Homework 1 Report - PM2.5 Prediction

學號：R06921038 系級：電機所碩一 姓名：謝宗宏

1. (1%) 請分別使用每筆data9小時內所有feature的一次項（含bias項）以及每筆data9小時內PM2.5的一次項（含bias項）進行training，比較並討論這兩種模型的root mean-square error（根據kaggle上的public/private score）。

   [設定]
   Learning Rate: 0.1 ， 並且使用 adagrad
   Lambda: 0 ，不使用 regularization
   Initialization: 使用 normal equation 初始化，再用 gradient descent 微調。
   Number of Iterations: 2000

   [9小時內所有feature的一次項]
   Score: 32.54171

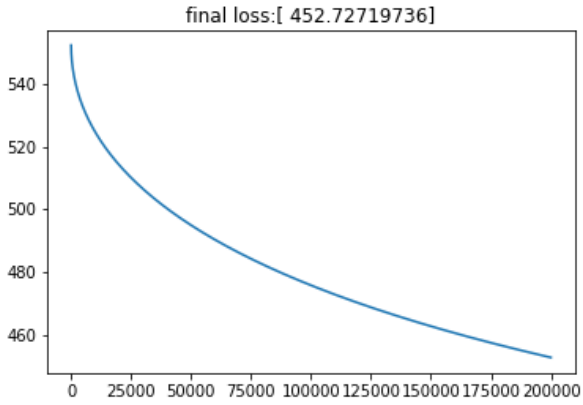
   [9小時內PM2.5的一次項]
   Score: 9.04825

   [討論]
   根據兩者比較結果，只使用 PM2.5 進行 training 的效果明顯比較好，推測是由於其他的 feature 可能不相關或甚至負相關，因此去除掉這些雜訊後能讓結果改善。從 train.csv 的資料來看，很明顯有些 feature 是不能用的，例如 rainfall 本身就有非常多 NR 的值，因此就算人為補零再一起拿下去 train 直覺上也不會有好結果。

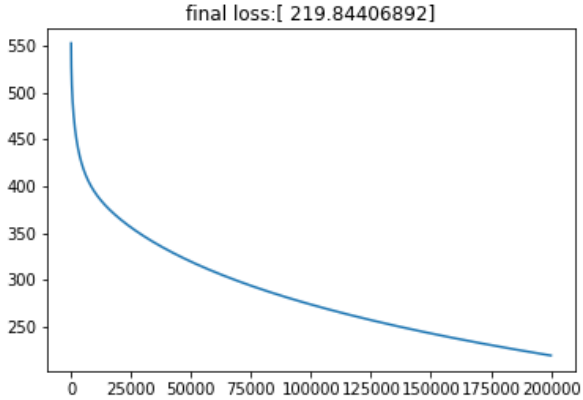
2. (2%) 請分別使用至少四種不同數值的 learning rate 進行 training（其他參數需一致），作圖並且討論其收斂過程。

[設定]
Learning Rate: (0.001, 0.01, 0.1, 1, 10) ， 使用 adagrad
Lambda: 0，不使用 regularization
Initialization: 參數初始化成零向量
Number of Iterations: 200000
Features: 只使用每9小時內的PM2.5

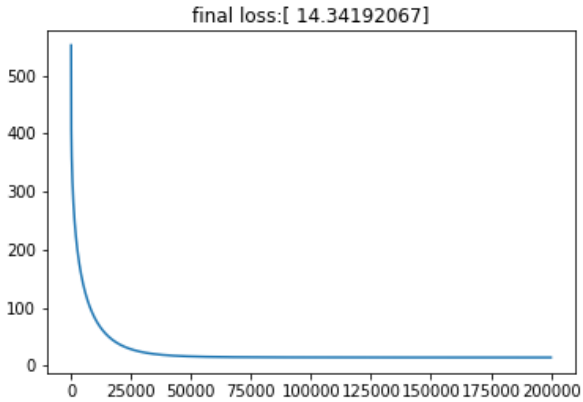
[Learning Rate = 0.001]



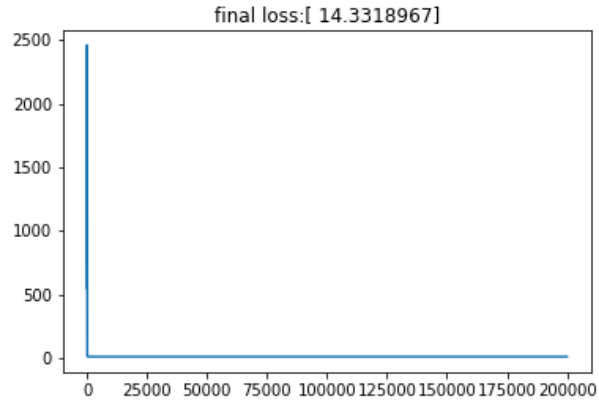
[Learning Rate = 0.01]



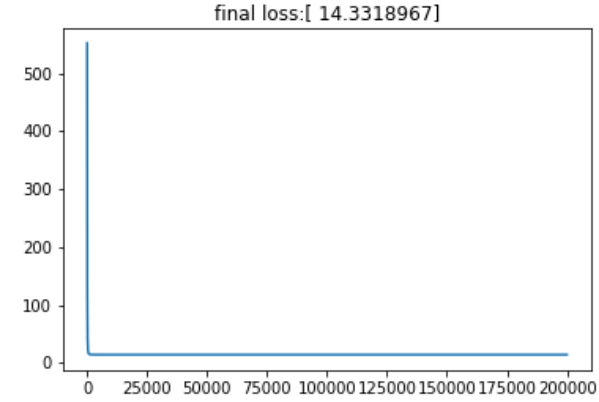
[Learning Rate = 0.1]



[Learning Rate = 1]



[Learning Rate = 10]



[討論]

由作圖結果可看出，learning rate 越大，loss 隨著 iteration 次數下降的速度也越快。當 learning rate 太小的時候 (0.001, 0.01)，在 200000 iterations 之後仍然沒有完全收斂，而當 learning rate 夠大的時候 (大於 0.1)，基本上 loss 收斂的速度很快，而且最終都收斂到 14.33 左右。由圖也可以看出來，learning rate 必須要適當挑選，否則有可能 train 很久卻還是得不到好結果。

3. (1%) 請分別使用至少四種不同數值的regulization parameter λ進行training（其他參數需一至），討論其root mean-square error（根據kaggle上的public/private score）。

[設定]

Learning Rate: 0.1 ，使用 adagrad

Lambda: (0.001, 0.1, 0, 1, 100, 10000)

Initialization: 參數初始化成零向量

Number of Iterations: 200000
Features: 只使用每9小時內的PM2.5

[Lambda = 0]
Score:8.99357

[Lambda = 0.001]
Score: 8.99355

[Lambda = 0.1]
Score: 8.99125

[Lambda = 1]
Score: 8.97074

[Lambda = 10]
Score: 8.80066

[Lambda = 100]
Score: 8.40825

[Lambda = 1000]
Score: 9.15462

[Lambda = 10000]
Score: 12.09427

[討論]
由實驗結果可以發現，隨著 regularization 的強度增加 (Lambda 變大)，Score 呈現先降後升，而在 Lambda = 100 這個數量級左右時，有較好的表現。推測是由於 regularization term 一開始確實有減少 overfitting 的狀況，所以 Score 有改善，然而後來隨著 regularization 變得太強，反而讓 model 不能很好地去 fit data，於是 Score 又再度變差。

4. (1%) 請這次作業你的best_hw1.sh是如何實作的？（e.g. 有無對Data做任何Preprocessing？Features的選用有無任何考量？訓練相關參數的選用有無任何依據？）

[設定]
Learning Rate: 0.01，使用 adagrad
Lambda: 100
Initialization: 使用 normal equation 初始化參數（不考慮 regularization），再用 gradient descent 調整 (考慮 regularization)
Number of Iterations: 2000
Features: 只使用 PM2.5 ，取每 9 個小時中的後5個小時

[實作]
選取 Feature：根據第一題的結果，發現其他的 feature 可能是干擾因素，因此決定只取看起來比較相關的來使用，然而試過的組合中，取 PM2.5 和 PM10 的效果較好，卻仍然沒有只取 PM2.5 來的好，因此最後決定只用 PM2.5 。
猜測：觀察 training set ，個人感覺其實 9個小時內 PM2.5 的變化比起一次線性回歸，可能更適合用二次或以上的多項式回歸，因此決定用更短的時間間隔來計算 (在較短時間內比較接近直線) ，最後決定取後4個小時去 training。
初步資料篩選：檢查 training data 時，有發現不少奇怪的資料，例如有些資料在短時間內變化過大、突然出現高達九百多的數值、連續的 0 或甚至出現負值。因此我決定先把這類我認為明顯不合理的資料排除。
刪除 outliers ：第一步篩選過後，我直接使用 normal equation (暫不考慮 regularization) 得到近似的 theta (參數) ，再根據此參數算出每一筆資料的 error ，發現有數筆資料的 error 特別大 (數千到數萬不等) ，因此我決定將這幾筆資料也排除掉。
Gradient descent with regularization：將資料做完我認為合理的前處理後，才真正用 gradient descent 去調整參數，這時才有考慮 regularization term。