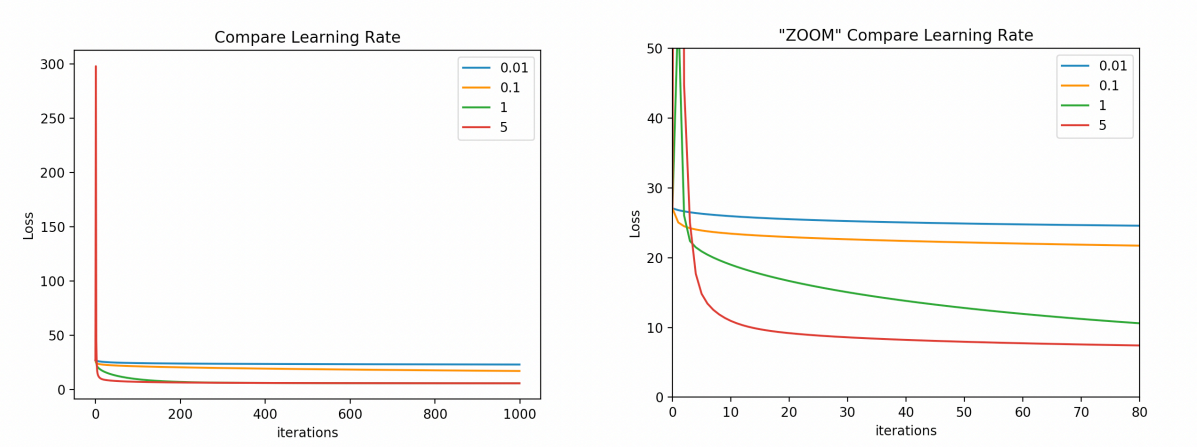
Machine Learning HW1

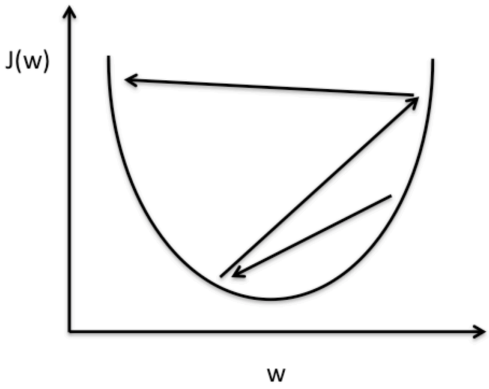
B06902135 資工三 蔡宜倫

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）。



- 由上圖可看出，learning rate 越小收斂得越慢；相反的，learning rate 越大收斂得越快，而且一開始因為w會被改變得比較多，可能會讓loss驟增而overshoot：雖然知道要往哪個方向走但因為learning rate過大導致走過頭，如下圖所示：



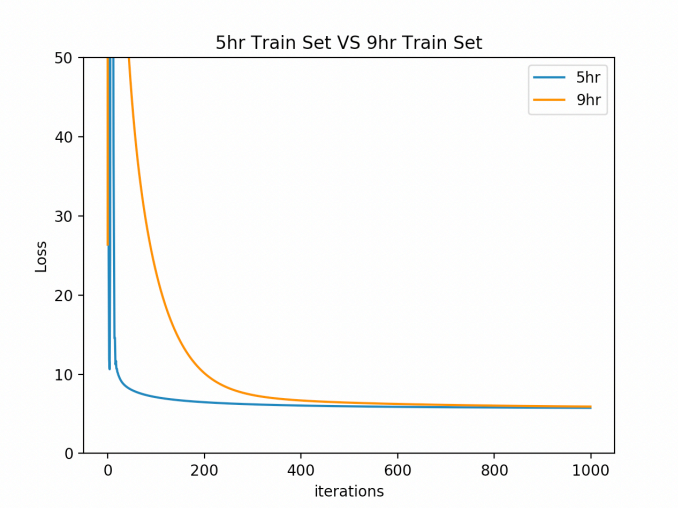
Large learning rate: Overshooting.

- 而由於使用Adagrad，在loss變小的同時趨勢也會逐漸穩定，不同的learning rate收斂到相同的minimum。

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）。

- **Validation set: 20%, Iterations: 1000, learning rate: 5**



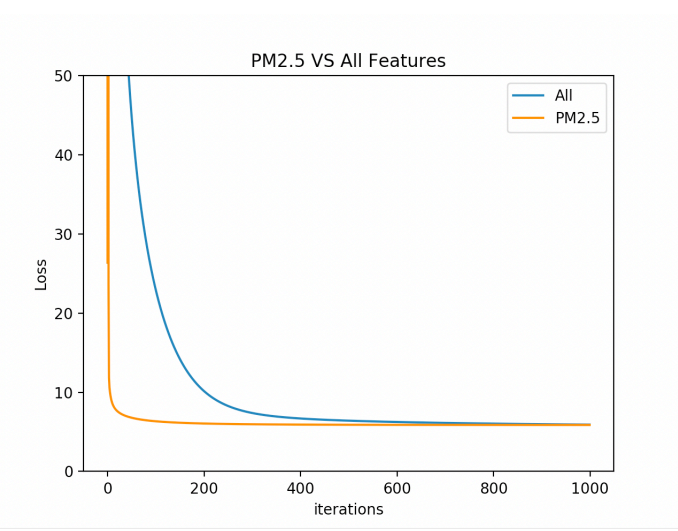

```
0:26.387424998563592
100:7.100663807892575
200:6.464612733307388
300:6.194420841650392
400:6.04210994446614
500:5.94530414571893
600:5.8795166703017845
700:5.832840518036859
800:5.798704278604927
900:5.773165399827167
5hr Val Loss: 5.753884854056803
0:26.387424998563592
100:22.91711315749352
200:10.138603857150231
300:7.37525198454542
400:6.683043106208704
500:6.407825632847055
600:6.242669385054877
700:6.124375307021735
800:6.034846764987169
900:5.965859552095241
9hr Val Loss: 5.912679333503007
```

- 在前1000次iteration中，考慮5小時的預估準確率略高於考慮9小時，5小時一開始震盪大且收斂的較快，但後來逐漸被9小時追上，我認為可能的原因有：
  - 後五小時的資訊對於預測第10個小時PM2.5較為重要，也就是前四個小時和第10個小時相關度低。因此若一起考慮前四個小時的資訊，可能會對整個model造成雜訊干擾使準確率下降。
  - iterate的次數不夠多，還沒有到最好的minimum：我算過大概iterate 10000次時9hr的Loss會開始比5hr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。

- **Validation set: 20%, Iterations: 1000, learning rate: 5**




```
0:26.387424998563592
100:22.91711315749352
200:10.138603857150231
300:7.37525198454542
400:6.683043106208704
500:6.407825632847055
600:6.242669385054877
700:6.124375307021735
800:6.034846764987169
900:5.965859552095241
All Val Loss: 5.912679333503007
0:26.387424998563592
100:6.331227737050581
200:6.048529389808491
300:5.9532462718382675
400:5.9114487413338015
500:5.890597671611179
600:5.87928162493229
700:5.872739662703342
800:5.868763674962414
900:5.866249914229503
PM2.5 Val Loss: 5.864625020149734
```

- 只考慮PM2.5時一開始收斂得較快，但逐漸被考慮全部feature的模型給追上，可能的原因有：
    - 在iterate不夠多次的情況下，只考慮PM2.5的情況會比較好，其他feature有的資訊可能會造成干擾，要到iterate更多次以後才能消除負面影響。
    - 去除了一些負相關的feature只考慮PM2.5的情況下，可以看到前9小時和第10小時的PM2.5有一定的相關性，在考慮全部feature的model還沒有降到minimum前，只考慮PM2.5的loss反而更低。

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）。

- 首先我將輸入的資訊去雜訊（將負值改為整列的平均值），再交叉比對哪些feature影響比較大，哪些會造成負面的影響（只考慮其中一項和PM2.5的loss和只去除某一項的loss），最後將NO2, RAINFALL, THC和WIND_DIRC摘除。將iteration次數從10000調為60000，learning rate 則下調到2。另外，使用了adagrad 的Gradient Descent方式和linear regression的model。最後我發現，由於七八月可能較容易受天災影響，使得資訊成為noise，於是我將七月的資訊全數移除，得到現在的結果。