說明：請各位使用此 template 撰寫 report，如果想要用其他排版模式也請註明題號以及題目內容（請勿擅自更改題號），最後上傳至 GitHub 前，請務必轉成 PDF 檔，並且命名為 report.pdf，否則將不予計分。

中英文皆可，但助教強烈建議使用中文。

----------閱讀完以上文字請刪除----------

學號：b06502158 系級：機械三 姓名：陳柏元

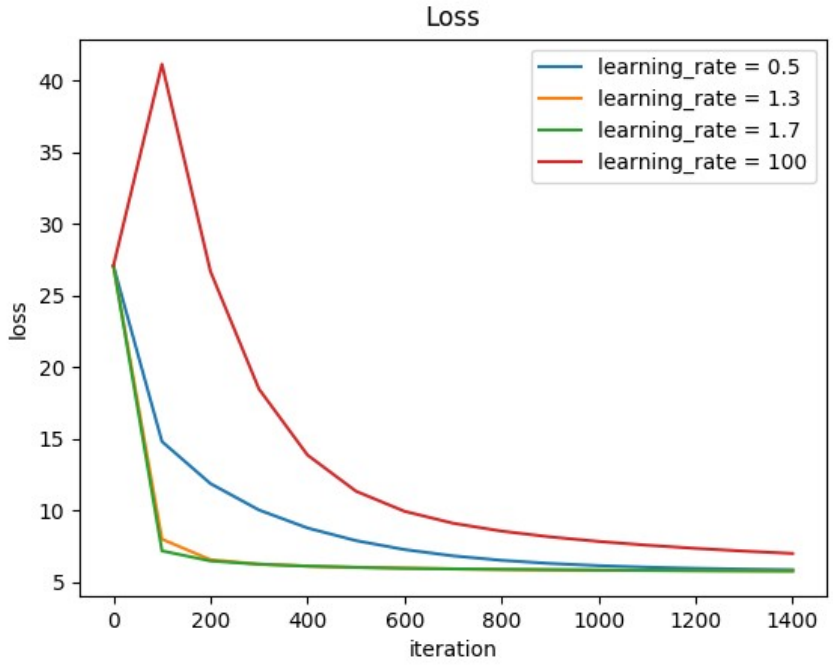
備註：

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。

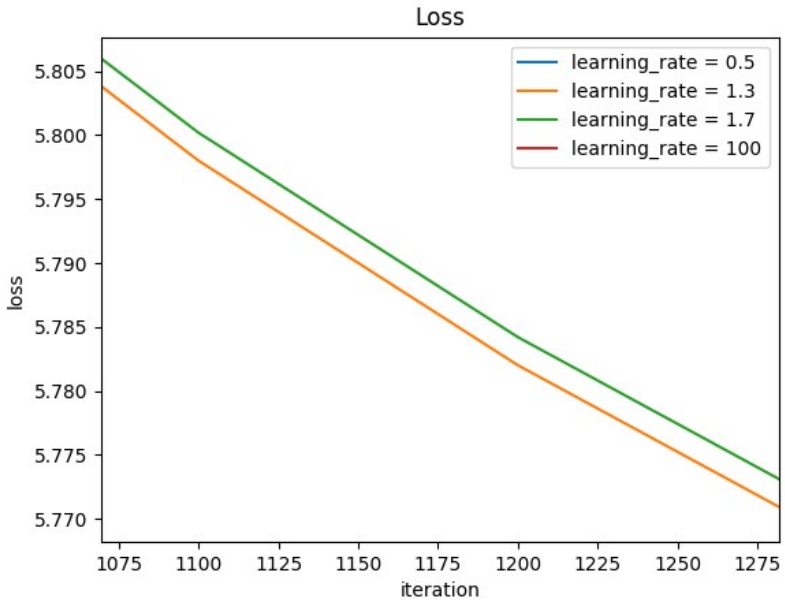
b. 可以使用所有 advanced 的 gradient descent 技術（如 Adam、Adagrad）。

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）。



圖一



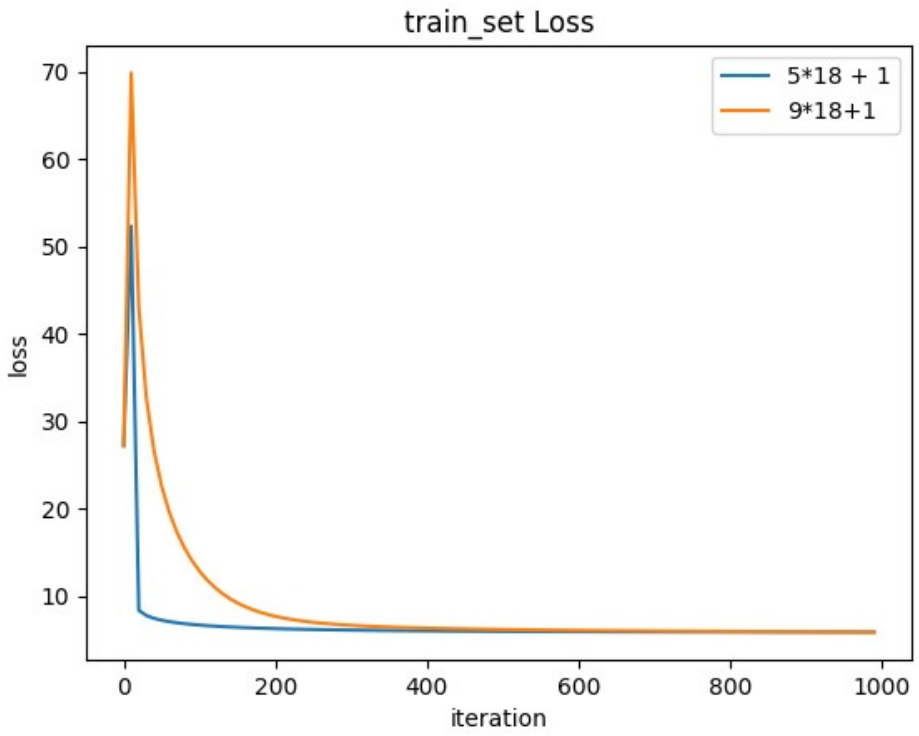
圖二

由圖一可以發現，當 learning rate 較低時，初始 loss 下降的斜率，會下降的比較慢，如圖一藍色線（lr=0.5），而較高 learning rate（lr=1.7）如圖一綠色線，則下降較快。但若將 learning rate 設置太大，如圖一中紅色線，則將經歷先升後降的過程，導致需要較長的 iteration 才能達平衡，且最後趨於平衡的 loss 也將遠高於其他 learning rate。而從圖二可觀察到，當 iteration 達到較高次數時，原本在 iteration=200 附近，loss 較高的橘色線（lr=0.5），在 iteration=1200 附近時，變得低於綠色線（lr=1.7），代表當 learning rate 較低時，初始 loss 下降的斜率，會下降的比較慢，但當 iteration 較高時，learning rate 較大的可能最低的 loss 因為 gradient decent 的降幅太大導致無法達到較低值。

2. (1%) 比較取前 5 hrs 和前 9 hrs 的資料（5*18 + 1 v.s 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）。

learning rate = 50

Feature	5*18+1	9*18+1
Train_set loss	5.905132081866303	5.916218109257459
Validation_set loss	5.752615085408606	5.891695188465846



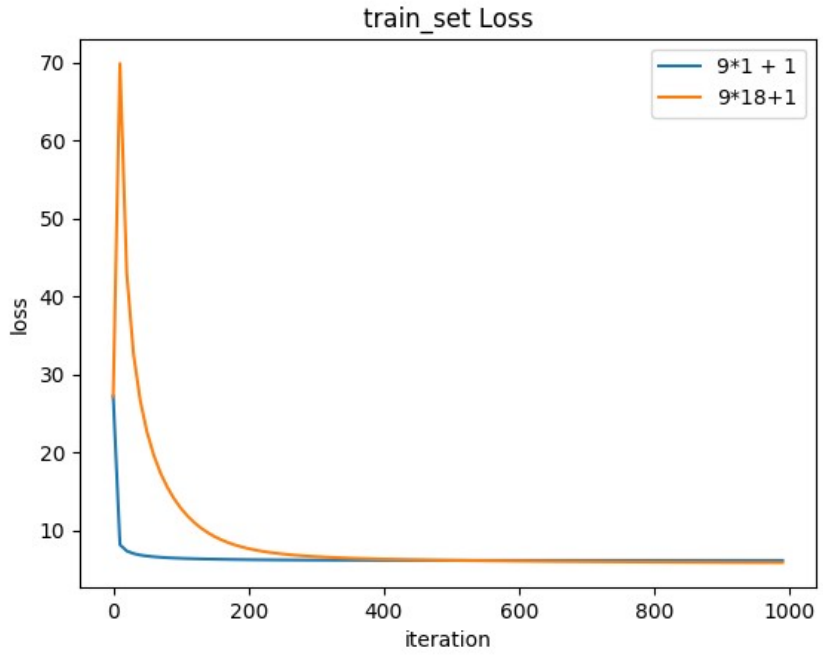
圖三

由上圖可以觀察到，當 feature 減少時，train_set 的 loss 兩者最後趨近的值差不多，但 iteration 於 0 至 200 附近時 feature 數 5×18+1 明顯下降較快。可能原因應該是 train feature 減少，在多維度的運算時 gradient decent 較快。而 validation set 觀察到的結果來看，loss 5×18+1 明顯低於 9×18+1。可能原因應該是 9×18+1 的維度過多 導致產生了 overfitting ，而 5×18+1 的 validation loss 則反之，故較低。

3. (1%) 比較只取前 9 hrs 的 PM2.5 和取所有前 9 hrs 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。

learning rate = 50

Feature	9*1+1	9*18+1
Train_set loss	6.1942119836159115	5.916218109257459
Validation_set loss	5.864396779303341	5.891695188465846



圖四

由上圖可以觀察到，當 feature 減少時，train_set 的 loss 兩者最後趨近的值差不多， 但 iteration 於 0 至 200 附近時 feature 數 9×1+1 明顯下降較快 ，可能原因與上圖相同是 train feature 減少 在多維度的運算時 gradient decent 較快，但不同的是 train set 的 loss 最後會因 9×1+1 的 feature 數太少導致最後平穩值 loss 較高。而 validation set 觀察到的結果來看 loss 9×1+1 還是低於 9×18+1。可能原因應該是 9×18+1 的維度過多，導致產生了 overfitting，而 9×1+1 的 validation loss 則較低，但可觀察到，與上題比較後發現，因只挑了 pm2.5 這個 feature，導致 loss 較 5×18+1 高呈現 underfitting 的結果。

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（例如：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）。

實作上我選擇了優化 linear regression 來突破 baseline，

首先呢我先進行了 feature 的挑選 ，我將 18 個 feature，挑選了 16 組出來除，去除了 rainfall，WIND_SPEED，而當中我將 feature WD_HR 及 WIND_DIERCT 乘上 cos 以確立此站的風向在東北季風上有較大的值，而非較不具意義的角度值，而因幾番測試後，發現兩者依然有較大的影響力，故新增兩者之二次式以來估算 weight，新增較少二次 feature 是因為在前幾題例子中以及實作發發現，若 feature 量過多，只會增加 overfitting 並不會降低 validation 的 loss 而 learning rate 則發現在這些 feature 的條件下約略 35 能達到最低 Iteration 則以 8000 能趨近 loss 最低點。