資工所 碩一 黃郁庭 R05922038

**[Linear regression function by Gradient Descent]**

```
LAMBDA=0
alpha=0.1

weight=np.array([[0.0]*9]*18)
bias=0.0
gb=0.0
wb=np.array([[0.0]*9]*18)

iteration=30000
for i in range(iteration):

    b_grad=0.0
    w_grad=np.array([[0.0]*9]*18)
    for n in xrange(len(trainSet)):
        sum_WX=np.sum(weight*trainSet[n,...,:9])
        b_grad=b_grad-2.0*(trainSet[n][9][9]-bias-sum_WX)*1.0
        w_grad=w_grad-2.0*(trainSet[n][9][9]-bias-sum_WX)*trainSet[n,...,:9]
    w_grad+=2.0*LAMBDA*weight

    gb+=(b_grad**2)
    wb+=(w_grad**2)

    bias=bias-alpha*(1./(gb**0.5))*b_grad
    weight=weight-alpha*(1./(wb**0.5))*w_grad
```

圖一、update bias and weight

在實作的 linear_regression.py 中，Bias 與 weight 皆初始為 0，更新 bias 與 weight 的 function 如圖一所示，其中每筆 data(共 5652 筆 data)使用 162 個 features。因為 loss $L=\sum_n^{5652}(\hat{y}^n-(b+\sum_i^{162}w_i x_i^n))^2+\lambda\sum_i^{162}w_i^2$，所以$\frac{\partial L}{\partial w_k}=\sum_n^{5652}2(\hat{y}^n-(b+\sum_i^{162}w_i x_i^n))(-x_k^n)+2\lambda w_k$，k=1,2,...,162，$\frac{\partial L}{\partial b}=\sum_n^{5652}2(\hat{y}^n-(b+\sum_i^{162}w_i x_i^n))(-1)$，learning rate of b：$\eta\, b_t=\frac{\eta b_0}{\sqrt{\sum_j^{t-1}gb_j^2}}$ (其中 $gb_j=\frac{\partial L}{\partial b}$)，learning rate of $w_k$：$\eta\, w_{k_t}=\frac{\eta w_{k_0}}{\sqrt{\sum_j^{t-1}gw_{k_j}^2}}$ (其中$gw_{kj}=\frac{\partial L}{\partial w_k}$)，因此在每一 iteration 更新 bias 的 function 為：$b_{t+1}=b_t-\eta\, b_t*\frac{\partial L}{\partial b}$，更新 weight 的 function 為：$w_{k\,t+1}=w_{k\,t}-\eta\, w_{k_t}*\frac{\partial L}{\partial w_k}$。程式碼中 sum_WX=$\sum_i^{162}w_i x_i^n$ (python 的 ndarray 有個特性是兩個相同大小的 2d-array 相乘的結果即為一個 2d-array 且其中 element 值為這兩個 2d-array 在相對應之位置之 element 相乘之結果)，b_grad=$\frac{\partial L}{\partial b}$，w_grad=$\begin{pmatrix}\frac{\partial L}{\partial w_1}\\ \vdots \\ \frac{\partial L}{\partial w_{162}}\end{pmatrix}$ (為 1d-array)，gb=$\sqrt{\sum_j^{t-1}gb_j^2}$，wb=$\begin{pmatrix}\sqrt{\sum_j^{t-1}gw_{1_j}^2}\\ \vdots \\ \sqrt{\sum_j^{t-1}gw_{162_j}^2}\end{pmatrix}$，alpha=$\eta b_0=\eta w_{k_0}$=0.1。

**[Describe my method]** (詳細步驟如上所示)

本作業利用 train.csv 中連續十小時的(24hr*20days/10hr*10-9)*12=5652 筆資料預測 test_X.csv 中

240 筆中每筆 data 第 10 小時的 PM 2.5，其中使用 18(觀測項目數)*9(連續 9 小時)=162 個 features，因為 Gradient Descent 的 train 頗慢，所以取不同的 feature 對誤差的影響會在 kaggle_best part 討論。本作業的 Gradient Descent 是使用 adagrad， λ 設為 0、alpha(=η$b_0$=η$w_{k_0}$)設為 0.1，adagrad 的特性為每一次都會調整 bias 或 weight 前進的大小(利用上述 $\eta\, b_t$ 與 $\eta\, w_{k_t}$，若每一次的 gradient $\frac{\partial L}{\partial b}$與$\frac{\partial L}{\partial w_k}$越大、 $\eta\, b_t$與 $\eta\, w_{k_t}$會越小)，依觀察的結果、其較(沒使用 adagrad 的)Gradient Descent 可以更快到達目的地，可能是因為後者容易進入 local minimum。

**[comparison]**

註一、Error=Loss/N，N=validation set 之 data 數，**avg. error = cross validation's average error**。

註二、表一~表三與表五之 error 皆為將 train set(5652 datas)**任意均分成 3 組做 cross validation**、且皆使用 162 features 計算得出。

表一為 regularization 對 error 的影響，這裡使用 adagrad，alpha=η$b_0$=η$w_{k_0}$=1.0，每次 train 10,000 次。由結果得知 λ 越小 error 越小。

| λ | 0 | $1.0*10^{-13}$ | $1.0*10^{-10}$ | $1.0*10^{-7}$ |
|---|---|---|---|---|
| avg. err | 36.036566441 | 36.036566441 | 36.036566441 | 36.0365664411 |
| λ | 1 | 10 | 100 | 1000 |
| avg. err | 36.0374640164 | 36.0454893955 | 36.1212528508 | 36.7043610144 |

**表一、regularization 對 error 的影響**

表二為 learning rate 對 error 的影響，在這裡不使用 adagrad，即$b_{t+1}=b_t-\eta*\frac{\partial L}{\partial b}$，$w_{k_{t+1}}=w_{k_t}-\eta*\frac{\partial L}{\partial w_k}$， λ=1.0，每次 train 10,000 次。由結果得知 η 在 $2.0*10^{-10}$ 效果較好，但如果 η 在大一點會產生 overflow 的結果。

| η | $1.0*10^{-13}$ | $1.0*10^{-12}$ | $1.0*10^{-11}$ | $1.0*10^{-10}$ | $2.0*10^{-10}$ |
|---|---|---|---|---|---|
| avg. err | 242.111171005 | 171.445020947 | 93.9070230552 | 53.4541009199 | 45.6914089546 |

**表二、learning rate 對 error 的影響(非 adagrad)**

表三為 initial learning rate 對 error 的影響，在這裡是用 adagrad， λ=1.0，每次 train 10,000 次。由結果得知 initial learning rate (alpha=η$b_0$=η$w_{k_0}$)在 0.1 時效果較好。

| η$b_0$/η$w_{k_0}$ | 10.000 | 1.000 | 0.100 | 0.010 | 0.001 |
|---|---|---|---|---|---|
| avg. err | 36.0046386703 | 35.9921798586 | 35.8855707994 | 36.8213481328 | 72.1151525515 |

**表三、initial learning rate 對 error 的影響(adagrad)**

表四為 train 的次數與 Kaggle score 之關係，在使用 adagrad 時，當 λ=1.0、alpha=η$b_0$=η$w_{k_0}$=1.0 時，train 的次數越多 kaggle score 越好(但也有可能花更多時間 train 更多次反而效果變差只是這裡沒有那麼多時間測)。

| iteration | 10,000 | 20,000 | 40,000 |
|---|---|---|---|
| Kaggle score | 5.97348 | 5.79287 | 5.72676 |

**表四、train 的次數對 Kaggle score 的影響**

表五為不同 linear model 對 error 的影響，Linear regression by adagrad 在當 λ=1.0、alpha=η$b_0$=η$w_{k_0}$=1.0 時、train20,000 次時，效果好於 Linear regression by closed form，但 error 最小的是 Ridge regression by closed form(後兩種 linear model 在本作業中實作的方法詳述於下頁)。

| method | Linear regression by adagrad | Linear regression by closed form | Ridge regression by closed form |
|---|---|---|---|
| avg. err | 35.0955957174 | 36.4387397142 | 34.8675066862 |

**表五、不同 linear model 對 error 的影響**

**[kaggle best]**

實作的 kaggle_best.py 使用 Ridge regression 的 closed form solution 解法，即 $w=(X^TX+\lambda I)^{-1}X^TY$，w 為 weight vector，X 為 features，Y 為 label vector，λ 是為了使 weight 盡可能小以減少 noise。

表六有三種 Feature type：A type 為本來所使用的 18(觀測項目數)*9(連續 9 小時)=162 個 features；B 為 A 加上：9 個連續的 PM2.5 乘上 9 個連續的 PM10(即 $X_{PM2.5}*X_{PM10}$)、9 個連續的 PM2.5 乘上 9 個連續的 $SO_2$、以及最後 2 個連續的 PM2.5 乘上最後 2 個連續的 $O_3$，共 182 個 features；C 為 A 加上：最後 2 個連續的 PM2.5 乘上最後 2 個連續的 PM10、最後 2 個連續的 PM2.5 乘上最後 2 個連續的 AMB_TEMP、最後 3 個連續的 PM2.5 乘上最後 3 個連續的 CO、最後 1 個 PM2.5 乘上最後 1 個 $O_3$、最後 2 個連續的 PM2.5 乘上最後 2 個連續的 $NO_2$、最後 1 個 PM2.5 乘上最後 1 個 $CH_4$、最後 1 個 PM2.5 乘上最後 1 個 WIND_DIREC、最後 1 個 PM2.5 乘上最後 1 個 RAINFALL，共 175 個 features。B type 是因為發現 PM10 與 $SO_2$ 的變化趨勢與 PM2.5 較像、並經過手調產生的 features type，而 C 只是根據 error 不斷嘗試手調所產生的結果。這裡是將 train set 分成 18 組做 cross validation 所得出的 error 平均，由表六可知在 C type 的 features 時誤差最小，但 kaggle 的分數卻是 B type 較好，很有可能是 λ 沒有調好所以有可能產生 overfitting。

另外我也實作了 Linear regression by closed form：$w=X^+Y$ ($X^+$為 X 的 pseudo inverse)，在同樣的條件下與 Ridge regression by closed form 的比較如表六，後者的結果皆比前者好，兩者基本上只差在 λ 的有無(λ 能降低 overfitting 發生的機會或降低 noise 對 train 的影響)。

| Feature type | A (162) | B (182) | C (175) |
|---|---|---|---|
| Ridge regression by closed form avg. err | 34.4056340525 (kaggle score=5.69990) | 34.4599011002 (kaggle score=5.50032) | 33.5890003216 (kaggle score=5.72128) |
| Linear regression by closed form avg. err | 34.9414766524 (kaggle score=5.80644) | 35.0190379973 (kaggle score=5.62695) | 34.1774420554 (kaggle score=5.76370) |

**表六、不同 feature 對 error 的影響**

因為考慮到每筆 data 的連續十小時的前幾小時可能用處不大，嘗試刪掉前兩小時、以連續 8 小時為一筆 data 的 train set 共可產生(24hr*20days/8hr*8-7)*12+240(from test set)*2=5676+480=6156 筆資料。表七有兩種 feature type：A type 為 18(觀測項目數)*7(連續 7 小時)=126 個 features；B type 為 A type 之 features 加上最後連續 5 個 PM2.5 乘上最後 5 個練續 PM10、最後 1 個 PM2.5 乘上最後 1 個 CO、最後 1 個 PM2.5 乘上最後 1 個 RH，B type 是經由手調並利用一次次 27-cross validation 找到 error(目前)最小之 features 組合。由結果得知 Ridge regression by closed form 之誤差皆小於 Linear regression by closed form，且 B type 較 A type 好。

| Feature type | A (126) | B (133) |
|---|---|---|
| Ridge regression by closed form avg. err | 34.27 (kaggle score=5.57888) | 33.79 (kaggle score=5.45797) |
| Linear regression by closed form | 34.61 | 34.15 |

| avg. err | (kaggle score=5.62868) | (kaggle score=5.49130) |
|---|---|---|

**表七、不同 feature 對 error 的影響**