學號：B0490lll7 系級： 電機三 姓名：毛弘仁

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 的一次項(加 bias)
(2) 抽全部 9 小時內 pm2.5 的一次項當作 feature(加 bias)

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

(1) Private error 6.73403, public error 10.48996, 平均 8.611995。我設定的參數為 learning rate 10^{-5}，太大的話無法 converge，太小的話 train 很慢（我忙著找 feature，先不用 optimizer，照理來說最後會因為 convex 特性而 converge）。Batch size 似乎在 20 的時候最好。

(2) Private error 5.93563, public error 7.65508, 平均 6.795355。設定的參數為 learning rate 10^{-4}。Batch size 似乎在 10 的時候最好。

抽所有 features 時，learning rate 需要設定成比較小，可能因為變數較多而 loss 也容易產生較大幅度的變化。如預期，所有的 feature 需要的 training time 比較長。實驗中，只參考 PM2.5 的 model 比參考所有 model 來得準。觀察到 private error 比 public error 高，看來 private set 跟 public set 的分佈應該是有所不同。

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

(1) Private error 6.96088, public error 8.04365, 平均 7.502265。平均 error 較 9 小時的 model 低，train 速度提升，蠻不錯的。但需要重新 tune hyperparameters，batch 現在改成 30 較佳。error 還是沒有單獨 9 小時 PM2.5 當 feature 好。

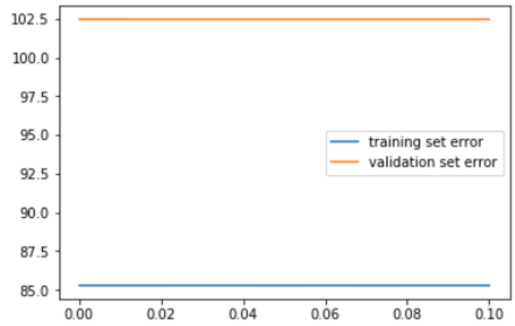
(2) Private error 6.02114, public error 7.18664, 平均 6.60389。平均 error 一樣會下降，train 速度也提升。Hyperparam 沒改變 。

可發現將 9 小時改為 5 小時，兩種 model 準確度均有所提升，看來時間點近的數據較有預測能力。在這邊也持續觀察到 private error 比 public error 高，增強 private set 跟 public set 分佈不同的想法。

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

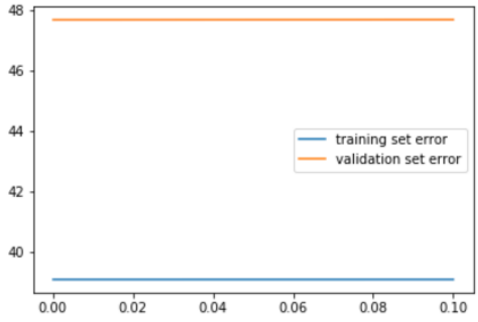
全部 feature 都是一次方，好像沒有顯著效果，不過可以根據 regularized weights，看出哪些 feature 對 model 判斷比較有幫助。

```
average test error 102.451483209
average train error 85.2933035522
average test error 102.454188206
average train error 85.2957322945
average test error 102.454459
average train error 85.295975506
average test error 102.454486082
average train error 85.2959998305
```



所有 features (9hr)

```
average test error 47.6733889449
average train error 39.0982811789
average test error 47.6690100773
average train error 39.0976086586
average test error 47.6686454195
average train error 39.0976182101
average test error 47.6686096965
average train error 39.0976199441
```

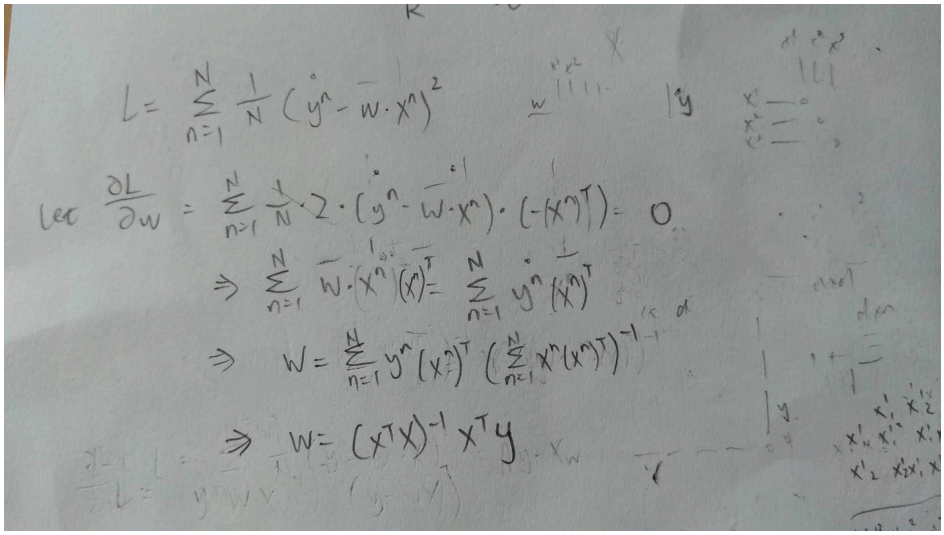


只有 PM2.5 (9hr)

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 $\mathbf{x}^n$，其標註(label)為一存量 $\mathbf{y}^n$，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$\sum_{n=1}^{N}(y^n - w \cdot x^n)^2$。若將所有訓練資料的特徵值以矩陣 $\mathbf{X} = [\mathbf{x}^1\ x^2 \ldots x^N]$ 表示，所有訓練資料的標註以向量 $\mathbf{y} = [\mathbf{y}^1\ y^2 \ldots y^N]^T$ 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 ？請寫下算式並選出正確答案。

(a) $(X^TX)X^Ty$
(b) $(X^TX)^{-0}X^Ty$
(c) $(X^TX)^{-1}X^Ty$ ㄊ
(d) $(X^TX)^{-2}X^Ty$

答案是 (c) $(X^TX)^{-1}X^Ty$



個人心得：

我用 closed form 跑出來的 test error 不一定有 trained linear regression model 來得好，推測可能是 closed form 在 training data 上 overfit。不過大致上來說，跟 feedforward network（我自己另外實作）比起來，closed form 的 error 不差，代表 linear assumption 應該還算有些合理。

我使用「每個月前 6 天」（佔全部 train.csv 20%）的資料當作 validation set，想說這樣的分佈應該和 test set 的「每個月後 10 天」差不多。在這種假設下，一定想辦法讓 validation error 最低(反正不是在它身上 train)，但我遇到 validation error 變低，test error 變高的怪異情況，代表我的 validation set 和 test set 有差距。可以嘗試 cross validation,但訓練時間會變成 5 倍，而且不敢保證那五組 validation set 就會跟 test set 很像。另一個方法是隨機抽 validation set，這個方法我下次可能會實作看看。

本次作業讓我獲益良多，學會很多重要的訓練技巧，也讓我深刻體會理論與實務的差距。查了一些資料，希望以後對於 model training 會有更好的成果!