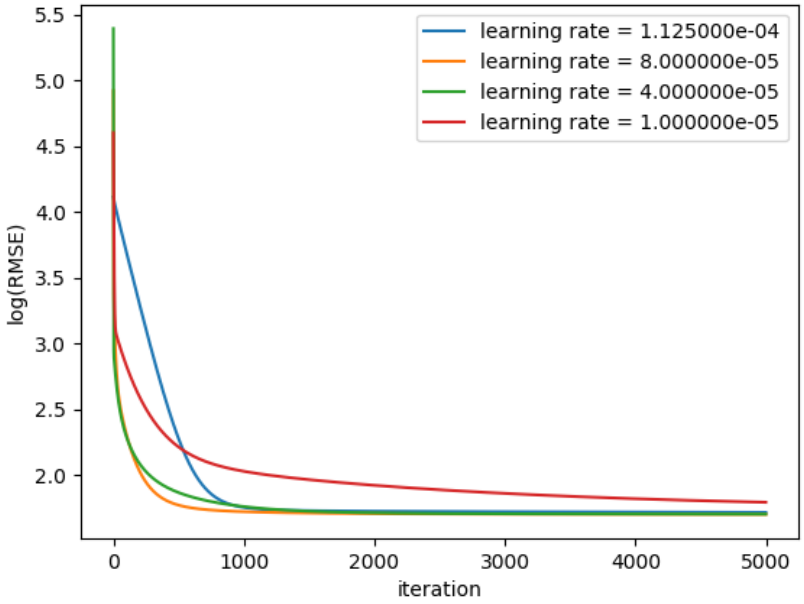
Homework 1 Report - PM2.5 Prediction

學號：b05902127　系級：資工二　姓名：劉俊緯

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）。
（完全沒有任何預先處理。）
9hr 只用 PM2.5: public score : 7.33332 private score: 8.49910
9hr 參數全下　: public score : 12.16007 private score: 12.10767
參數過多本身就容易造成 overfitting，因此造成 162 個參數下下去，不但難以收斂，更難拿的到好結果。有鑑於此，再做機器學習相關課題時，應該先觀察資料的關係，並取出好的 feature 來 train。

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。

由這四種 learning rate 我們可以知道：learning rate 太大太小其實沒有一定的相關性，因為 lr 太大可能會跳到另外一個波，而我們無法得知現在的 lr 叫做大還是小。



在這張圖我們可以看出來，lr = 8e-5 和 4e-5 做的差不多好，至於 lr 太大(1.125e-4)初期收斂較慢，可能是一直跳到山谷另一端，而 lr 太小(1e-5)看起來應該就是走太慢，慢慢滾到 minima，所以相較其他 lr 收斂的較慢。(y 軸是 log_2{RMSE(train)})

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）。
（有做 data cleaning 和 correcing.）
λ 0.1 :　private 21.25999 public :21.32376
λ 0.01 : private 12.60463 public :12.26473
λ 0.001 : private 9.78238 public　9.16728
λ 0.0001 : private 8.37864 public :7.60791
原本 regularization 的意義在於避免 overfitting。但是我拿去做的參數已經不會 overfitting 了。因此在這樣的條件下加上 regularization 也只是使原本的 model 無法發揮實力而已。換句話說，如果 overfitting 很嚴重，卻又不想拔掉一些參數，則 regularization 才有用。否則會做的更差。

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？）

Feature engineering: 透過比較 factor 之間的相關係數，得知 PM10 和 CO 與 PM2.5 的相關係數相較之下高(0.8/0.6)
Feature extracting: 使用九小時的 PM2.5 / PM10 的值。CO 因為加上去沒有顯著效果，所以不用了 QQ。
Traning data preprocessing: 我們知道 training data 的每個月前二十天是連在一起的。於是我們就相接起來，並取連續十小時作為 training data 的 X,Y。
Training data cleaning: 當 PM2.5/PM10 的值不介於(0,120]之間時，當作此 data 有問題，不採用並扣除。
Testing data correcting: 當 PM2.5/PM10 的值不介於(0,120]之間時，當作此 data 有問題，利用前後 interpolation 取代。如果是在頭或尾，則以最靠近的兩項取 mean。
Training data cleaning 2: 當 PM2.5/PM10 的值在連續時間下預測的 y 過於奇怪(當(min(x) -5 > y 或 max(x) + 5 < y) 且 y 和 x[-2]與 x[-1]的垂直距離>3 時)，當作這筆 data 有問題，也會被扣除掉。
No bias: 我認為以 PM2.5 預測 PM2.5 本身不該有 bias。有 bias 的情形應該在於平均不同的變數才會需要，因此我拔掉了 bias。實際就算加上 bias，算出來的值也會很小。
Linear regression formula: 其實 linear regression 根本就有公式，為 pseudo inverse(trainX) * trainY. 因此直接套公式可以省去不少時間。除非要做非線性函數再來搞 Gradient Descent 吧。
透過以上的工程，在 public 得到的結果為 6.14715，我覺得看起來還行啦。