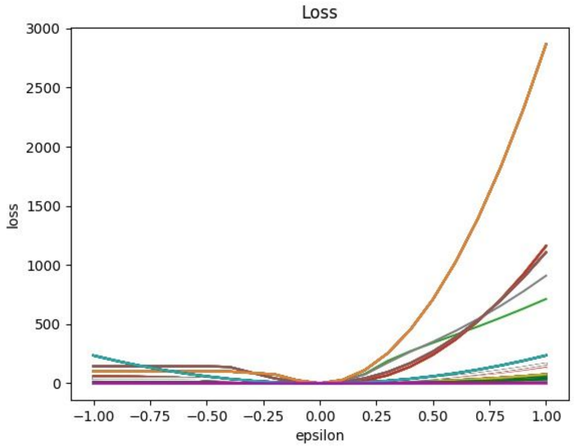


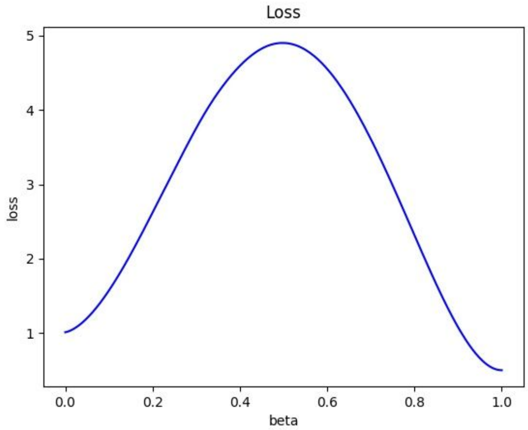


  - Comment your result. (1%)
    左圖是 gradient norm 對於 epoch 數作圖，可以發現雖然振盪得很明顯，但整體趨勢是下降的。右圖是 loss 對於 minimum ratio 作圖，可以觀察到右下方的點是最密集的，代表當 minimum ratio 愈大 (hessian matrix 正的特徵值比例愈大)， model 有愈好的表現 (loss 小)，當 minimum ratio 愈小 (hessian matrix 正的特徵值比例愈小)， model

有愈差的表現（loss 偏大，圖中左上方多點重疊)，但圖中可以發現右上方有一塊比較稀疏的點，推測是因為在以 mse 為 loss function 的階段，loss 的值沒有降下去，可能是 initial point 落在一個比較平坦的區域，導致10000 epochs 也沒辦法收斂。

- Bonus (1%)
    - Use any method to visualize the error surface.

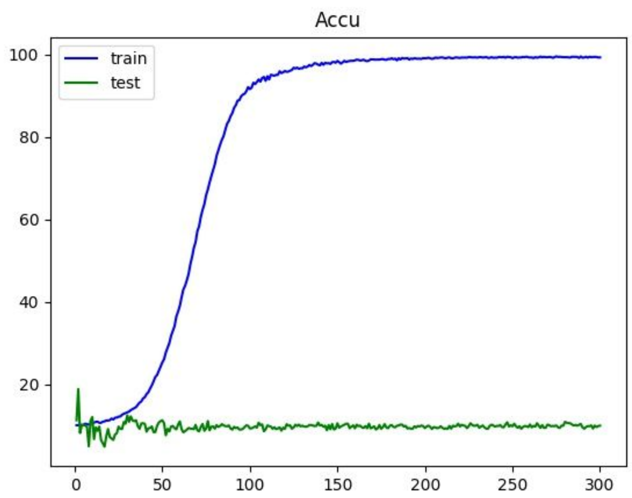


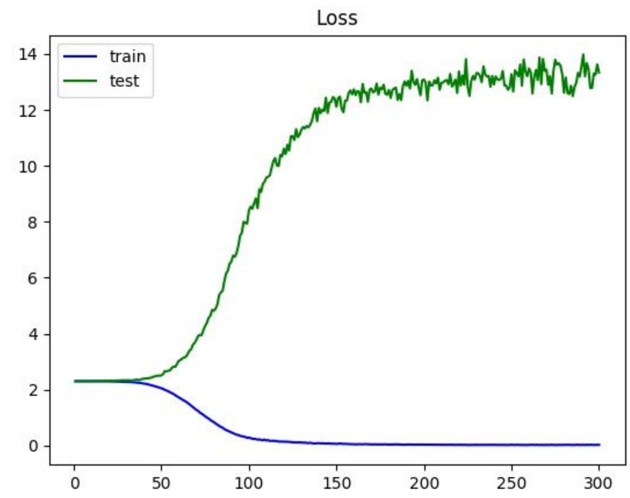


- Concretely describe your method and comment your result.
  1.繪製左圖時，我們直接算hessian matrix，為了不花費超過一天的時間，我們把參數量減少到161個，可是繪製出來的圖並不會像助教所繪製的。
  2.繪製右圖時，我們把起始點與終點之間切出50000個等分點，可是依舊沒有看到助教所繪製的振盪圖形。

1-3

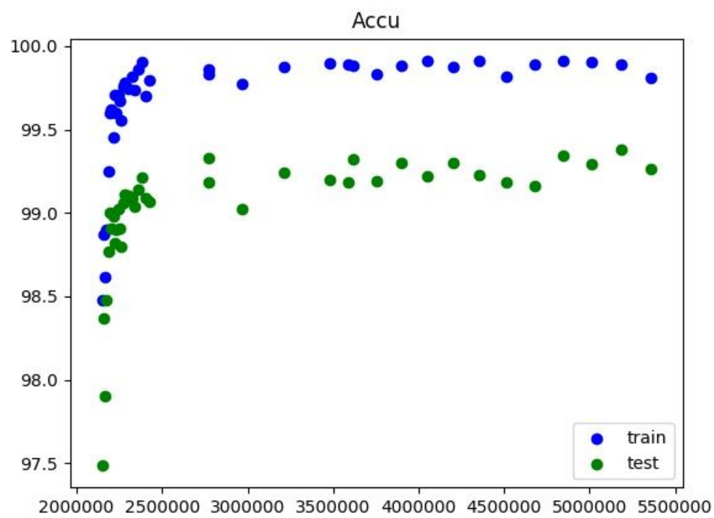
- Can network fit random variables?
    - Describe your settings of the experiments. (e.g. which task, learning rate, optimizer) (1%)
      Task : MNIST, Optimizer: Adam, Learning rate: 0.001
      CNN: 16
      DNN: 2048, 2048, 2048, 2048
    - Plot the figure of the relationship between training and testing, loss and epochs. (1%)

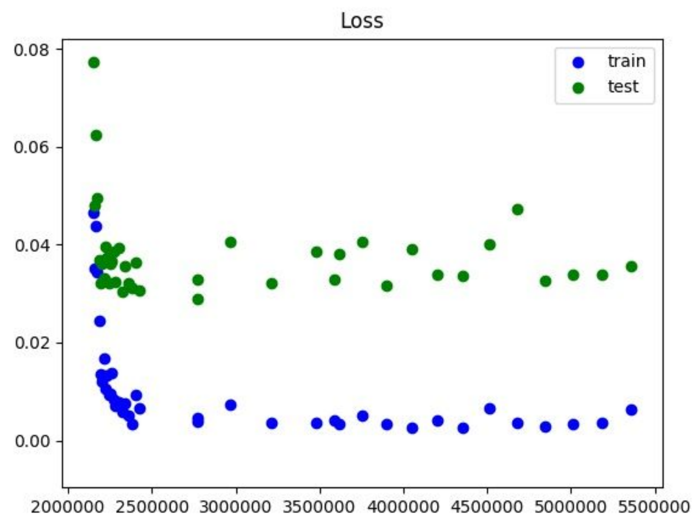




- Number of parameters v.s. Generalization
  - Describe your settings of the experiments. (e.g. which task, the 10 or more structures you choose) (1%)
    Task : MNIST, Optimizer: Adam, Learning rate: 0.001
    CNN: X, X, X
    DNN: 1024, 1024, 1024
    (Change X to change parameter amount.)

  - Plot the figures of both training and testing, loss and accuracy to the number of parameters. (1%)

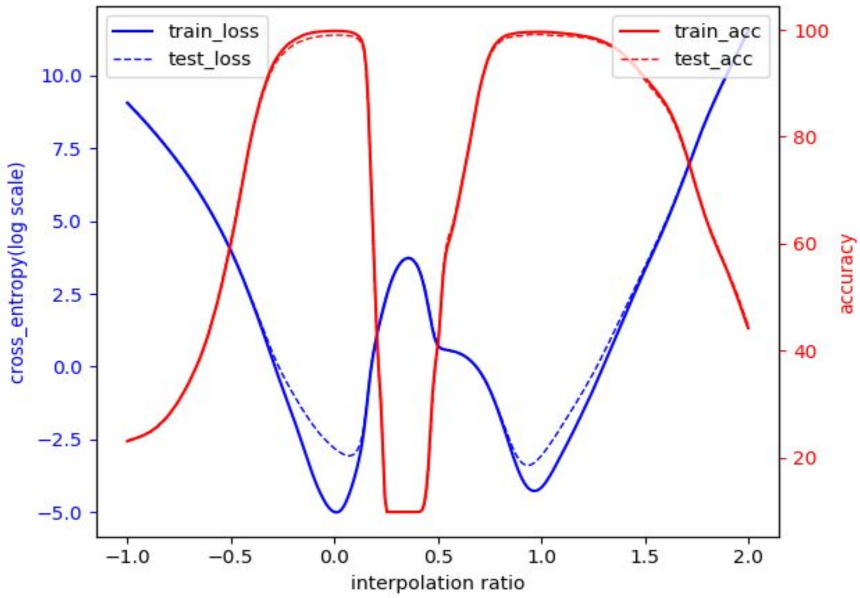




  - Comment your result. (1%)
    上課時說到，Neuron Network 自帶regularization，我們在可以train到100%Accuracy時，繼續增加參數量。實驗結果也如上課內容，參數量增多並不會影響generalize的能力。

- Flatness v.s. Generalization
  - Part 1:
    - Describe the settings of the experiments (e.g. which task, what training approaches) (0.5%)
      Task : MNIST, Optimizer: Adam, Learning rate: 0.001
      Model architecture : CNN 64,128,128 + DNN 1024,1024,512
      兩個model分別為batch_size = 64, 1024所train出來

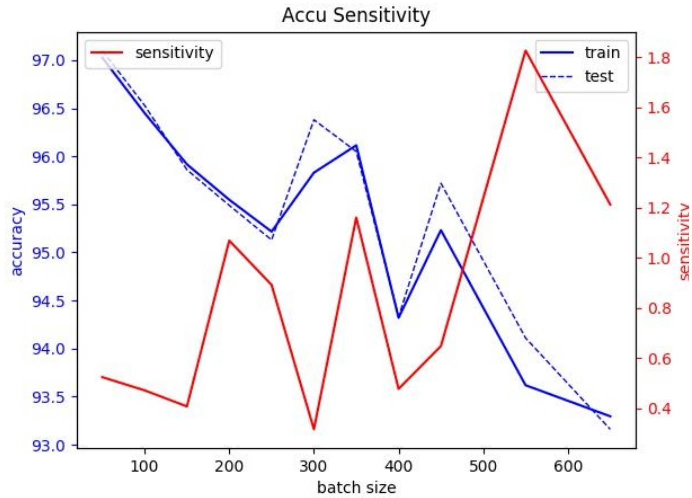
    - Plot the figures of both training and testing, loss and accuracy to the number of interpolation ratio. (1%)

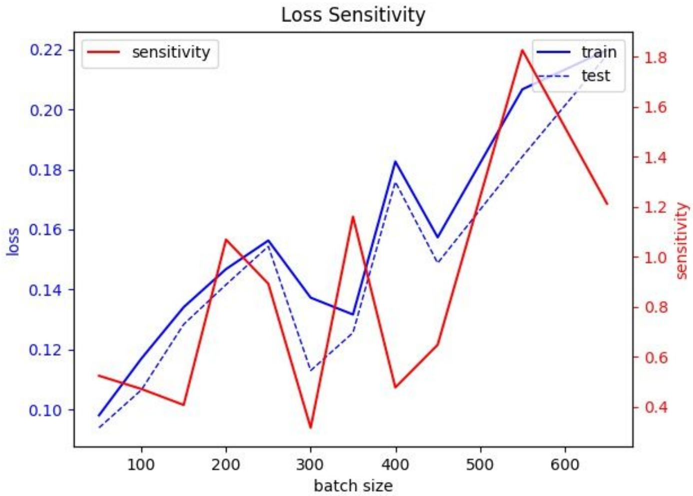


    - Comment your result. (1%)
      從圖中可以看出只有當interpolation ratio在0和1附近時才會得到最好的準確率，loss最低，且training的cross_entropy loss比testing還要低，和預期結果相符。

  - Part 2 :
    - Describe the settings of the experiments (e.g. which task, what training approaches) (0.5%)
      Task : MNIST, Optimizer: Adam, Learning rate: 0.001
      CNN units : 3, 3
      DNN units : 16

- Plot the figures of both training and testing, loss and accuracy, sensitivity to your chosen variable. (1%)





- Comment your result. (1%)
  我們的sensitivity採用的方式與助教相同，是gradient的frobenious norm，可是我們使用愈小的batch產生的sensitivity是愈小，也就是batch size 愈小，generalize能力愈強。

- Bonus : Use other metrics or methods to evaluate a model's ability to generalize and concretely describe it and comment your results.
  我們使用Hessian matrix 的最大eigen value 當作sharpness