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 P M2.5 相似的項目。

2. 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.


```
w2:
[[0.00932009]
 [0.00618544]
 [0.00431545]
 [0.00440361]
 [0.00486678]
 [0.00545708]
 [0.00825261]
 [0.0139166 ]]
w1:
[[0.13713366]
 [0.13115987]
 [0.12713296]
 [0.12231566]
 [0.12191603]
 [0.12310089]
 [0.12555388]
 [0.13238816]]
bias:
[0.36294667]
```