

HW#1 Linear Regression Report

B03901027 徐彥旻

1. Linear regression function by Gradient Descent.

```
# generating W, and W[-1] is B
W = np.array([random.random() for i in range(len(X[0]) + 1)])
L = np.array( [( Y[j] - W[-1] - np.inner(W[:-1],X[j]) ) for j in range(len(X)) ] )
Delta = np.zeros(len(X[0]) + 1)
for ii in range(nEpoch):
    # compute Delta and change parameters
    for j in range(len(X[0])):
        # g_j is the gradients w.r.t. W[j]
        g_j = (-2) * math.fsum(L * np.array([X[k][j] for k in range(len(X))])
    ) + 2 * RC * W[j]
        Delta[j] = - LR * g_j
        W[j] = W[j] + Delta[j]
    # change B
    g_j = (-2) * (math.fsum(L))
    Delta[-1] = - LR * g_j
    W[-1] = W[-1] + Delta[-1]
    L = np.array( [( Y[j] - W[-1] - np.inner(W[:-1],X[j]) ) for j in range(len(X)) ] )
```

2.(1%) Describe your method.

2.1 training feature (X,y)

將所有時序上連續的資料串起來，對於所有連續十個小時的資料，取其前九小時作為 X，第十個小時的 PM2.5 當作 y。

在 kaggle_best 中，只取 PM2.5 這一項目。將所有的項目做過比較，與 PM2.5 相關係數最高的前七項分別是，PM2.5、PM10、NO₂、NO_x、SO₂、O₃、THC。在跑訓練用的資料時，加入這些參數的結果是比較好的，但是 Kaggle 上的公開分數卻會隨著項目數量的增加而遞減，懷疑是 Overfitting，但也有可能是在公開的資料組上表現較差，但是整體來說表現比較好。

2.2 other methods

有使用 regularization， $\lambda = 100$ ，最後使用的兩組 kaggle 分別是 (1) 只有考慮前九個小時的 PM2.5 以及 (2) 考慮相關係數前三高的三個項目 (PM2.5、PM10、NO₂)。其餘有嘗試過但結果不甚理想的技術於第五部分討論。

3.(1%) Discussion on regularization.

附圖一是有無使用 regularization 的比較。藍色的是 $\lambda = 0$ ，紅色為 $\lambda = 100$ ，從圖中可發現有使用的 loss 比較小，原因是（1）使得不會有特定的 weight 特別大，突然地微小改變使得預測值有大幅度的變動。（2）變成 smooth function

4.(1%) Discussion on learning rate.

附圖二是 learning rate 適中、過大、過小的狀況，藍色的是 $\eta = 2.7 \times 10^{-8}$ ，紅色為 $\eta = 1.6 \times 10^{-7}$ ，綠色則是 $\eta = 2.7 \times 10^{-10}$ ，若過大則會因為每次更新參數的幅度太大，導致 loss 反而上升；若過小則是 loss 雖會降低，但是降低的速度過慢，沒有效率。

5.(1%) TA depend on your other discussion and detail.

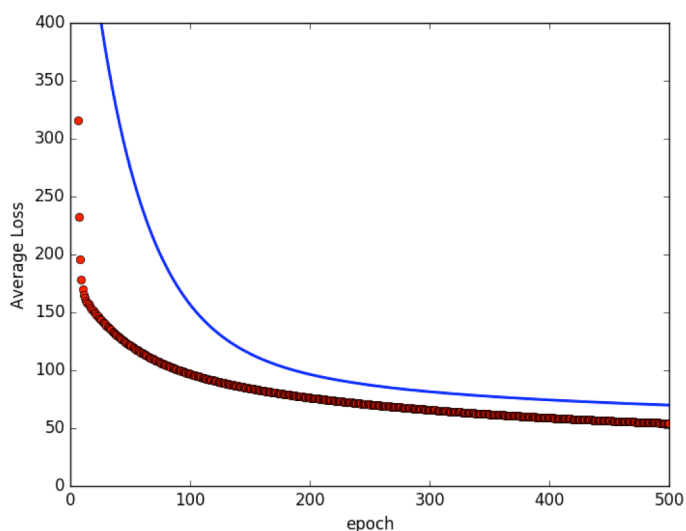
5.1 Failed Trial

嘗試過 stochastic gradient decent、Adagrad、Adadelta、mini batch，其中 Ada 系列的要接在只有 regularization 的之後訓練才能夠達到跟只有 regularization 差不多的 loss，否則會訓練不起來。stochastic gradient decent 跟 mini batch 則是會導致效能變慢，程式無法在合理的時間內跑完。此外，在項目多於一項的時候（不只考慮 PM2.5）還有試過 normalization，但是效果也沒有比較好。

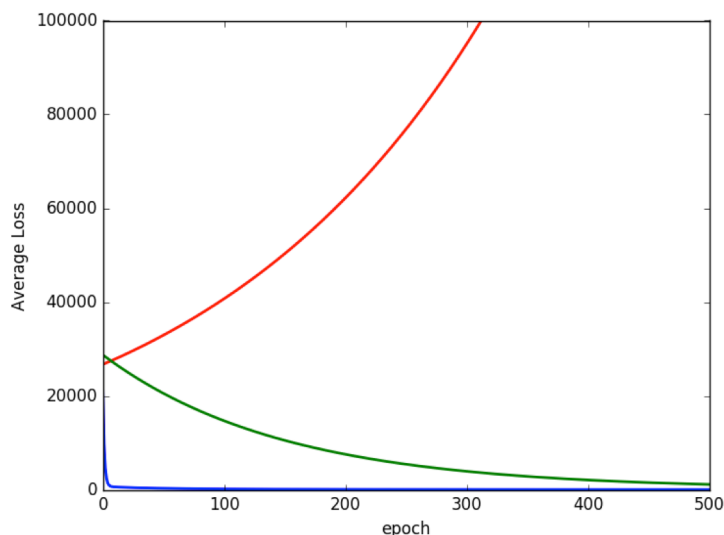
5.2 Future Improvement

可以看看實作成功的人是怎麼寫這些功能的，因為對於 python 尚不算是熟悉，可能會選到效能極差的實作方式。如果能參考別人的寫法，就能驗證這些技術理論上會有的改進效果。

此外，也可以考慮繼續實作 Adam，或是將 loss function 換成 cross entropy，應該都有機會使得訓練的效率與結果有所提升。



附圖一



附圖二