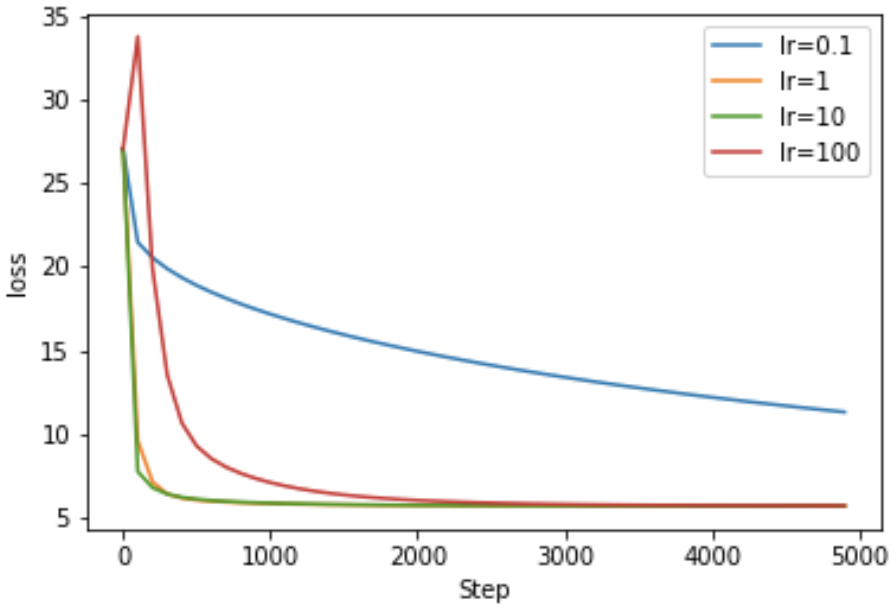
學號：B05902010　系級：資工四　　姓名：張頌平

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）。



Settings:
Total steps: 5000, save loss every 100 steps

如上圖所示可以發現，當初始learning rate等於1和10的時候，收斂速度較快，而等於10又再快一點點，learning rate等於100時，前期有較大震盪，原因應為learning rate過大，因此在尋找loss最低點時步伐過大，導致前期無法收斂，而後隨著step增加，learning rate也有隨之變小的情況下，就得以順利收斂，當learning rate等於0.1時，因為learning rate過小，導致需要收斂的時間過長，在我們設定的step內尚未收斂。

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）。

Settings:
Total steps: 5000, learning rate: 1

取前9小時:
Train loss: 5.72260
Validation loss: 5.66583

取前5小時:
Train loss: 5.86309
Validation loss: 5.67285

從兩者的loss可以發現，取前9小時的資料在train loss的部分好上許多，其原因可能為擁有的參數較多(多了4hr*18 = 72個feature)，因此在訓練的時候可以更擬合於我們的data，但在validation loss的部份兩者就相差很少了，前9小時還是好一些些，這說明了前6~9小時的資料在真實預測中的重要性可能沒有那麼大，如果當運算量很大的時候，就可以思考多餘的運算量所花的時間是不是值得提升非常小的準確率或loss。

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。

Settings:
Total steps: 5000, learning rate: 1, 取前 9 hrs data

所有features:
Train loss: 5.72260
Validation loss: 5.66583

只取PM2.5:
Train loss: 6.19403
Validation loss: 5.86119

所有features所預測的結果在train和validation的loss，都比只用PM2.5預測來得好，而且好上許多，我們知道前幾小時的PM2.5絕對是對於預測下一小時的PM2.5很重要的特徵，但從這個實驗，也能得知，其他特徵在前幾小時的數值一樣能有助於我們預測下一小時的PM2.5。

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）。

- 首先我做了一些 data pre-processing，像是**刪掉*pm2.5*等於*-1*的*data***、**刪掉前後小時所有特徵完全一致的*data***(應該是資料有缺漏，不然不可能前後兩小時所有特徵完全一樣)、**刪掉PM10=0的data**(大部分為突然變0，推測是資料錯誤)
- featrue selection的部分，我的方法是將**和*PM2.5*相關性比較低的特徵**刪掉，然後再去比對其loss有沒有真的比較好，依據此來進行特徵選擇或刪除。此外，我有增加一個新feature - 和**前一小時的*PM2.5*變化量**，這讓我的分數在validation set和public test set都進步了大概0.05的分數，在最後上傳的結果中，我還有加上**前一小時的*O3*變化量**，在我自己的validation set 和 public test set 都有很微小的進步
- learning rate 我設定在***0.8~1***左右，有做regularization，lambda設為***1~5***之間
- 在經過比較後，我只取了**前*7*小時**的data進行訓練，也因為只取了前7小時的data，我可以**利用部分*test data*來進行訓練**，這幫助我在validation set 和public test set 進步了0.01的分數
- 有使用NN來進行訓練，但可能是因為資料量不夠，模型的預測和loss很不穩定，因此最後模型還是選擇使用原本的linear regression
- 在public test set上，發現**刪除*PM10***會讓分數大幅躍進，但在我自己的validation set測試時發現結果反倒退步，推測是overfitting了，而且因為test的資料量其實很少(只有240筆)，所以很有可能會和原本實際資料有很大的bias，因此在最後選擇模型上，並沒有選擇刪除PM10的結果(儘管他在public test set 上分數最高)

Reproduce result (public dataset)：

hw1.sh: 5.48128
hw1_best.sh: 5.32568

How to train my code:

hw1_train.py: python3 train.py [train.csv位置]
hw1_best_train.py: python3 train.py [train.csv位置][test.csv位置]