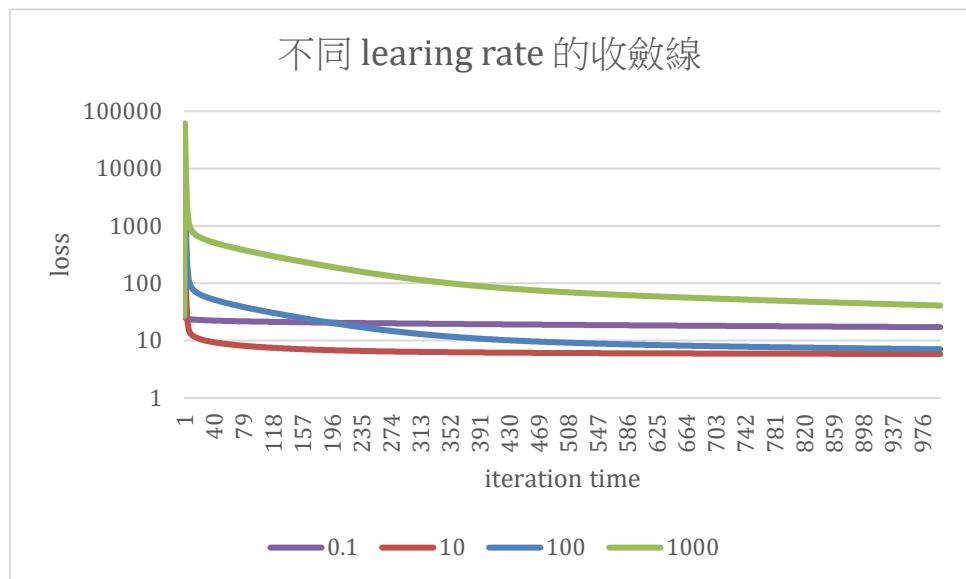


備註：

- 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）。



當 iteration time = 1000 時過大或過小的 learning rate 都會導致最終收斂的結果不理想。過大 (learning rate = 1000) 會收斂在較大的值，過小 (learning rate = 0.1) 則是收斂過慢，故此模型的最佳 learning rate = 10。

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)。

features	train_set loss	valid_set loss
$5*18 + 1$	5.982668367147294	5.526824936259741
$9*18 + 1$	5.854867290534058	5.459852575458676

可以明顯看到9小時的表現較好，其中可能的原因是5小時的模型過於簡化，資料不足以準確預測。

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。

features	train_set loss	valid_set loss
9*1 + 1	6.357397636944634	5.55460422689614
9*18 + 1	5.854867290534058	5.459852575458676

可以看到18features的表現較好，其中可能的原因是只取PM2.5的模型過於簡化，而其他的資料應該是有幫助於預測的，像是風速、降雨等。

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）。

用跟原本一樣的模型(linear regression)，只是根據第一題的結果，learning rate = 10 且 iteration time = 10000。本來有嘗試lasso regression的，但結果並沒有比較好。