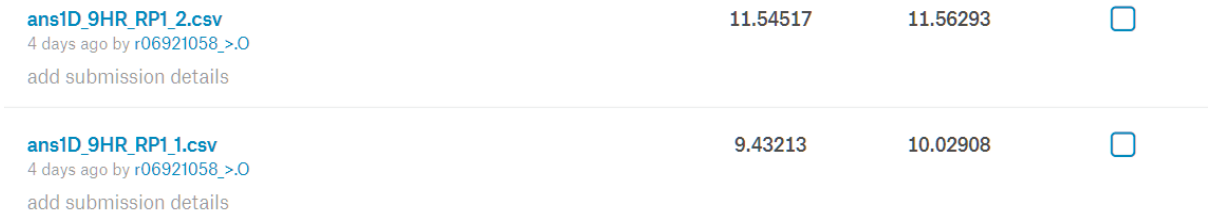
Homework 1 Report - PM2.5 Prediction

學號：r06921058 系級：電機碩一 姓名：方浩宇

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）。

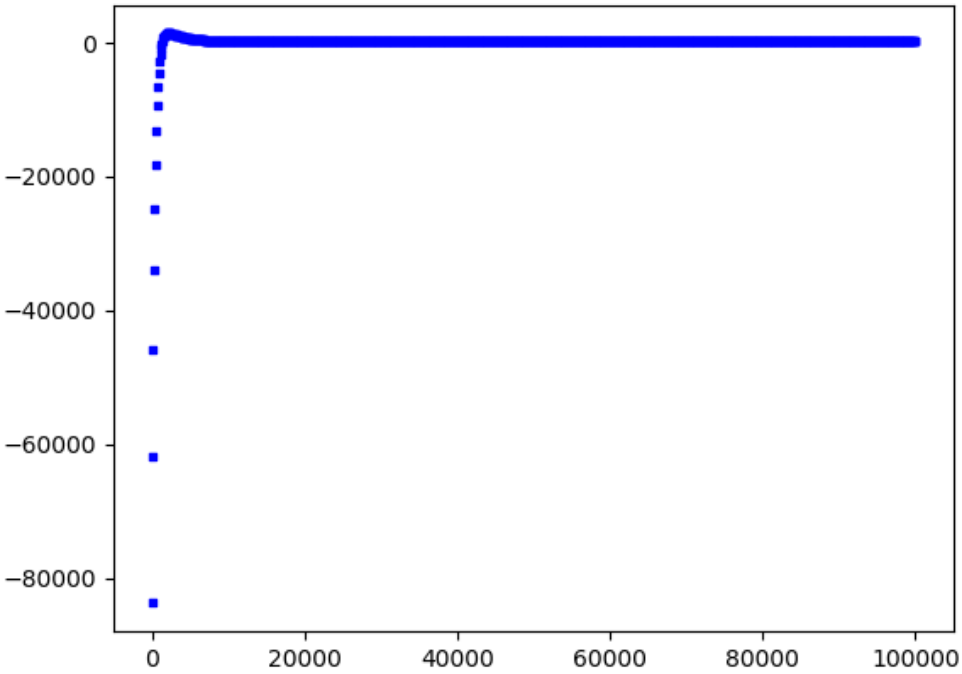


其中 RP1_1 為 Train 所有 feature 的一次項的結果，RP1_2 為 Train PM2.5 的一次項的結果。

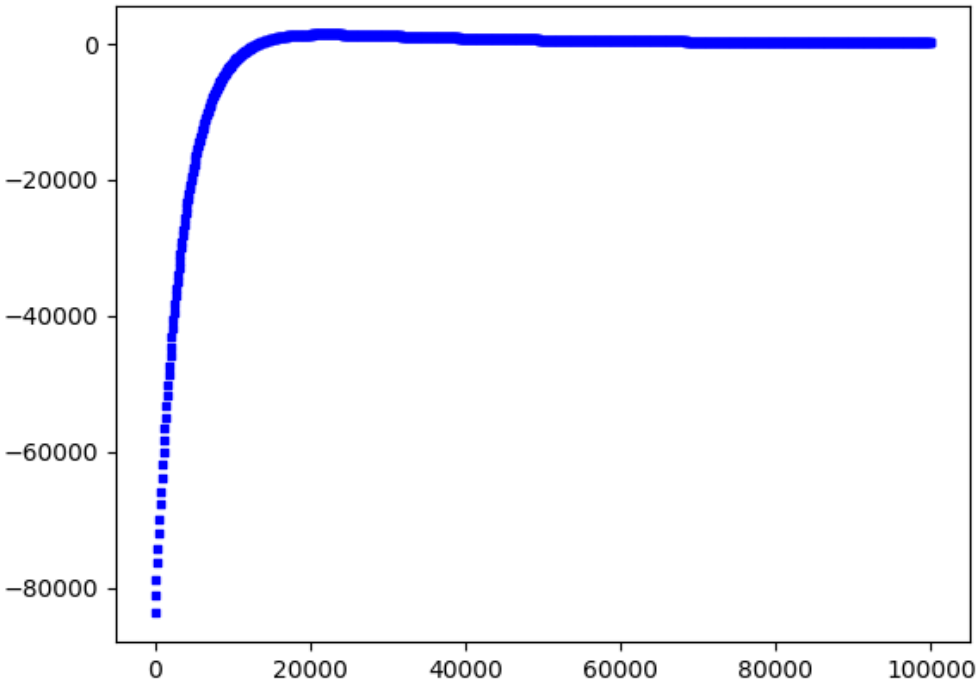
從 Kaggle 分數可以發現，其中 Train 所有 feature 的一次項的分數比 PM2.5 的分數高大約 1.5 左右，所以可以得出：用較多的資料進行 Train 會得到較好的結果，但是差距只有 1.5 也代表說，PM2.5 是其中與目標較為相關的 feature，所以說只用 PM2.5 所 Train 出的 Model 也能夠有一定的準確率。

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。

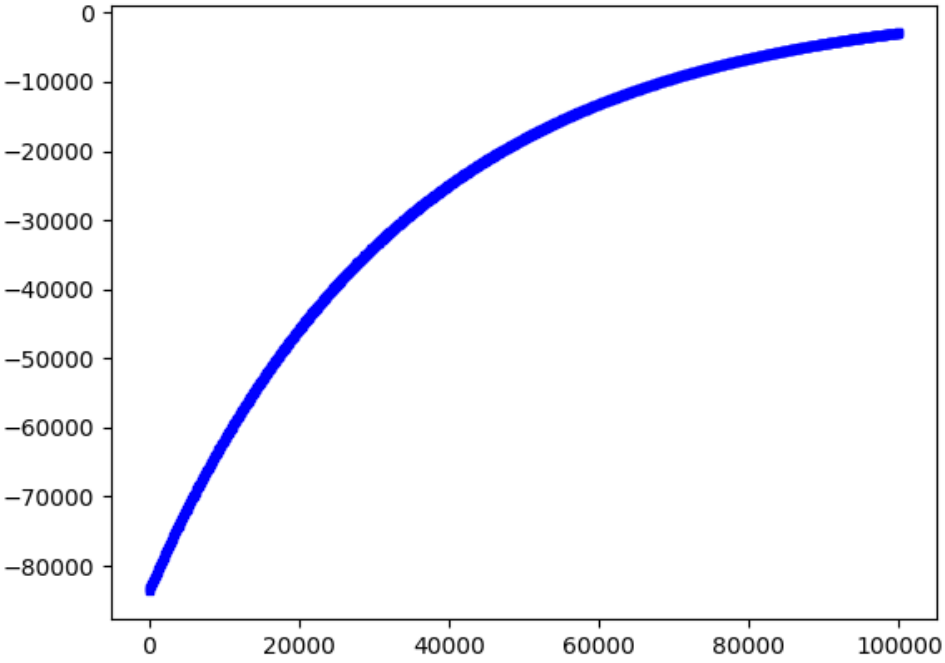
A.LearningRate=0.001



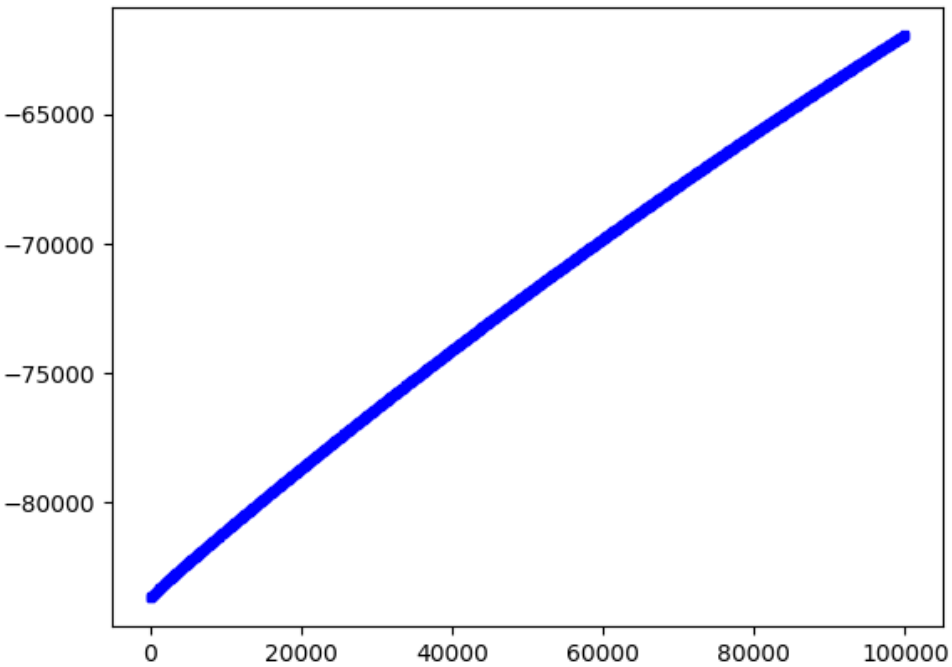
B.LearningRate=0.0001



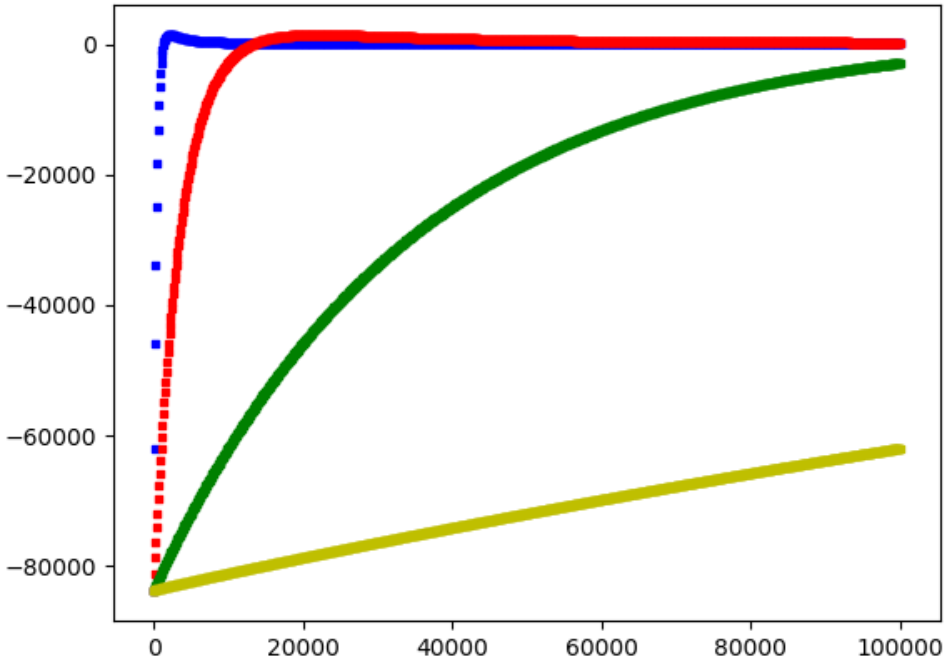
C.LearningRate=0.00001



D.LearningRate=0.000001



E.四種不同 LR 的比較(藍紅綠黃 LR 依序為 0.0001,0.00001,0.000001,0.0000001):



X 軸為執行步驟數量，Y 軸為誤差(Y^-Y 的總和)

從上圖可以發現，在同樣步驟數量(100000 步)的情況下，LR 越大的越快收斂，這是因為 LR 越大時，在一開始誤差較大時踏的步伐非常大，所以較快到達收斂的情況，然而，如果 LR 設定過大的話，可能會造成步伐太大而直接跑到另外一邊超出許多，結果因為根據誤差去設定步伐大小，造成下一步又跨更遠，循環幾次之後就會誤差就會超出最大值了。

這幾張圖中的誤差是採用每個 Y^-Y 的總和，所以也可以發現，通常都會從負非常多(大約十萬)往正的移動，最後超出一些(大約在正 500 左右)之後又會慢慢往 0 靠近。

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）。

Submission and Description	Private Score	Public Score
ans1D3HR_X2_norm_SelectMonth_Bias_Regu4.csv just now by r06921058_>.O add submission details	7.91492	9.02347
ans1D3HR_X2_norm_SelectMonth_Bias_Regu3.csv a few seconds ago by r06921058_>.O add submission details	7.92379	9.01438
ans1D3HR_X2_norm_SelectMonth_Bias_Regu2.csv a minute ago by r06921058_>.O add submission details	7.92485	9.01342
ans1D3HR_X2_norm_SelectMonth_Bias_Regu1.csv 2 minutes ago by r06921058_>.O	7.92475	9.01351

這四筆都是使用 0.001Learning Rate，跑 30 萬步，其中 Regu1 的 Regulation Param 是 0.1，Regu2 是 0.01，Regu3 是 1.0，Regu4 是 10，可以發現其實 4 筆的結果在 Public Score 的分數相差並不大，這應該是代表 30 萬步有充分的 fit 到 Training data，不過 Private Score 就可以發現，參數越大的在 Private 表現就越好，這也可能代表了這有發揮它的功用，就是讓曲線變得更平滑，避免 Overfitting。

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？）

這次作業我的 best_hw1.sh 的實作分成以下幾點討論:

A. Preprocessing:
   我使用 N-fold Validation，發現 3,4,9,12 四個月分的資料品質較為不好，因此我直接將這四個月的資料去除，而這個動作在 Kaggle 上的 Score 也有不小進步(大約 0.5 多)
B. Normalize:
   我是根據每個項目的大小，直接製作一個 Normalize Matrix 來負責處理 Normalize。參數為 30,2,1,0.5,50,50,50,55,50,50,1,80,3,3,350,10000000,5,3，其中 10000000 是為了將風向這個參數的影響降低。運算是 X=np.multiply(X,1/NormM)，其中 X 為 DATA。
C. Feature Select:
   在選擇 Feature 方面，除了前面提到的去除資料以及把方向去除之外，我還把試著用 3 小時、5 小時、9 小時三種去 Train，最後發現 3 小時的 Score 比其他高了大約 0.5 左右，因此採用 3 小時(每日 21 組資料)，並且將 54 項 feature 都加上 2 次項去 train(一共 108 項 feature，1+2 次項)，再加上額外的 Bias 一共 109 項參數
D. Gradient Descent:

在這部分我是設定 Learning Rate 為 0.0001，執行 100 萬次，沒有使用 Adagrad 之類的方法，原本有試著加過但是反而讓結果變差，因此乾脆直接採用一般的 Gradient Descent。在 train 時，和 train data 之間的誤差大約在 30 萬次左右時就達到最小值了，不過上傳的 Kaggle 分數卻是 100 萬次比較好，因此還是使用 100 萬次。