HW1

1. Linear regression function by Gradient Descent

(a) define the costfunction

```
def costfunction(X, y, theta, londa): # theta 為所有 features 的 weight, londa 為 λ
    import numpy as np
    X = np.matrix(X)
    y = np.matrix(y)
    theta = np.matrix(theta)
    m = X.shape[0]
    return (np.sum(np.square((X * theta) - y)) + londa * np.sum(np.square(theta[1:]))) / (2*m)
    # 回傳 ( ∑(x ∗ theta − y)^2 + λ ∗ ∑(theta)^2 ) / (2*m) , theta 總和不包含 bias
```

(b) define the gradient descent (batch)

```
def gradientdescent(X, y, theta, alpha, num_iters, londa):
    # theta 為所有 features 的 weight , londa 為 λ , alpha 為 learning rate
    import numpy as np
    X = np.matrix(X)
    y = np.matrix(y)
    theta = np.matrix(theta)
    m = X.shape[0]
    L_history = np.zeros(num_iters)
    adagrad = np.zeros(theta.shape)
    for i in range(num_iters):
        temp = X.T * (X * theta - y)   # 計算 costfunction 的微分
        temp[1:] += londa * theta[1:]   # + λ ∗ ∑(theta)  總和不包含 bias
        temp = (alpha/m) * temp   # 乘上 learning rate
        adagrad += np.square(temp)   # 使用 adagrad
        temp = temp/np.sqrt(adagrad)
        theta -= temp   # 進行 weight 更新
        L_history[i] = costfunction(X, y, theta, londa)   # 紀錄每次更新 weight 後的 cost
    return theta, L_history, adagrad
```

2. Describe your method

由於 test data 是給連續 9 小時的 18 個測項來預測第 10 小時的 PM2.5，則我的 training data 則一樣是取連續 9 小時的 18 個測項共 162 個數值當作 X (X=[x_1, x_2,…., x_{162}])，還有這 162 個數的平方 (X^2=[x_1^2, x_2^2, …., x_{162}^2]) 和一項 Bias (b)，並且取第 10 小時的 PM2.5 作為 Y，則我的 function 如下。

$$f(X) = b + \sum_{i=1}^{162} w_i x_i + \sum_{i=1}^{162} w_i' x_i^2 = Y$$

我將每個月 20 天的 0 到 23 小時串連在一起得到每個月連續 480 小時的各個測項，再於每個月中 480 小時取所有連續 9 小時的測項數值和第 10 小時的 PM2.5，總共 12 個月取得 5652 筆 data (12*471 =5652)，並將所有 X (X, x^2)都進行 feature scaling。

最後再將 data 再切分為 training data 和 cross-validation data，其比例為 2:1。

3. Discussion on learning rate

由於在 gradient descent algo.裡加了 adagrad ，則初始的 learning rate 則不會影響最後的準確率，但也不要設太大，目前使用 0.1。

4. Discussion on regularization

欲比較在不同 λ 值 train 出來的 model 在 training set 與 validation set 預測的成效，我將 5652 筆 training set 再細分，前 3 分之一和後 3 分之一為 training set，而中間 3 分之一為 validation set，並將 λ 設為 0、0.1、1、10、50、100 分別進行 training (0 表示沒有做 regularization)，而 gradient descent 設定執行 3000 次迴圈，如下圖所示，紅色線條為 validation set，藍色線條為 training set。