Homework 1 Report - PM2.5 Prediction

學號：b03901109　系級：電機四　姓名：陳緯哲

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）。

	所有 feature	PM2.5 PM10 O3 NOx NO2 NO	PM2.5
Public score	8.23	6.01	8.75
Private score	7.89	6.31	8.40
Training Loss	4.63	4.73	5.25

如果我們只使用 PM2.5 來進行 Training，儘管他的 Training Loss 看起來不錯，但 Test 的結果卻十分糟糕，但他參數太少，因此我覺得這應該不是 OverFitting，而是過少的參數讓前處理的效果(請見第 4 題)沒有顯現出來，就算對 test data 也進行了一樣的前處理，依舊沒辦法將 PM2.5 的特性顯現出來。

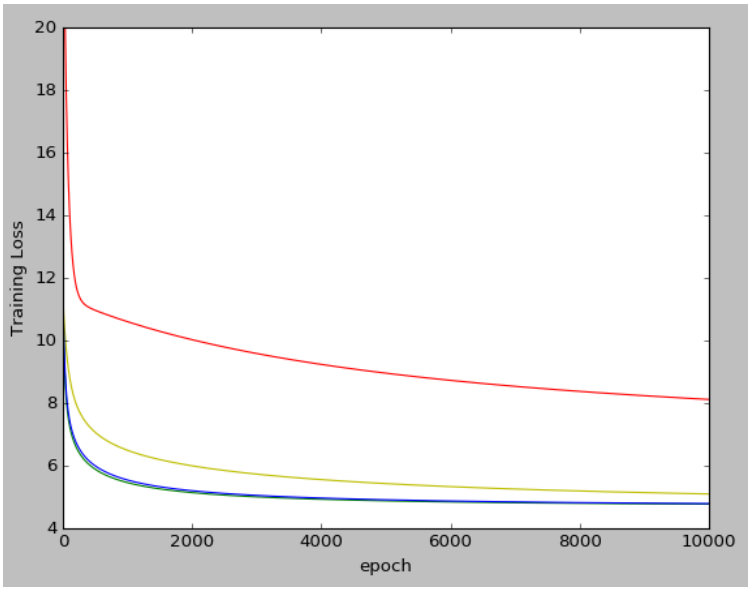
而如果採用了所有 Feature 並進行 Normalization，則會導致 Overfitting 的問題，有以下可能兩種原因

(1) 資料的前處理：當資料遇到連續的 0 時，前處理會將不採用這些資料，而這個方式採用了所有資料，勢必得進行標準化，而此時的標準化也會讓前處理失去效果，無法找出那些無效的資料。

(2) 資料的特性：在觀測站所測得的所有資料，不見得都跟 PM2.5 的趨勢有很大的關係，而為了 fit training data，這些資料可能會讓 PM2.5 的預測失準

綜合以上兩者，在取用 training data 時，必須使用數量足夠多且必須跟 PM2.5 有關的參數，我在實作此次作業時，便是使用 PM2.5、PM10、O3、NOx、NO2、NO 等參數進行訓練，這些參數與 PM2.5 作圖就算直接用人眼看出其趨勢的相關，因此拿來進行訓練所得出的結果十分的不錯。

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。

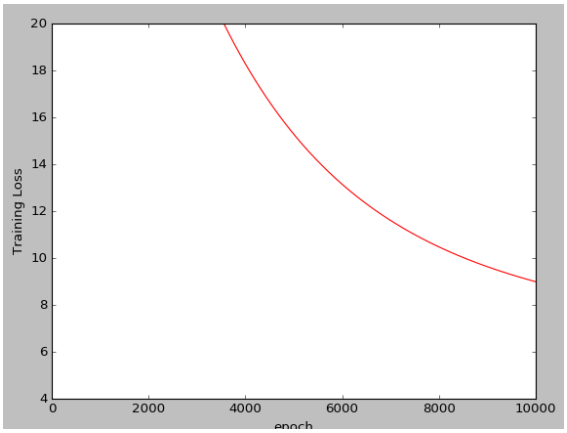


右圖為針對不同 Learning rate 所訓練出的 training error，以下說明各顏色所代表之 Learning Rate：

1. 紅色：0.001
2. 黃色：0.01
3. 綠色：0.1
4. 藍色：1

隨著 Learning Rate 的增加，可以看到 Training Loss 的收斂變快，在第 2000 次 epoch 的時候，豪瑟的 Training Loss 在 10 左右，黃色的則約下降到 6，而綠色與藍色皆已收斂到 5，說明了 Learning Rate 越大，其收斂速度越快；但 Learning Rate 的增加會產生飽和的效果，也就是儘管一樣是增加 10 倍的 Learning Rate，其收斂速度卻幾乎不會改善，如上圖的藍線與綠線，兩者的 Learning Rate 相差了 10 倍，但兩線幾乎是重合的。

這是不是說明了 Learning Rate 越大越好，不管再怎麼大頂多也只是飽和呢？其實不是的，左圖的 Learning Rate 是 10^17，為了方便觀察收斂速度，因此將他以同樣的量尺呈現。他的收斂速度十分的緩慢，直到第 10000 次 epoch 左右才降到 10，因此若 Learning Rate 太大，也會讓訓練的結果變得更差，甚至找不到合適的 Minimum Valley



3. (1%) 請分別使用至少四種不同數值的 regulization parameter λ 進行 training（其他參數需一致），討論其 root mean-square error（根據 kaggle 上的 public/private score）。

	0	10	100	1000	10000	100000	1000000
Public	6.01	6.02	6.02	6.05	6.32	7.56	10.20
Private	6.31	6.31	6.31	6.32	6.43	7.29	9.86
Training	4.73	4.73	4.74	4.74	4.89	5.83	7.89

當 λ 於 0-1000 之間時，對 Training 與 Test 的影響並不會很大，儘管有小幅度的改善 Test 的結果，但其改善幅度約在小數點後 3 位，實在是以此無法認定 Regularization 的效果，而 λ 大於 1000 後，便會使 Loss 逐漸增加。

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？）

本次作業是以手刻 Linear Regression 進行實作，Epoch = 120000，Learning Rate = 1，並使用 Adagrad 的演算法，選用了 PM2.5、PM10、O3、NOx、NO2、NO 作為 Feature 進行 Training，之所以是使用這幾個參數主要是因為我利用 excel 對資料作圖，發現以上幾個參數的趨勢與 PM2.5 的趨勢十分相像，因此使用了這幾個參數。

而選用好以上幾個參數後，會發現以下兩件事情：

(1) 資料常常有一大段的 0

(2) 一筆為 0 的資料，他前後資料的大小十分奇怪，如 100、0、102

發現以上兩件事情後，我便到網路上查詢相關資料，發現觀測站所測得的資料，有時可能也是無效的，因此我進行 Data 的 Preprocessing，只要有任何以每 9 小時為一組的 Feature，內有 PM2.5 小於 0 的資料，便刪除這個 Feature；而當 Test Data 遇到小於 0 的資料時，由於不能像 Training data 一樣捨棄，只能用內插法希望能重現此筆無效的資料。經過測試後 Test Score 下降了許多，說明了這樣的處理是必須的；

另外一件值得一提的是，由於 Test 的上傳次數有限，因此必須實作 Data Validation 讓自己在 Local 端也能進行測試，我將 Training Data 分為 10 份，並選出一份作為 Test Data，其他的資料

作為 Training Data；過程中，當我選用第 3 份 data 作為 test 的時候，與其他 validation dataset 相比，Training Loss 下降了許多，Test score 卻是上升，於是我更細部的找出這些 error 是哪些資料所導致的，將範圍縮小到 200 筆後，將這些資料刪除，成功的讓我的 Test Score 從 8.4 下降到 6.3，後來仔細檢視這些資料，這些資料都異常的大，極有可能是無效的，因此將他們刪除應該是正確的選擇。