Machine Learning HW1 Report

b02901122電機四 劉致廷

1.Linear regression function by Gradient Descent.

A:

linear function : y = Xw (X 為input data, bias 用 1 併入了X 中, w為對應data的參數)

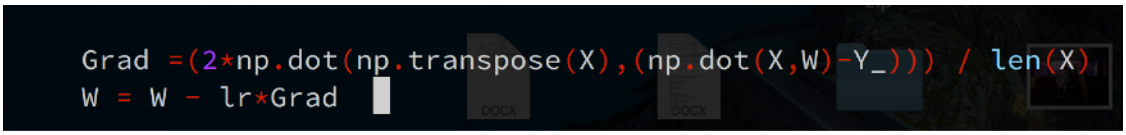
計算 **loss: L = ll y - y_ ll^2** (y_為label)

由於我是一次看所有training data再一次利用Gradient Descent 更新Weights

因此，X會是矩陣而y 則是一個向量，利用矩陣的轉置與微分等規則，可以得出

∇ L(w) = 2* X.T*(y-y_) / (data總數) ← 做平均

→ w = w - (learning_rate)* ∇ L(w) ← Gradient Descent



```
Grad =(2*np.dot(np.transpose(X),(np.dot(X,W)-Y_))) / len(X)
W = W - lr*Grad
```

2.Describe your method.

A:

(1).Dara processing:

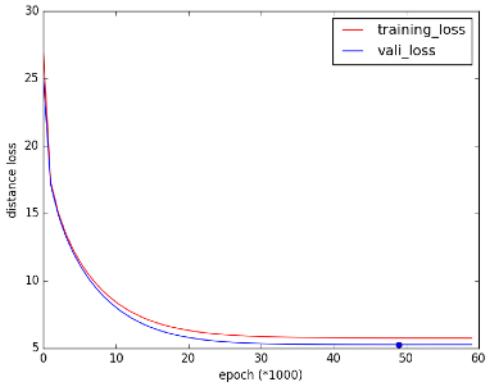
首先，利用csv library,我把.csv檔讀成一個(12月*20天*18參數,24 小時)大小的numpy array，由於月份之間並沒有連續，因此先把array切成240份後再每20天串接起來，接著，因為每個月份中的20天是連續的，因此把每九個小時的(18*9)=162筆training data抓出來，因此每個月會有(20天*24小時-9)=471筆 162維 的data。最後，在做完十二個月一樣的處理後，會是一個維度為（471*12,162)＝**(5652,162)**的numpy array，我們稱做X。

(2). Train & Validation:

隨機把training data中的 5000筆資料當作真正的Training data，把剩餘的652比當作validation的data。便可以利用cross validation的方法來找到最適合的model。

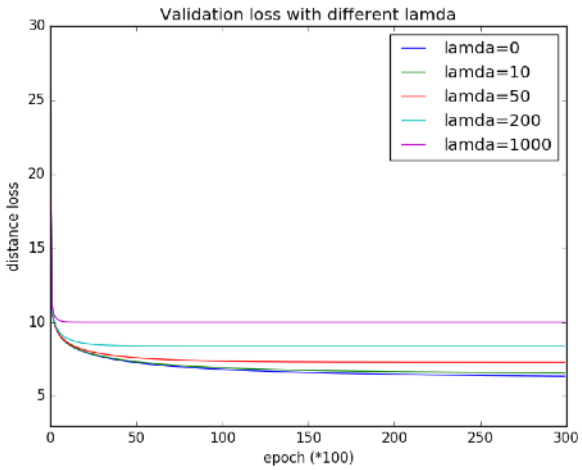
(3). Model

利用linear regression(y=wx+b)，但由於運算方便，因此把b併入w & x 中，因此先創造一個**w為(163,1)的array**，使之 y = Xw ，而y就是一個(5652,1)的array，最後再用第一題所說的方法來做Gradient Descent。我的epoch大約都在5~8萬次，learning_rate為 0.0000015。我會在training過程中即時的利用validation data來test，如果發現validation 的loss開始上升，便會停下training。如下圖，藍點為validation loss 開始上升的點，因此就可以知道大概在50000 epoch就可以停止了。



3.Discussion on regularization.

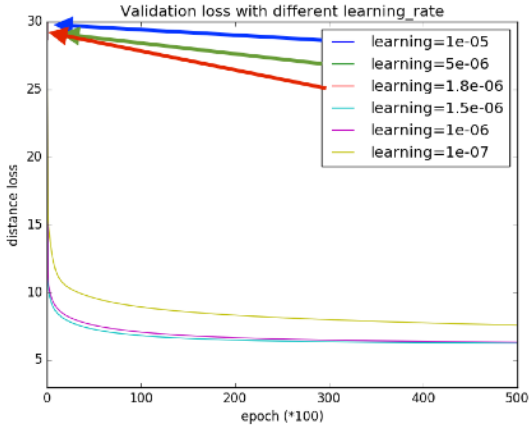
A:　　我利用validation data 來test 不同 lamda 的model，如下圖，首先可以發現，loss似乎還是沒有regularization的會比較小。



4.Discussion on learning rate.

A:

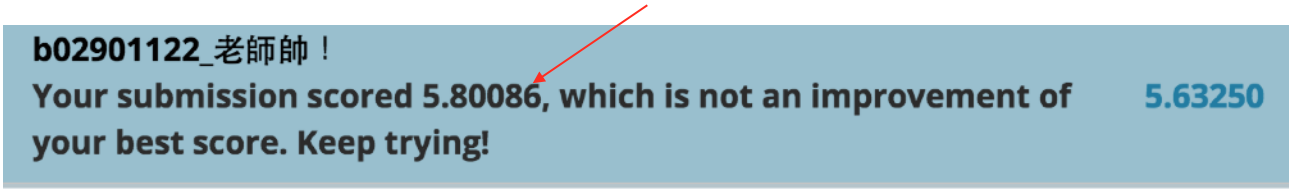
learning如果稍微大了一點，幾乎就會立刻nan掉，所以我的learning rate大約都是設在0.0000015左右，詳細如下圖。



5. Other Discussion

(1)有無normalize 的比較：

我有嘗試把training & testing data 全部放在一起，做一個rescaling，也就是((x-平均)/標準差)，做完之後因為每一維的data都在同個scale上，因此我的learning rate可以調到0.01，然而似乎並沒有得到比較好的效果，如下圖所示，



(2)減少training data的參數量：

根據污染物特性，大減少不相關的參數，只留下NO2,NOx,O3,PM10,PN2.5,SO2,使得參數量從162個降為54個，在相同的learning rate下，成績有稍微進步。