

1.

- 環境變數設定值：
  - a. Regularization weight = 0.1
  - b. #Iteration = 50000
  - c. 12-Fold
- 實作了以下兩種模型：
  - a. 用前 9 小時的所有污染源當 feature
  - b. 用前 9 小時的 PM2.5 當 feature

- 使用演算法：Adagrad

下表整理此兩種模型上傳到 Kaggle 後得到的分數各為多少

Model	Kaggle private	Kaggle public	Training time
9 小時的所有污染源	5.28843	7.51501	~8min
9 小時的 PM2.5	5.29770	7.42451	<1min

- 使用演算法：Adam(相關參數參考 google tensorflow)

下表整理此兩種模型上傳到 Kaggle 後得到的分數各為多少

Model	Kaggle private	Kaggle public	Training time
9 小時的所有污染源	5.46875	7.53407	~10min
9 小時的 PM2.5	5.64346	7.48308	<1min

- 結論

若以 Adagrad 實作 Gradient descent，其平均的誤差與訓練時間都比 Adam 的表現來的好。在 Public 的測資上不管是 Adarad 還是 Adam，只使用前 9 個小時的 PM2.5 來當 feature 表現都較使用前 9 個小時的所有污染物佳，在 Private 的測資上剛好相反，使用前 9 個小時的所有污染源來當 feature 表現較佳

2.

- 環境變數設定值：
  - a. Regularization weight = 0.1
  - b. #Iteration = 50000
  - c. 12-Fold

- 實作了以下四種模型：
  - a. 用前 9 小時的所有污染源當 feature
  - b. 用前 9 小時的 PM2.5 當 feature
  - c. 用前 5 小時的所有污染源當 feature
  - d. 用前 5 小時的 PM2.5 當 feature
- 使用演算法：Adagrad

下表整理此四種模型上傳到 Kaggle 後得到的分數各為多少

Model	Kaggle private	Kaggle public	Training time
9 小時的所有污染源	5.28843	7.51501	~8min
9 小時的 PM2.5	5.29770	7.42451	<1min
5 小時的所有污染源	5.54888	7.52083	~8min
5 小時的 PM2.5	5.61191	7.45807	<1min

- 使用演算法：Adam(相關參數參考 google tensorflow)

下表整理此四種模型上傳到 Kaggle 後得到的分數各為多少

Model	Kaggle private	Kaggle public	Training time
9 小時的所有污染源	5.46875	7.53407	~10min
9 小時的 PM2.5	5.64346	7.48308	<1min
5 小時的所有污染源	5.44382	7.69456	~10min
5 小時的 PM2.5	5.79082	7.5954	<1min

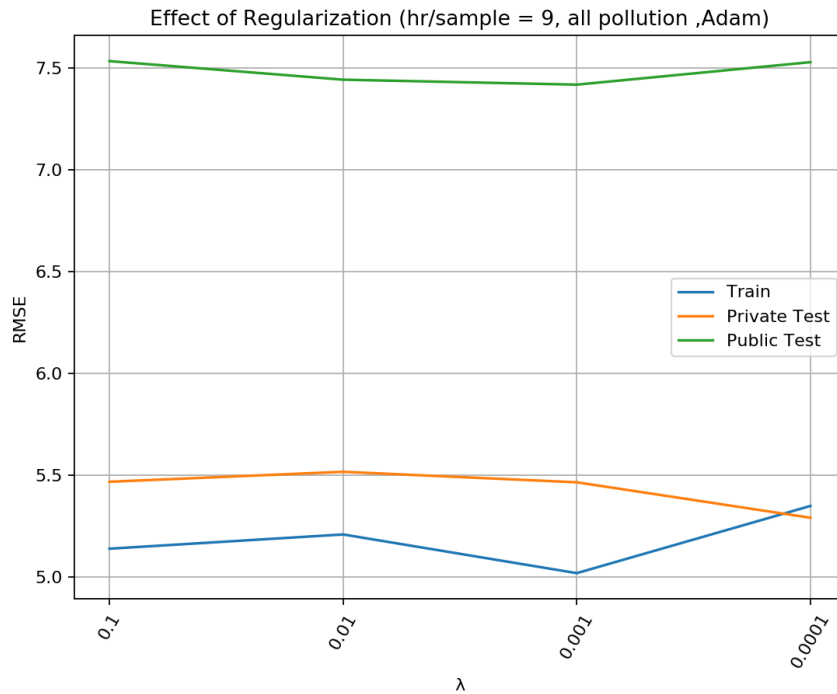
- 結論

若以 Adagrad 實作 Gradient descent，其平均的誤差與訓練時間都比 Adam 的表現來的好。在 Public 或 Private 的測資上不管是 Adagrad 還是 Adam，只使用前 9 個小時的資料來當 feature 表現都較使用前 5 個小時佳。

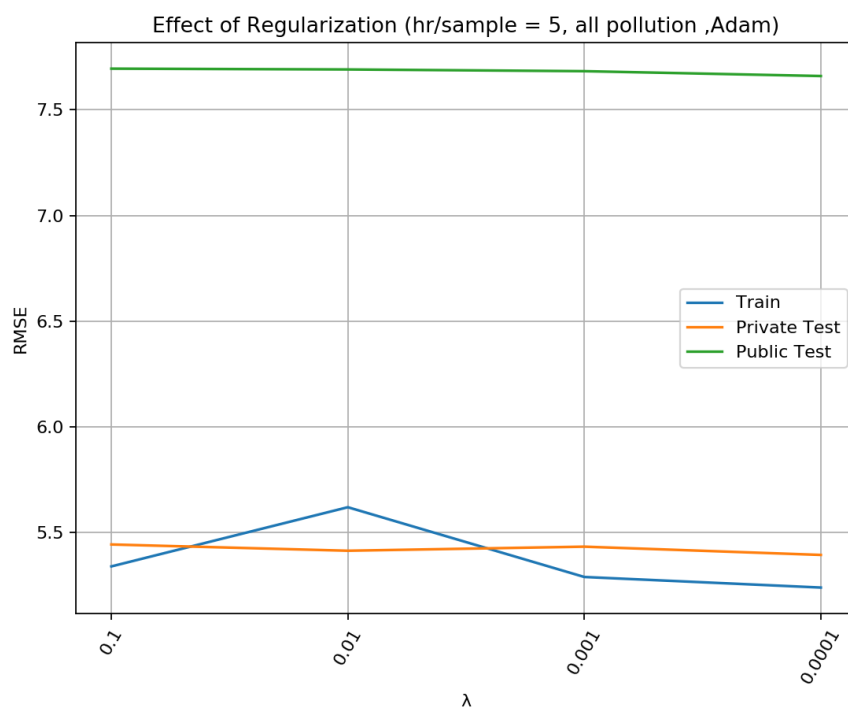
3.

為求討論簡潔，以下我都使用 Adam 演算法來實作 Gradient descent

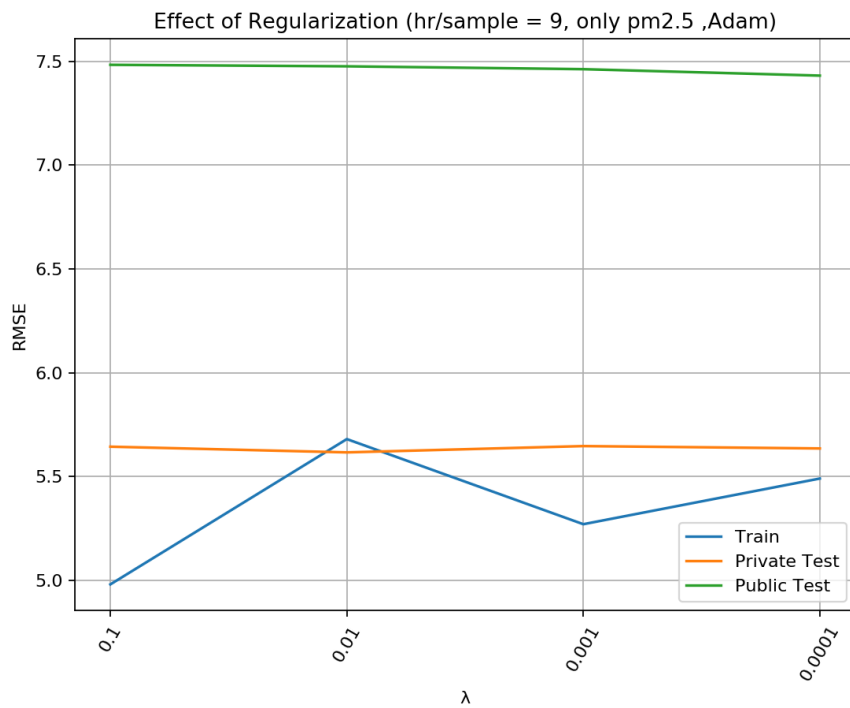
- 用前 9 個小時所有污染物當 feature 建模



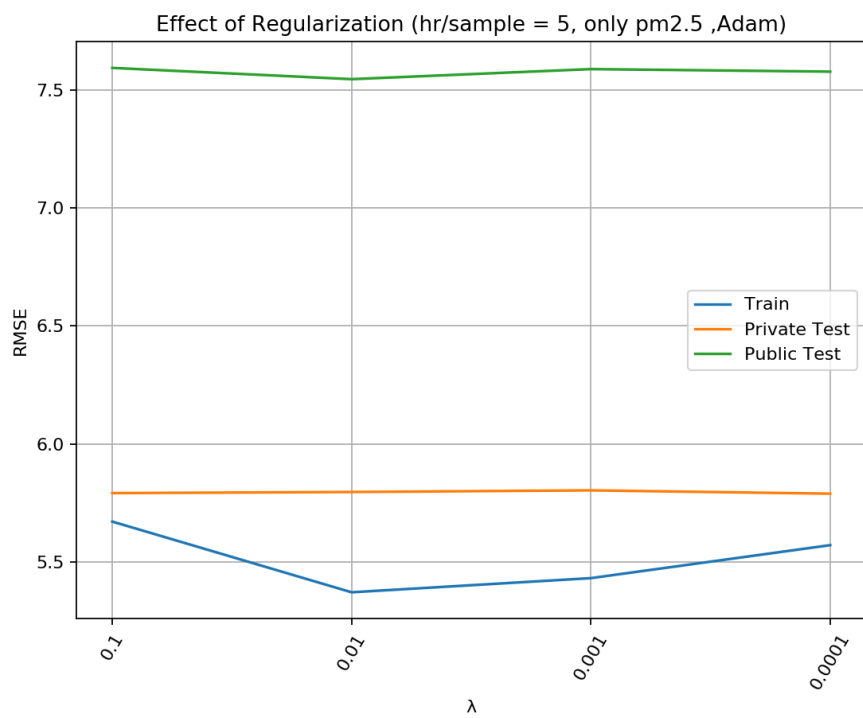
- 用前 5 個小時所有污染物當 feature 建模



- 用前 9 個小時的 PM2.5 當 feature 建模



- 用前 5 個小時的 PM2.5 當 feature 建模



4.

此問題可以先從求解一 Normal equation 著手，如以下

$$Xw = Y$$

考慮誤差  $e$  的存在，方程式如以下

$$Xw + e = Y$$

則誤差  $e$  可以表示為：

$$e = Y - Xw$$

今考慮有  $n$  個樣本點，則 loss function 可以表示為：

$$L = (Y - Xw)^T(Y - Xw)$$

今透過 Gradient descent 的方法來求找 loss function 的極小值：

$$\frac{\partial L}{\partial w} = 2X^T(Y - Xw) = 0$$

求解過程如下：

(1)兩邊同乘 $X^T$

$$X^T Xw = X^T y$$

(2)因為 $X^T X$ 反矩陣必存在，則兩邊同乘 $(X^T X)^{-1}$

$$(X^T X)^{-1} X^T Xw = (X^T X)^{-1} X^T y$$

(3)整理得

$$w = (X^T X)^{-1} X^T y$$

Ans:(c)