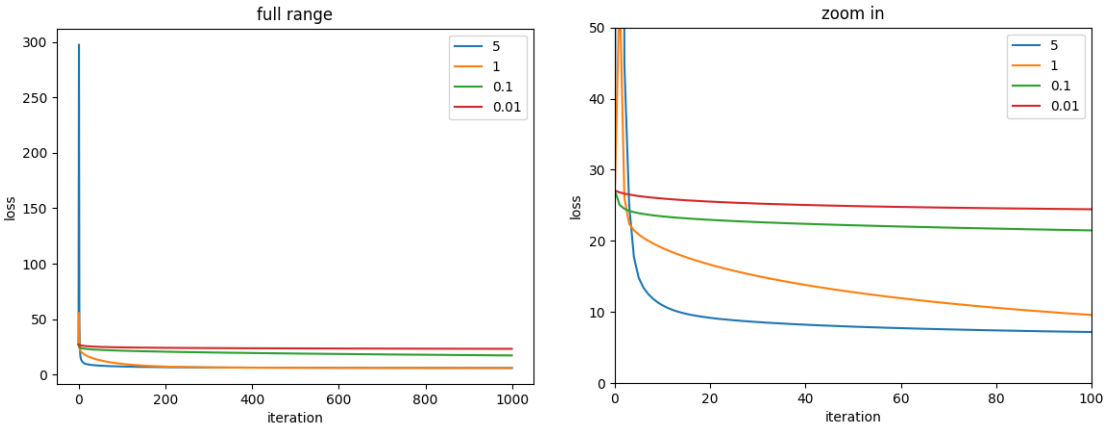
學號：B06901066　　系級：電機三　姓名：孟妍

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）。



這裡社的四種learning rate分別是(5,1,0.1,0.01)，設比較大的時候一開始會收斂的比較快，也有一個現象就是大一點的learning rate一開始的loss大約是27.x，但一開始會突然暴增再將回來。較小的learning rate沒有此現象，收斂比較緩慢。

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）。

（這邊的loss是validation）

取前9小時(loss: 5.91220546628651)

```
0:27.239591682144457
100:22.723975744599546
200:10.427772652266519
300:7.635824772204668
400:6.849878685598302
500:6.518567463859378
600:6.32564308053946
700:6.1943518687643575
800:6.09908116190811
900:6.027838242824769
loss:  5.91220546628651
Yende-MacBook-Pro:hw1 yenmeng$
```

取前5小時(loss: 5.7537144915068925)

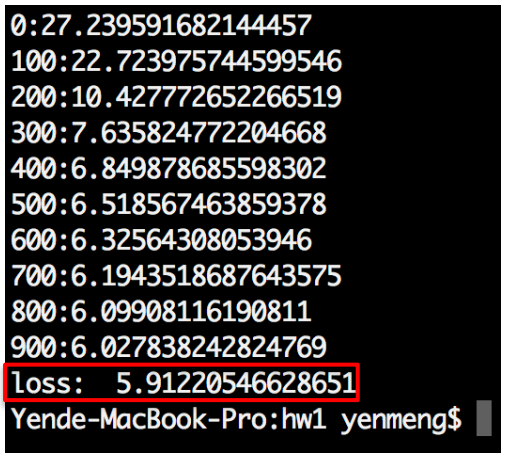
```
0:27.239591682144457
100:6.769146376178991
200:6.318289527592378
300:6.144478318308482
400:6.054277686471182
500:6.000403922058336
600:5.965544643193107
700:5.941882069664843
800:5.925330200419585
900:5.913518104032259
loss:  5.7537144915068925
Yende-MacBook-Pro:hw1 yenmeng$
```

比較取前 5 hrs 和前 9 hrs 的資料，只取前 5 hrs的資料在validation set的loss較小。
推測可能因為在資料量不是那麼大的狀況下，用過長的時間訓練會影反而效果會比用接近一點的、短一點的時間訓練來得差。前五小時比較接近要預測的第十小時，所以可能相關度較高，而前四小時資訊可能相對來說沒那麼重要。在這裡因為其他參數不能動，所以造成這樣的結果也可能和其他參數相關，例如iteration、learning rate等，像從上圖可以看到只取前 5 hrs的一開始收斂較快，所以iteration不夠多的情況下只取前 5 hrs的loss的確會比較小。

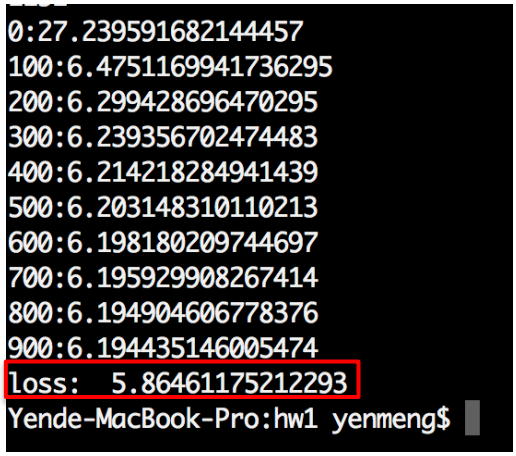
3. (1%) 比較只取前 9 hrs 的 PM2.5 和取所有前 9 hrs 的 features（9*1 + 1 vs. 9*18 + 1）在 validation set上預測的結果，並說明造成的可能原因。

（這邊的loss是validation）

all features (loss: 5.91220546628651)



only PM2.5 (loss: 5.86461175212293)



比較只取前 9 hrs 的 PM2.5 和取所有前 9 hrs 的 features，只取前 9 hrs 的 PM2.5的loss較小。這題的原因跟前一題可能有點類似，在資料量不夠大、iteration不夠多次的狀況下，太多feature會像雜訊一樣造成干擾，所以只保留PM2.5這個target feature反而可以比較準確預測。

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）。

一開始把learning rate調小，iteration調大。有對PM2.5的row做pre-processing，因為發現裡面有一些值是-1，代表沒量到，所以我就把這些-1的地方填上該row的mean，loss就有小很多。另外，在training的時候我把7月份的資料拿掉，因為發現有一些不太合理的值，loss也有因此變小。feature selection的部分時做了兩種版本，一種是有抽掉PM10，另一

種沒有，有抽掉PM10的版本是抽掉['RAINFALL','PM10','WS_HR']，另一種是抽掉['RAINFALL','WIND_DIREC','NO2','THC']。兩種都有過public strong baseline，抽掉PM10的效果非常好，但按常理來說PM10跟PM2.5應該是蠻相關的（？）所以不太確定是否已經有overfitting的現象。