學號：B06902066　系級：資工二　姓名：蔡秉辰

請實做以下兩種不同 feature 的模型，回答第 (1) ~ (3) 題：

1. 抽全部 9 小時內的污染源 feature 當作一次項(加 bias)
2. 抽全部 9 小時內 pm2.5 的一次項當作 feature(加 bias)

備註：

a. NR 請皆設為 0，其他的數值不要做任何更動

b. 所有 advanced 的 gradient descent 技術(如: adam, adagrad 等) 都是可以用的

c. 第 1-3 題請都以題目給訂的兩種 model 來回答

d. 同學可以先把 model 訓練好，kaggle 死線之後便可以無限上傳。

e. 根據助教時間的公式表示，(1) 代表 p = 9x18+1 而(2) 代表 p = 9*1+1

1. (2%)記錄誤差值 (RMSE)(根據 kaggle public+private 分數)，討論兩種 feature 的影響

	E_in	E_public	E_private	E_test
方法一	5.72052	5.64841	7.24674	6.49691
方法二	6.15304	5.90352	7.23005	6.60020

註：(1) 方法一為取全部作為 feature，方法二為僅取 pm2.5 作為 feature

(2) E_in: 表示做完 linear regression 後，training data 的 RMSE。

E_public/private: 表示 public/private data 的 RMSE。

E_test: 表示全部 testing data 的 RMSE(非前兩者相加除二)

(3) 參數如下：iteration=200000, eta(learning rate)=10。且採 adagram 之方法

討論：

(1) 因為方法一的維度較方法二高，故方法一的 E_in 會較方法二的小上許多，這也符合做出來之結果。

(2) 在 E_test 中，方法一的誤差雖較方法二小，但差距並沒有像 E_in 一樣差這麼多。可以推斷出有一點 over-fitting 的現象出現。

(3) 除(2)所提到的之外，方法一在 private data 中的 Error 甚至較方法二來得略大一點；且 E_public 和 E_private 相差較大。亦可由此推斷出使用 162+1(bias) 維是有一點點 overfit 了。

2. (1%)將 feature 從抽前 9 小時改成抽前 5 小時，討論其變化

	E_in	E_public	E_private	E_test
方法一_9hr	5.72052	5.64841	7.24674	6.49691
方法一_5hr	5.85305	5.98291	7.19183	6.61504

方法二_9hr	6.15304	5.90352	7.23005	6.60019
方法二_5hr	6.24960	6.22857	7.22794	6.74678

註：參數如下：iteration=200000, eta(learning rate)=10。且採 adagram 之方法

討論：

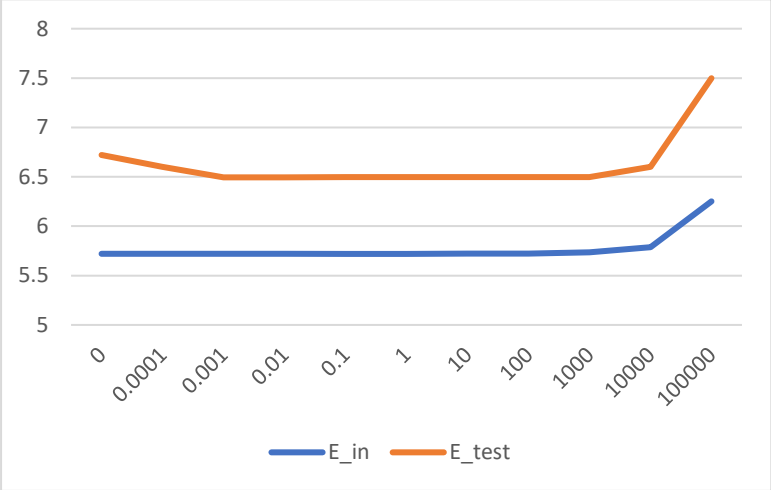
無論是方法一或是二，在將 feature 減少成抽 5 小時後，在 training data 和 testing data 的 Error 都更大了。我認為是因為只取 5 小時的數據並未使 E_in 和 E_test 的差距縮小很多，且 E_in 變大不少。故在 testing data 的誤差是較取 9 小時來得大。

3. (1%)Regularization on all the weight with λ=0.1、0.01、0.001、0.0001，並作圖

註：(1) 方法一固定參數為：iteration=200000, eta=10。且採 adagram 之方法

(2) 方法二固定參數為：iteration=100000, eta=10。且採 adagram 之方法

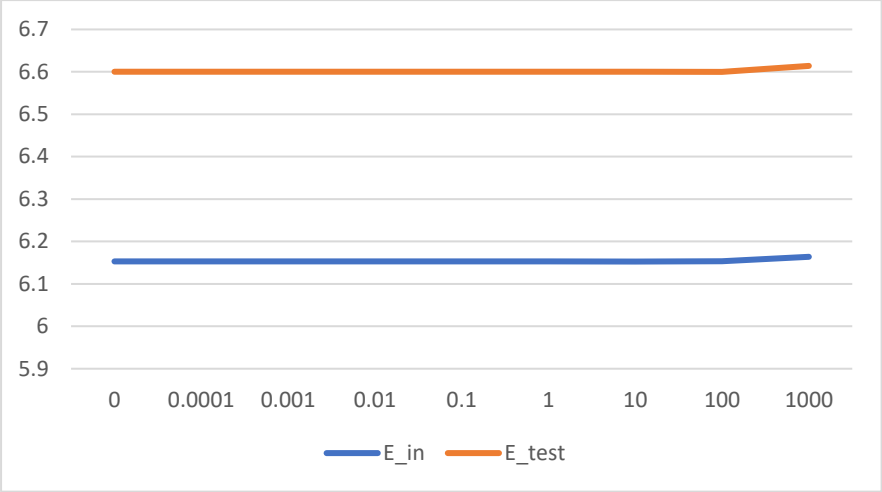
(1) 方法一，lambda 對 Error 之關係：(橫軸-lambda 值；縱軸-Error 大小)



討論：

方法一中，在 lambda 介於[0, 0.001]時，可以看出隨著 lambda 上升，E_test 有下降之趨勢，而 E_in 幾乎不變(其實 E_in 是有上升的，但因幅度太小，故這張圖看不太出來)。這也驗證了在前面所提，方法一的確是有一點 over-fitting，故在加入了 regularization 後，在 testing data 上有變好的趨勢。而為了更好做分析，故我多做了 lambda=1~10^5。可以看出，在 lambda 介於[0.001, 1000]時，E_in 一樣是增加但增加量很小，E_test 大約在(6.494, 6.499)這極小的區間震盪。到了 lambda 大於 1000 後，E_in 和 E_test 有明顯的增加，可以看出此時 lambda 已經太大，呈現 under-fitting。

(2) 方法二，lambda 對 Error 之關係：(橫軸-lambda 值；縱軸-Error 大小)

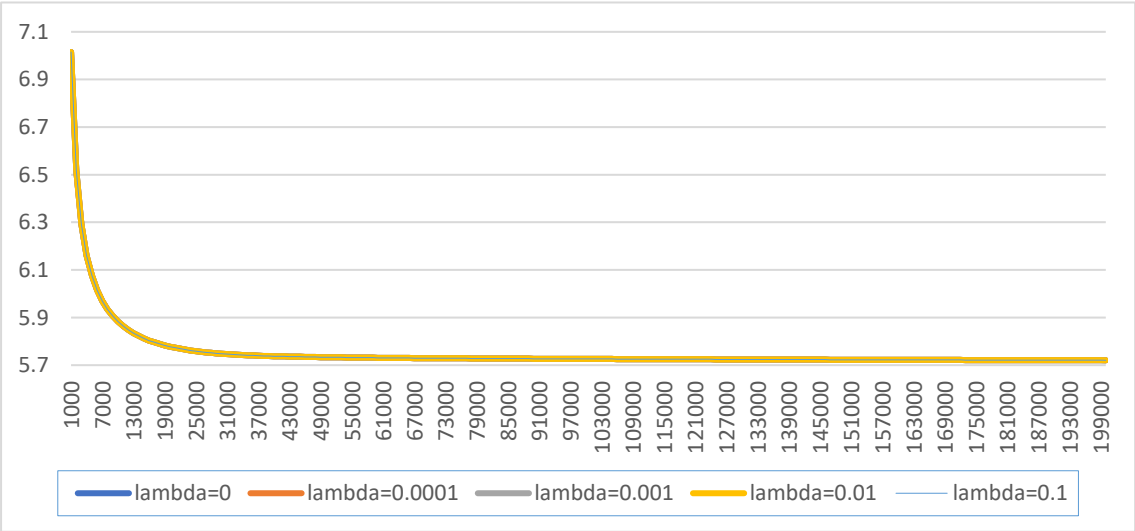


討論：

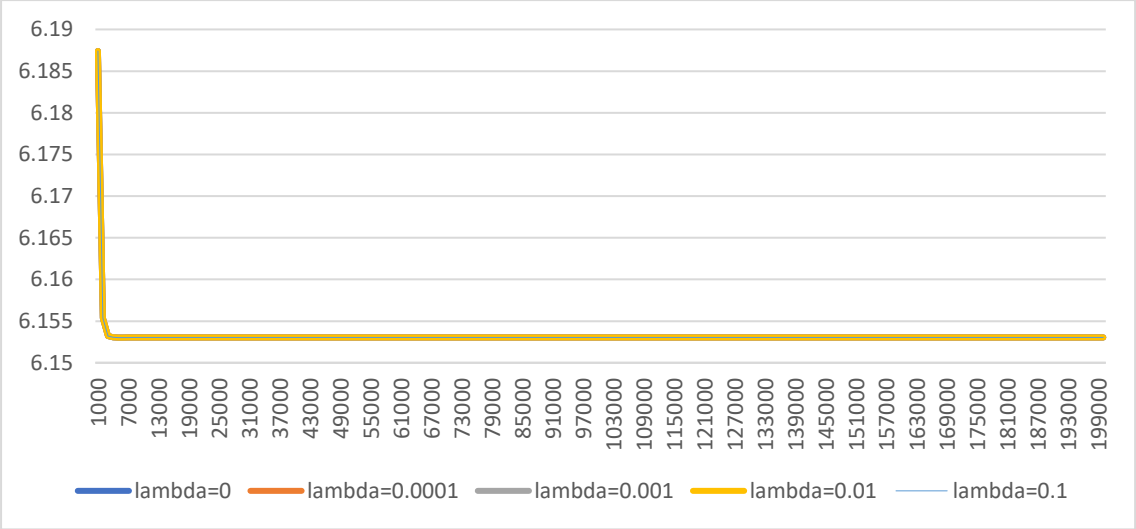
方法二中，無論是 E_in 或是 E_test，在題目所給定之範圍作出之結果幾乎相同。Regularization 本身的意義就是要使我們在做 gradient descent 時，走的距離不要那麼大，且無論 iteration 放多大，最終一定會跟最低點有些微距離。而因為方法二沒有 over-fitting，且 dimension 只有 10 維。故在 lambda 值很小(<100)的時候基本上影響相當的小。在額外做了 lambda=1, 10, 100, 1000 後，可以看到在 1000 時，才會因為距離最低點已有一定的距離使 E_in 和 E_test 提高。可以預期若再提高，便會有 under-fitting 的發生。

(3) 不同 lambda 下，iteration 對 E_in 的變化：

(i) 方法一：(橫軸-iteration 次數；縱軸-E_in)



(ii) 方法二：(橫軸-iteration 次數；縱軸-E_in)



討論：
無論是方法一或是二，lambda 在[0, 0,1]區間時，雖然增加 lambda 會使 E_in 上升，但幅度並不大，故作圖出來後感覺幾乎是重疊的。

4. (1%)在線性回歸問題中，假設有 N 筆訓練資料，每筆訓練資料的特徵(feature) 為一向量 x^n，其標註(label)為一純量 y^n，模型參數為一向量 w (此處忽略偏權值 b)，則線性回歸的損失函數(loss function)為 n=1Nyn-xnw2 。若將所有訓練資料的特徵值以矩陣 $X = [x^1\ x^2 \dots x^N]^T$ 表示，所有訓練資料的標註以向量 $y = [y^1\ y^2 \dots y^N]^T$表示，請問如何以 X 和 y 表示可以最小化損失函數的向量 w ？請選出正確答案。(其中 X^TX 為 invertible)

a. $(X^TX)X^Ty$
b. $(X^TX)yX^T$
c. $(X^TX)^{-1}X^Ty$
d. $(X^TX)^{-1}yX^T$

Ans: C