

Q1: Linear regression function by Gradient Descent.

#我以要預測的 pm2.5 的前 5 小時的 pm2.5 作為 feature

if data.iloc[row,2] == "PM2.5":

#以 window 的方式向後位移，可以充分利用 training data

for col in range(8,26):

#wX 表示第前 X 小時 pm2.5 的 weight

```
f = b \
    + w11 * data_float(row, col-1) \
    + w21 * data_float(row, col-2) \
    + w31 * data_float(row, col-3) \
    + w41 * data_float(row, col-4) \
    + w51 * data_float(row, col-5)
```

#將 cost function 微分並累減，G_X 為累計微分值的參數

```
G_b = G_b - 2.0 * ((data_float(row, col)) - f) * 1.0
G_w51 = G_w51 - 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 5)
G_w41 = G_w41 - 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 4)
G_w31 = G_w31 - 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 3)
G_w21 = G_w21 - 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 2)
G_w11 = G_w11 - 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 1)
```

#以下 SG_X 為 adagrad 累加一階微分值的平方的參數

```
SG_b += ( 2.0 * ((data_float(row, col)) - f) * 1.0) ** 2
SG_w51 += ( 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 5)) ** 2
SG_w41 += ( 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 4)) ** 2
SG_w31 += ( 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 3)) ** 2
SG_w21 += ( 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 2)) ** 2
SG_w11 += ( 2.0 * ((data_float(row, col)) - f) * data_float(row, col - 1)) ** 2
```

#以 adagrad 的方式去更新 weight

```
b = b - ((alpha * G_b) / ((SG_b)** 0.5))
w51 = w51 - ((alpha * G_w51) / ((SG_w51)** 0.5))
w41 = w41 - ((alpha * G_w41) / ((SG_w41)** 0.5))
w31 = w31 - ((alpha * G_w31) / ((SG_w31)** 0.5))
w21 = w21 - ((alpha * G_w21) / ((SG_w21)** 0.5))
w11 = w11 - ((alpha * G_w11) / ((SG_w11)** 0.5))
```

Q2:Describe your method.

我所選擇的 model 在 private 中最佳成績為 7.33644 是以欲預測的 pm2.5 的前 5 小時的 pm2.5 作為我的 feature，並加入 adagrad 的方式去更新 weight，但此成績並未達到 baseline 的要求。

但其實在我上傳的其他數據中是有 2 份數據的分數分別為 7.16943 與 7.19623 是有超過 baseline 的要求的。他們除了考慮過往的 pm2.5 數值，還加入 AMB_TEMP, O3 ,PM10 與 SO2 在欲預測的 pm2.5 前一小時的數值。

我在此討論我在所有 private 中最佳成績，在繳交程式部分還是以 private 排名上的該份預測的 model。

剛抓下 training data 時有先觀察一下 data 中所有紀錄數據與 pm2.5 的關係與分布情形，發現有些數據的起伏會略與 pm2.5 相同，其中最相符的就是 AMB_TEMP, O3 ,PM10 與 SO2。

有鑑於老師在上課中有說他如果用神奇寶貝所有測量數值去做預測，會有 overfitting 的現象，所以我只選了以上 4 項數據的前一個小時加上 pm2.5 的前 5 個小時做預測，做出來的結果在 public 上沒有達到 baseline 的要求，我本來以為可能是 overfitting 或者是 learning rate 取的不好，但是在後來與其他人討論後發現是 feature 取太少，本來想要重做一份 model，但已經來不及了。

從此次的作業中我學到，在使用機器學習的方法時，我們不能考慮太多我們已知的事情，例如經過觀察數據而去決定不使用某些數據，機器學習的目的就是為了在這些 data 中找到屬於機器的判讀方法，我們因該減少人為的因素，讓機器有更大的發揮空間。

Q3:Discussion on regularization.

老師在課程中講 regularization 時，說目的是為了讓整條 function 更平滑，可以較不受 training data 影響。但考慮到我沒有用到 2 次方以上的變數，function 都是以線性方式呈現，不太會有 overfitting 的問題，又怕加入 regularization 時會使我預測 function 偏離最佳的 function，故我最後選擇沒有使用。

Q4: Discussion on learning rate.

剛開始不知道 learning 要設多少才算合理，所以我非常小的數字開始，但是程式跑了一陣子數值沒有太大的變化，所以我猜測因該是 learning rate 太小，我逐漸地把 learning rate 加大，去觀察 weight 與 learning rate 的變化。

隨著 learning rate 的加大 weight 的變化速度的確是加快了不少，但最終因為 learning rate 調得太大 error 開始爆炸。我最後找 learning rate 的方法大約是讓 error 不會爆炸的最大值左右的級距。

在第 3 堂課，老師教了 adagrad 的方式去調整 learning rate，套用上去後發現原本所使用的 learning rate 因為會除上微分的平方，似乎可以設的更大。而在加入 adagrad 的 learning rate 的確可以使 weight 更快達到最佳的結果。