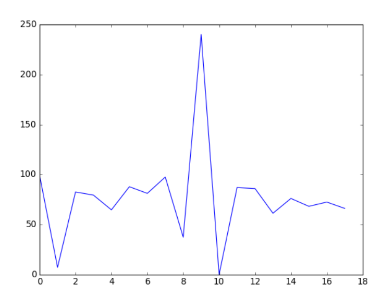
學號：R05921037 系級： 電機碩一 姓名：陳冠廷

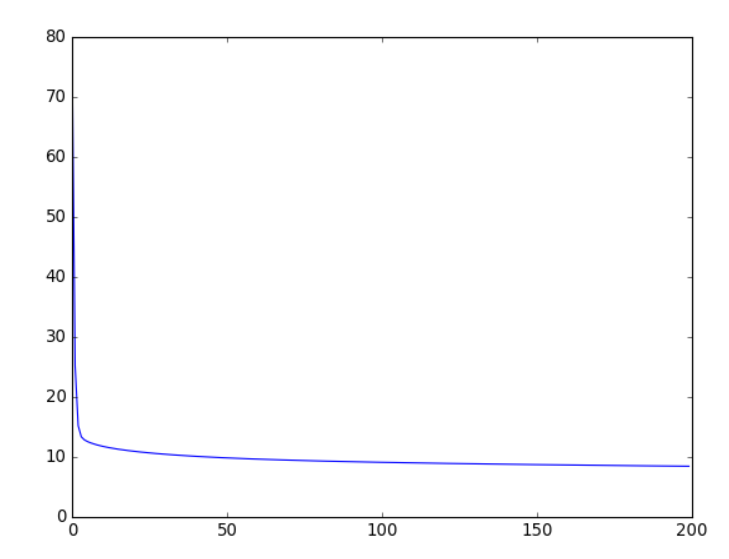
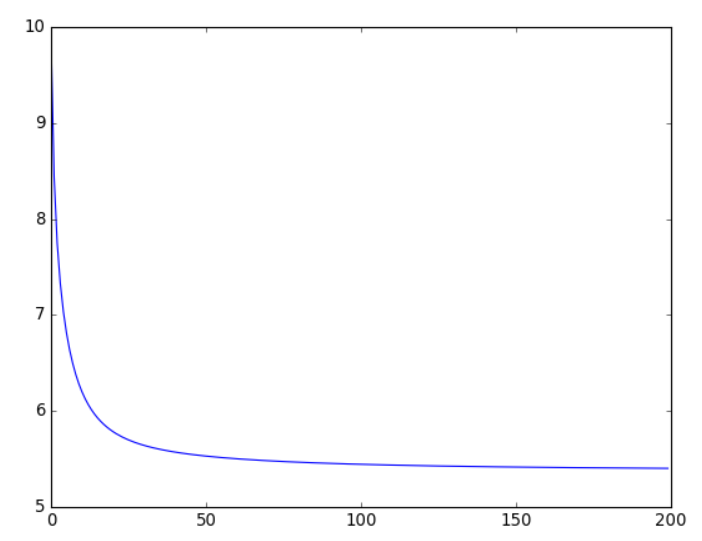
1. 請簡明扼要地闡述你如何抽取模型的輸入特徵 (feature)

答：

用open的方式先把每一行row抓下來，再把多餘的column、row濾掉(例如train.csv的第一行)，然後先用split(,240)先切成240個24\*18的矩陣，再用for迴圈把numpy矩陣用[i:i+9]決定162個feature，最後先用coorelation做出右圖，把最小的三個feature濾掉，形成135個feature的一個input data。

2.請作圖比較不同訓練資料量對於PM2.5預測準確率的影響

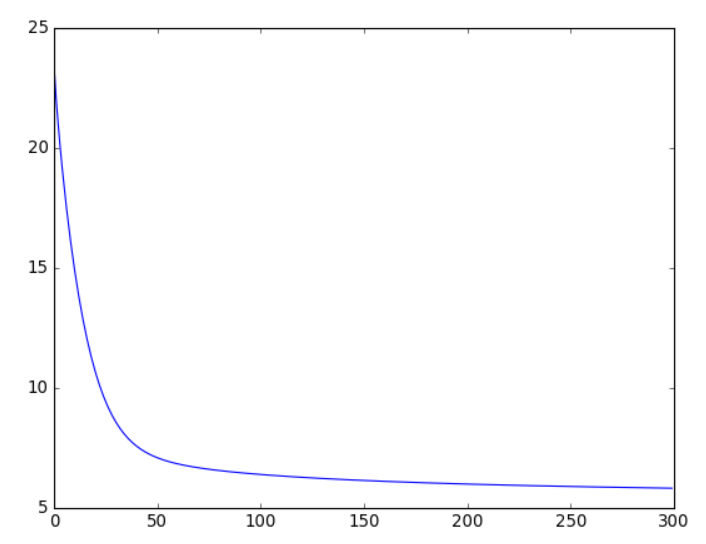
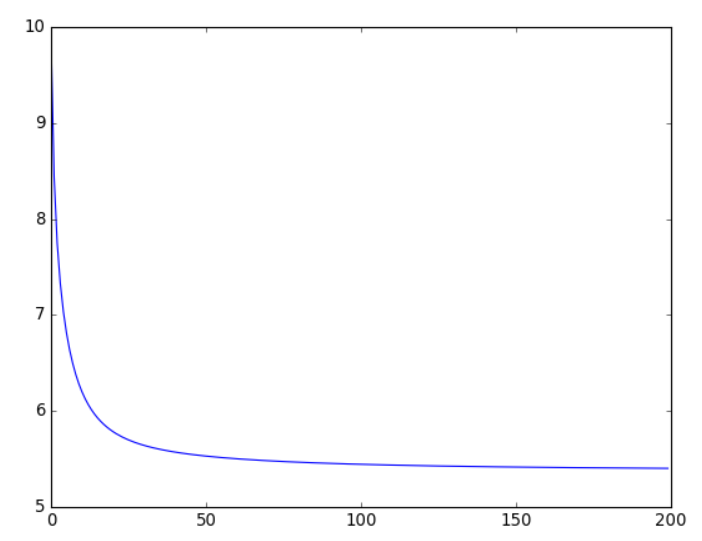
答：



左圖為用Stochastic Gradient Descent，右圖為一次丟進所有data，可以看到兩個極端例子，batch size=1時下降的幅度可以到5.5(RMSE)才趨於平穩，而batch size 是所有data時降到9.8多就不會往下掉，我想這是由於當一筆一筆資料去辨認時，路徑會比較明確，但若是整個資料丟進去，每個weight的變量會是所有資料所算出gradient descent的合，這樣會導致一些資訊因為加加減減彼此抵消而消失。

3. 請比較不同複雜度的模型對於PM2.5預測準確率的影響

答：



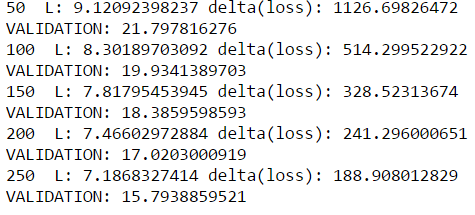
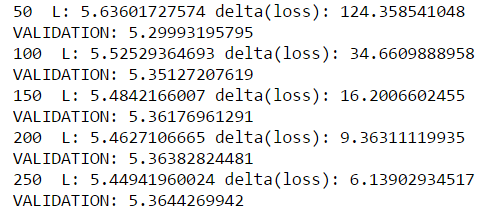
左圖為二次式(mx^2+nx+b)又圖為一次式(nx+b)，可以看出二次式明顯的比較快到最低點，且二次式收斂的位置較低，二次式收斂到5.513而一次式收斂到5.920，而為何兩個其實分數都不錯的原因我想是我已經把三個不相干的資訊去除的因素。

4. 請討論正規化(regularization)對於PM2.5預測準確率的影響

答：

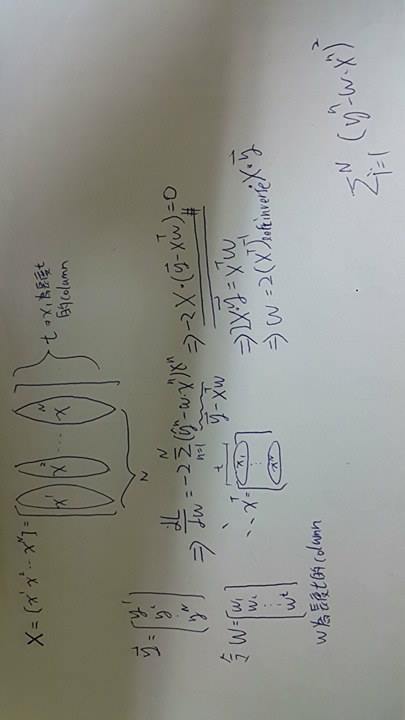
以這次實驗的感覺，我認為是在降低gradient descent的下降速度並且不讓學習overfitting，但其參數加上原本的learning rate有兩個參數變量，所以後來就用Adagrad的方式讓學習不會overfitting。

左圖是用adagrad右圖沒用，可以看到不只loss下降慢而且validation error也很大，我覺得可能是因為每次iteration的速率沒下降導致震盪讓學習曲線無法很直接地往正確的方向走。



5. 在線性回歸問題中，假設有 N 筆訓練資料，每筆訓練資料的特徵 (feature) 為一向量 xn，其標註(label)為一存量 yn，模型參數為一向量w (此處忽略偏權值 b)，則線性回歸的損失函數(loss function)為 。若將所有訓練資料的特徵值以矩陣 X = [x1 x2 … xN] 表示，所有訓練資料的標註以向量 y = [y1 y2 … yN]T表示，請以 X 和 y 表示可以最小化損失函數的向量 w 。

答： W= [(Xtranspose)left inverse] X



上面不小心加上了2所以用紅色蓋掉