Machine Learning HW1

學號:B06902060　系級:資工三　姓名:鄒宗霖

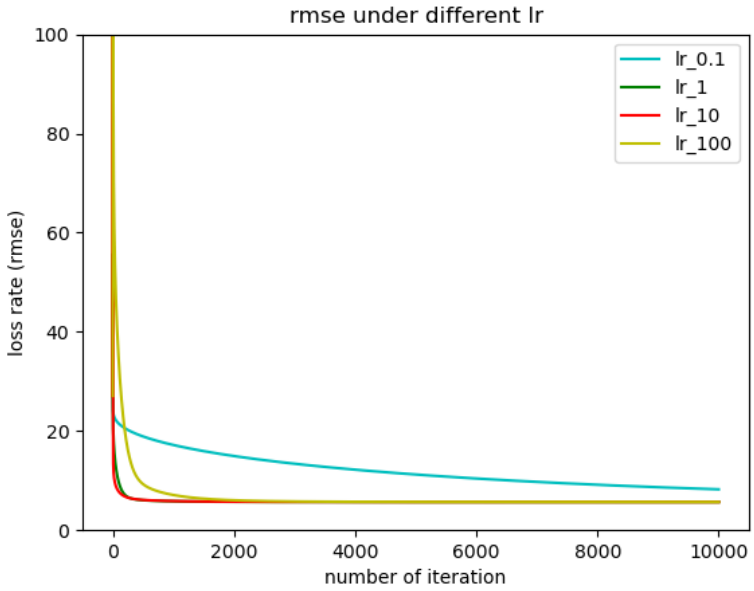
備註:

a. 1~3 題的回答中，NR 請皆設為 0，其他的數值不要做任何更動。

b. 可以使用所有 advanced 的 gradient descent 技術 (如 Adam、Adagrad)。

c. 1~3 題請用 linear regression 的方法進行討論作答。

1. (2%) 使用四種不同的 learning rate 進行 training (其他參數需一致)，作圖並討論其收斂過程 (橫軸為 iteration 次數，縱軸為 loss 的大小，四種 learning rate 的收斂線請以不同顏色呈現在一張圖裡做比較)。



我使用四種不同的 learning rate 分別是 0.1, 1, 10, 100，依照上圖來看，lr = 0.1 的時候每次更新參數的幅度太小了，導致收斂速度極慢；lr = 1, lr = 10 為比較好的 learning rate，更新參數的幅度適中，收斂速度快；lr = 100 的時候更新參數的幅度太大了，導致參數容易走過頭，不過我們使用 1/t decay 的技術，隨著時間的增加降低 learning rate，使得 lr = 100 最後也收斂至合理的 loss rate。

2. (1%) 比較取前 5 小時和前 9 小時的資料 (5 * 18 + 1 vs. 9 * 18 + 1) 在 validation set 上預測的結果，並說明造成的可能原因 (1. 因為 testing set 預測結果要上傳 Kaggle 後才能得知，所以在報告中並不要求同學們呈現 testing set 的結果，至於什麼是 validation set 請參考：https://youtu.be/D_S6y0Jm6dQ?t=1949 2. 9hr : 取前 9 小時預測第 10 小時的 PM2.5；5hr : 在前面的那些 features 中，以 5~9hr 預測第 10 小時的 PM2.5。這樣兩者在相同的 validation set 比例下，會有一樣筆數的資料)。

Models	iter_50	iter_100	iter_5000	iter_10000
5hr	7.75558	7.02752	5.67478	5.67469
9hr	22.91924	12.80692	5.66564	5.66504

上表為取前五小時和取前九小時的 Models 在不同 #iteration 時在 validation set 上的 loss rate (rmse)，我們可以看到 5hr Model 收斂的速度比較快，猜想可能是 feature 參數量比較少，在 (5 * 18 + 1) 維空間內找出最低點要比在 (9 * 18 + 1) 維空間內找出最低點容易。而在 iteration_10000 的時候，5hr Model 雖然有著比較高的 loss rate，但是差距頗小，表示只取前五個小時的 feature 並沒有明顯的 underfit。

3. (1%) 比較只取前 9 小時的 PM2.5 和取所有前 9 小時的 features (9 * 1 + 1 vs. 9 * 18 + 1) 在 validation set 上預測的結果，並說明造成的可能原因。

Models	iter_50	iter_100	iter_5000	iter_10000
PM2.5	6.74735	6.31558	5.86109	5.86109
all_features	22.91924	12.80692	5.66564	5.66504

上表為只取 PM2.5 和取所有 features 的 Models 在不同 #iteration 時在 validation set 上的 loss rate (rmse)，我們可以看到只取 PM2.5 Model 收斂的速度比較快，如上題所猜想的，可能是 feature 參數量比較少，在 (9 * 1 + 1) 維空間內找出最低點要比在 (9 * 18 + 1) 維空間內找出最低點容易。而在 iteration_10000 時，只取 PM2.5 Model 雖然有著比較高的 loss rate，但是差距頗小，表示取所有 features Model 裡面有些 feature 可能是不重要的，所以整體表現沒有比只取 PM2.5 Model 好很多。

4. (2%) 請說明你超越 baseline 的 model (最後選擇在 Kaggle 上提交的) 是如何實作的 (如 : 怎麼進行 feature selection，有沒有做 pre-processing、learning rate 的調整、advanced gradient descent 技術、不同的 model 等等)。

第一步就是先調整 learning rate，因為過大的 learning rate 更新參數的幅度太大了，導致參數容易走過頭，於是我讓 lr = 1，更新參數的幅度不會太大也不會太小，loss rate 蠻快就收斂了；第二步就是增加 iteration，因為 iteration 如果太小可能參數都還沒有收斂就停止更新參數了，於是我讓 iteration = 5000；第三步就是試著把一些參數拿掉，在 train.csv 裡找出一些看起來和 PM2.5 沒有甚麼關係的參數，在 raw_data 裡把他們的值改成 0 (或是直接把那些 features np.delete 掉，loss rate 結果一樣)。

曾經嘗試過減少 features 數目 (如取前 x 小時的所有 features、取前 x 小時的部分 features : x < 9)、增加 features 的二次項、以及對 loss function 實作 regularization，發現反而增高了 loss rate；曾經把 iteration 調大，但在 testing data set 的 loss rate 不降反增 (但只增加一點點)，推測是因為 training data set 和 testing data set 分布不太一樣，增加 iteration 來 fit training data set，反而導致 Model 偏離 testing data set。