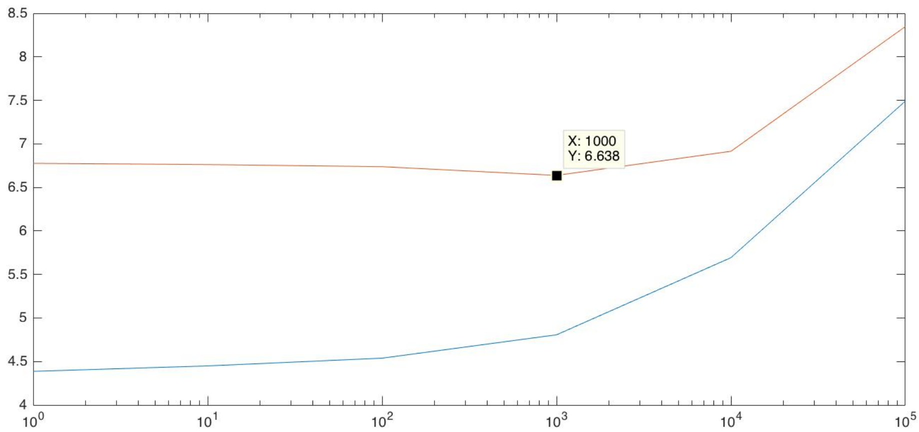
R05921063 陳定楷 HW1

1. **Linear regression function by Gradient Descent**
pseudo code of training model class

```
sigma_w_sq = 0
sigma_b_sq = 0
w = 0
bias = 0
for i in range(iteration):
        err = train_out -(bias+train_in*w)
        trainLoss = (sum(err**2)/len(train_out))**2
        pdv_w = (-2)*[err.T*train_in].T
        pdv_b = (-2)*sum(err)
        sigma_w_sq += pdv_w**2
        sigma_b_sq += pdv_b**2
        w = w-eta*pdv_w/(sigma_w_sq**0.5)
        bias = bias-eta*pdv_b/(sigma_b_sq**0.5)
test_out = bias+test_in*w
```

2. **Describe your method**
取用連續9個小時的18種資料作為feature，共9*18=162種features，預測第10小時的PM2.5濃度，總共做了5652次預測， $x \in R^{5652\times 162}$，$x_{i,j}$ 是第i次預測的第j個feature， $w = [w_1, w_2, \cdots, w_{162}]^T$ 是162種features的權重，w和bias的初始值都從0開始， 因為使用Adagrad，因此另外使用sigma_w_sq和sigma_b_sq紀錄微分的平方和。最後使用training 過iteration次的w和bias來預測testing data的output。kaggle_best使用的參數：$eta = 0.7$, $iteration = 100000$, training data使用3-fold cross validation

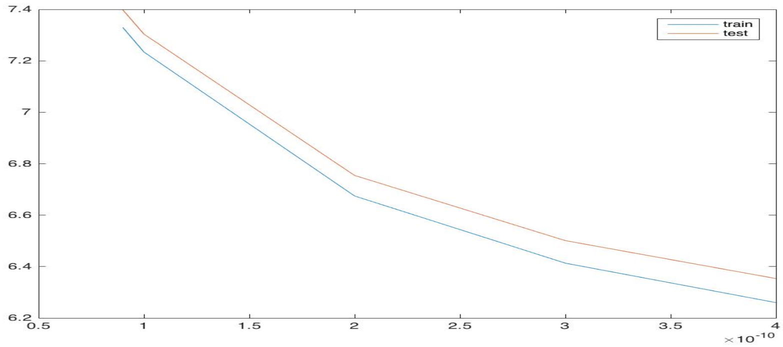
3. **Discussion on regularization**



(橫軸為lambda，縱軸為Loss，紅線為testing loss，藍線為training loss)
當資料量小的時候，容易overfit到training data，此時使用regularization可看出lambda=1000時loss達到最小值，故取lambda=1000

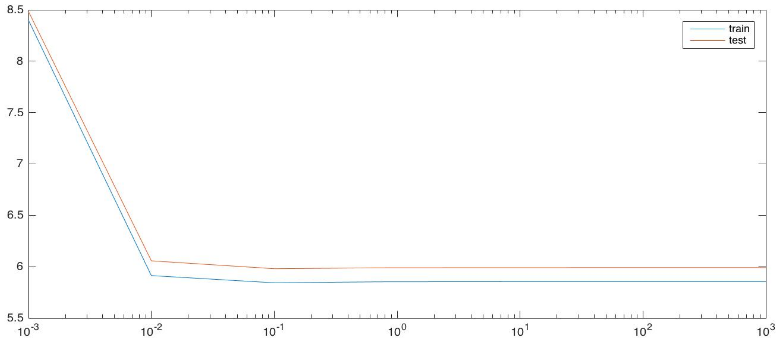
4. **Discussion on learning rate**

- 沒有使用Adagrad，iteration = 10000



(橫軸: Learning rate，縱軸: Loss)
Learning rate必須使用極小的值($\eta < 4 \times 10^{-10}$)才不會發散。
由Figure 1(a)，loss隨著learning rate變大而下降，代表iteration的次數還不足以收斂。

- 使用Adagrad，iteration = 10000



(橫軸: Learning rate，縱軸: Loss)
使用Adagrad後收斂變快，因此可從Figure 1(b)看出learning rate>0.01時，Loss會收斂到相同的值。
（Learning rate = 0.7時有Loss的最小值，Learning rate>0.7後Loss只會小幅上升（0.01））