**Machine Learning HW1**

土木五B08501011何子勤

**Programming part:**

1. 解釋什麼樣的data preprocessing可以improve你training/testing accuracy。請提供數據(例如 kaggle public score RMSE)以佐證你的想法。

挑選可能與PM2.5相關的數據，從數值組成的長條圖挑選和PM2.5相似的數據組，再挑出粒子類的項目，最後由繪製折線圖觀察與未來ＰＭ2.5相似的項目。



1. 請實作 2nd-order polynomial regression model (不用考慮交互項)。(1%)

貼上 polynomial regression 版本的 Gradient descent code 內容

for num in range(epoch):

for b in range(int(x.shape[0]/batch\_size)):

t+=1

x\_batch = x[b\*batch\_size:(b+1)\*batch\_size]

y\_batch = y[b\*batch\_size:(b+1)\*batch\_size].reshape(-1,1)

# Prediction of linear regression

pred = np.dot(np.square(x\_batch), w2) + np.dot(x\_batch, w1) + bias

# loss

loss = y\_batch - pred

# Compute gradient

g\_t1 = np.dot(x\_batch.transpose(),loss) \* (-2)

g\_t2 = np.dot(np.square(x\_batch).transpose(),loss) \* (-2)

g\_t\_b = loss.sum(axis=0) \* (-2)

m\_t1 = beta\_1\*m\_t1 + (1-beta\_1)\*g\_t1

m\_t2 = beta\_1\*m\_t2 + (1-beta\_1)\*g\_t2

v\_t1 = beta\_2\*v\_t1 + (1-beta\_2)\*np.multiply(g\_t1, g\_t1)

v\_t2 = beta\_2\*v\_t2 + (1-beta\_2)\*np.multiply(g\_t2, g\_t2)

m\_cap1 = m\_t1/(1-(beta\_1\*\*t))

m\_cap2 = m\_t2/(1-(beta\_1\*\*t))

v\_cap1 = v\_t1/(1-(beta\_2\*\*t))

v\_cap2 = v\_t2/(1-(beta\_2\*\*t))

m\_t\_b = 0.9\*m\_t\_b + (1-0.9)\*g\_t\_b

v\_t\_b = 0.99\*v\_t\_b + (1-0.99)\*(g\_t\_b\*g\_t\_b)

m\_cap\_b = m\_t\_b/(1-(0.9\*\*t))

v\_cap\_b = v\_t\_b/(1-(0.99\*\*t))

# Update weight & bias

w1 -= ((lr\*m\_cap1)/(np.sqrt(v\_cap1)+epsilon)).reshape(-1, 1)

w2 -= ((lr\*m\_cap2)/(np.sqrt(v\_cap2)+epsilon)).reshape(-1, 1)

bias -= (lr\*m\_cap\_b)/(math.sqrt(v\_cap\_b)+epsilon)

(b) 在只使用 NO 數值作為 feature 的情況下，紀錄該 model 所訓練出的 parameter 數值(w2, w1, b)以及 kaggle public score.



