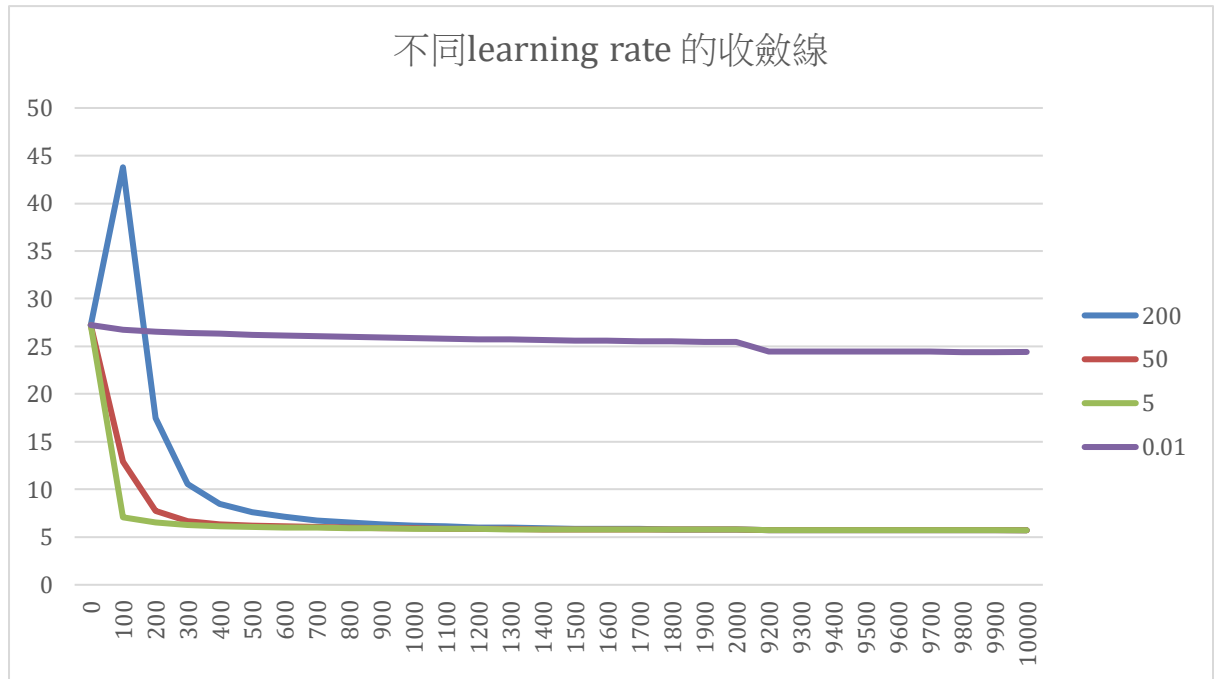


備註：

- 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）。



我選擇的 learning 分別是 200, 50, 5, 0.01，每 100 步就記錄一次 loss，而 2000 到 9100 中間的 data 我把他們領略掉，因為其實他們在 2000 之後其實就已經處於飽和狀態了，這樣圖表也會比較好觀察。

我們從圖表中可以看到，learning rate 為 200 的線條有一個明顯的上升而且下降的比較慢，若把收斂的過程比喻成在一個山谷裏面找對低點，200 的線就像一次邁出過大的步伐，反而跑到了對面的山壁上並且著陸點還比原先的位置高，之後的每一步都在山谷間跳躍，雖然一次前進的步伐很大，但是下降的並不多，效率不好，雖然經過 10000 的 iter time 還是會收斂到幾乎一樣的点，但是如果將 iter time 減少，他便不會是一個好的 learning rate。

而 5 下降的速度也比 50 的快上許多，可能是 learning rate = 50 也有跨太大步的問題，以我的測試，我最後是使用 learning rate = 5 的 model 來跑我的結果，他是我測過的數據中最剛好的 learning rate。

而 0.01 則是我故意將 learning rate 調至過小，雖然他有在穩定的下降，但是速度實在是太慢了，光走 10000 步還是無法到達飽和點(還差上許多)，雖然他最終還是會到達飽和點，並且結果可能比其他的 learning rate 還要來得好，但是以有限的時間以及資源，我想他還是不是一個好的 learning rate。

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

9hrs: 5.6648, 5hrs:5.6728

五小時的 validation loss 略大於 9hrs 的 validation loss，我想可能的原因應該是 pm2.5 跟先前的各項指數有關，時間距離越久越沒有影響但是還是站有一定的效力，參考九小時觀察的時間比較長，自然預測出來的資料也比較準。

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

9hrs 所有的 feature: 5.6648

只取前 9hrs 的 PM2.5: 5.8611

PM2.5 跟空氣中的各項指數有關，雖然跟先前的 PM2.5 的趨勢有密切的關係，但是還是不能忽略其他因素的影響。

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)。

我刪掉了 7 個 feature(分別是第 0, 4, 8, 10, 11, 14, 16 個 feature)，效果變好上許多。我一開始先各自把每一個 feature 刪掉，並且計算 validation loss，然後把他們拿去跟完全沒有把任何 feature 刪掉的 validation loss 相比，若 validation loss 有比較小便把他記錄下來。之後再一次性把剛剛紀錄的 feature 刪掉，再拿去跑結果，便得到了一個比參考 18 個 features 好上許多的結果。