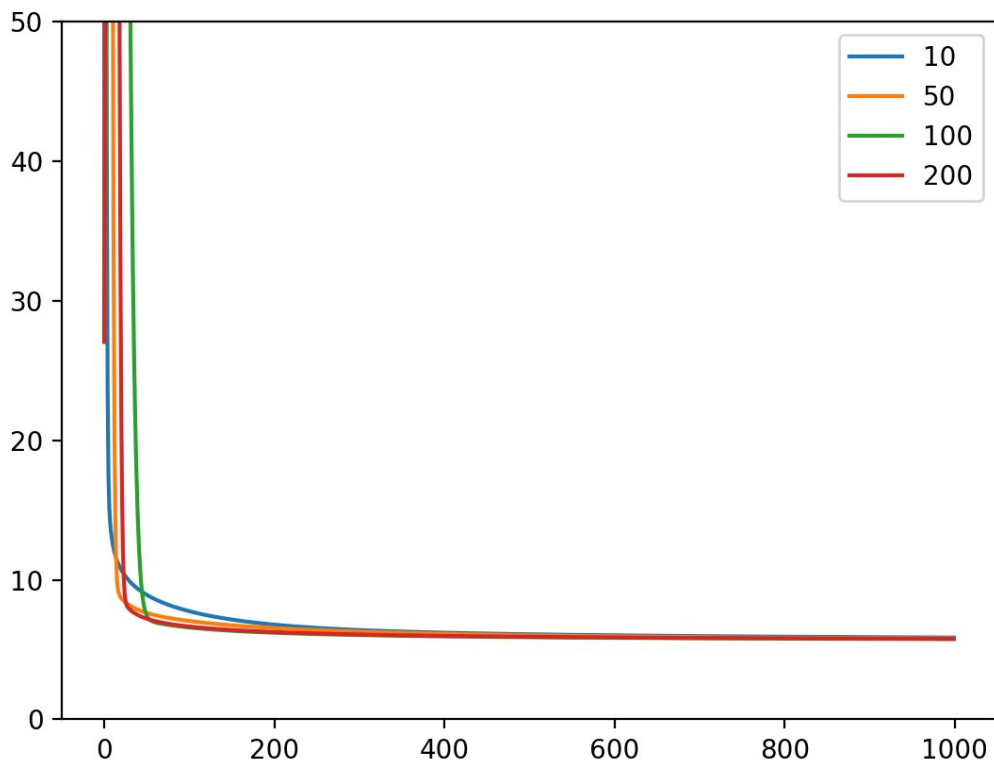


1. (2%) 使用四種不同的 learning rate 進行 training (其他參數需一致)，作圖並討論其收斂過程 (橫軸為 iteration 次數，縱軸為 loss 的大小，四種 learning rate 的收斂線請以不同顏色呈現在一張圖裡做比較)。



從圖中可以顯示當learning rate = 10的時候，最快進入較為穩定的情形，那是因為他每次都走比較小步，而learning rate 越大，震盪的起伏也會越大。因此，我認為以要train一個這樣較簡單的model來說，追求穩定的趨勢是比較好的，最後一定比較能穩穩地收斂到極值。不過在我這次的實驗中，四組的loss都可以收斂至5.72左右，代表這四個learning rate在此情形表現並不會相差太多。

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

	Training Set	Validation Set	Testing Set
All features in 9hrs	5.720219	5.678231	5.431982
All features in 5hrs	5.862310	5.679758	5.663820

實驗發現，無論是在training set 還是 testing set上面，取前9hrs的feature的表現普遍都比取前5hrs的表現還要好。這代表了取較多的data來train還是可以得到較好的準確率，並且沒有因為比較複雜而造成overfitting的現象。

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。

	Training Set	Validation Set	Testing Set
取所有features	5.720219	5.678231	5.431982
只取PM2.5	6.154903	5.821214	5.708349

從上表中，我們可以得知只取PM2.5來訓練我們的資料表現是明顯要比加入另外17個features差的。因此，我們除了從上一題可以知道PM2.5跟前9個小時的天氣狀況有關係，也會被當下的其他天氣數據所影響。

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)。

我超越strong baseline 的方法為使用feature engineering 的技術，將一些不那麼重要的feature拔掉來達成更高的準確率。此外，有發現拿掉PM10似乎在public有很好的表現，但是因為PM10與PM2.5的相關係數實在很高，因為我認為有overfitting的可能。最後我還調整了一些參數，以及將data分成training set 跟 validation set來驗證model的好壞，來獲得更高的準確率。