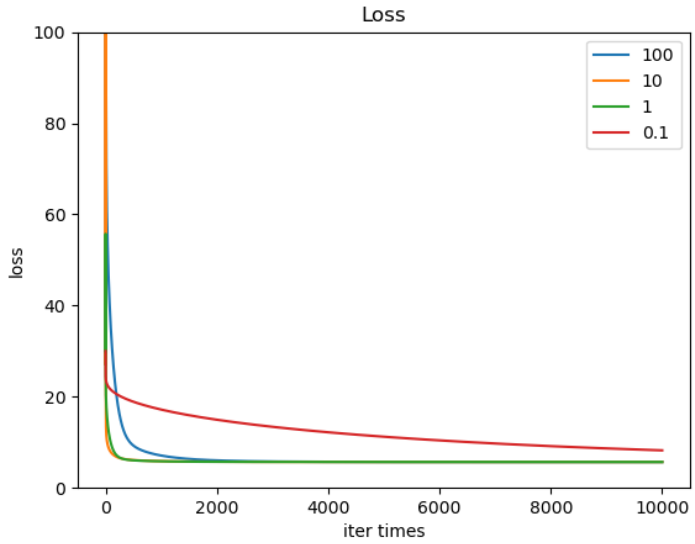
學號：B06902006　　系級：資工三　　姓名：王俊翔

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）。



由上圖可知，收斂速度為 learning rate = 1 及 10 的時候為最佳。當 learning rate 等於 0.1 時，由於每次更新得太少導致收斂的速度非常的慢；100 時，由於一次跨的步伐太大，很容易因為參數變化太大造成結果走偏(無法保證目前走的就是最後最佳解的方向)，而 1 及 10 是比較平衡的情況，所以收斂得比較好也較快，其中 10 又稍微好過 1。

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

參數 / 小時	前 9 小時	前 5 小時
iter = 1000 learning rate = 100	5.91220546628651	5.759285349672934
iter = 25000 learning rate = 10	5.665474379473833	5.6812353506909075

由上表可知，在 iteration 越大的情況下，前 9 小時會預測的比前五小時還好，不過若是在較少的 iter 的情況下，由於前五小時的 feature 較少，達到最佳的情況不需要那麼的久，所以會有比較優異的結果，不過其實相差也不大，由此可知，其實前 9 到前 5 小時的數據可能只對 training 有些微的幫助。

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。

參數 / 小時	18*9+1	9+1
iter = 1000 learning rate = 100	5.91220546628651	5.86461175212293
iter = 25000 learning rate = 10	5.665474379473833	5.861093589266866

其實這題最後做出來的結果與第二題類似，不過在 feature 更少的情況下，我們可以發現差異更加的明顯，一樣在 iter 等於 1000 時，由於只取 9 個 feature 很容易就達到最佳，所以其實會稍微優於 18*9 時的情況，但隨著 iter 次數增加及審慎的選 learning rate 後，只取 9 個就會被 18*9 打爆了。

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）。

我主要做了幾個方向的調整 ：

1. 對數值不正常的 feature 做處理，當數據小於 0 時，我把它轉變成此 24 內的平均(不包括<0 的數據)。(test 跟 train 都做)
2. 對每個類型的 feature 進行分開的 training 並計算其 loss，< 16 者就使用，所以會有{18 種中<16 的個數}*9 個 feature，再去做 training。
3. Learning rate = 10 Iter = 30000