Machine Learning Spring 2020
Assignment 1 Report

Ali Khalid
MSDS21001

March 31, 2022

0.1 Task 01: Train/Validation/Test Split

- **Briefly describe the outlier removal technique you used (if any)**.

  I have used the z-score normalization technique. In this technique we find the mean and standard deviation of each column(feature) in our data. Now for all the values in the column we find the z-score using the mean and standard deviation of the respective column. The formula to calculate z-score is given below.

  $$Z(x) = \frac{x - mean(X)}{std(X)}$$

  Here 'x' represents the value of the feature for one data point and X represent the the list of value of the respective feature for all data points.

  The z-score converts the data to normal distribution. I have marked all the data points with z-value greater than 3 or less than -3 as outliers.

0.1.1 Task 02: Linear Regression with One Variable

- **Describe any important decision you made and any interesting insights you gained.**

  Following are some of the decisions I made during this task

  - I normalized the data to avoid error because of large values of loss. to perform normalization I used Z-score method.
  - I didn't included the outliers in the training, validation or testing data.
  - I experimented with various values of learning rate and epochs but in the notebook I used the values mentioned in the assignment.

  Following are some of the insights I gained during this task

  - large values of feature require more memory to store loss history. It is better to normalize data to avoid memory issues.

- **Report the parameters which gave the minimum loss along with values of slope, intercept, training loss, validation loss and test loss.**

  - Best parameters:
    m (slope) = 0.8024303555851369
    c (intercept) = 0.13894580918323693
  - Train loss at these parameters: 0.6442186709459236
  - Validation loss at these parameters: 0.6701309238646099
  - Test loss at these parameters: 0.6750152448023103

- **Comment on training and validation loss curves.**

  These curves depends on the initial value of parameters and random distribution of data into training, validation and test set. Mostly train loss is less than validation and test loss but sometimes the situation is reversed. With the increase in epoch number, all the loss values decrease. For me, the case of overfitting never happened because I used epoch numbers equal to 100 which is very less.

0.1.2 Task 03: Linear Regression with Multiple Variables

- **Describe any important decision you made and any interesting insights you gained.**

  Following are some of the decisions I made during this task

  - I didn't included the outliers in the training, validation or testing data.
  - I experimented with various values of learning rate and epochs but in the notebook I used the values mentioned in the assignment.

  Following are some of the insights I gained during this task

  - Large values of feature require more memory to store loss history. It is better to normalize data to avoid memory issues.
  - Normalization is also required if we want to give equal importance to all the features. If we don't do normalization, features having values at higher scale will be given more importance. Therefore, all the algorithms using some sort of distance metric must normalize data.

- **Report the parameters which gave the minimum loss along with values of slope, intercept, training loss, validation loss and test loss.**

  - **Without Normalization**

    * Best parameters:
      c (intercept) = 0.547869
      m (slopes) = 13.8379712, 0.71665652, 0.2253491, 3.89487018
    * Train loss at these parameters: 6376556550.137805
    * Validation loss at these parameters: 6376556550.137805
    * Test loss at these parameters: 7294261039.283305

  - **With Normalization**

    * Best parameters:
      c (intercept) = 0.64651867
      m (slopes) = 0.95205173, 0.67138062, 0.24190606, 0.89115124
    * Train loss at these parameters: 1.1174994105792841
    * Validation loss at these parameters: 1.1399414257814546
    * Test loss at these parameters: 1.1185733858825293

- **Comment on training and validation loss curves.**

  These curves depends on the initial value of parameters and random distribution of data into training, validation and test set. Mostly train loss is less than validation and test loss but sometimes the situation is reversed. With the increase in epoch number, all the loss values decrease. For me, the case of overfitting never happened because I used epoch numbers equal to 100 which is very less.

  If we use normalization, then there are chances that we reach the minimum value with less number of epochs.

- **Which method gave you the best results with normalization or without normalization? And why? Briefly explain.**

  Normalization method gave better results. This is because we use a euclidean distance in our loss function which is sensitive to to scales of values. If the features are closed to each other but have high values, their difference will also be higher and vice versa. Therefore, the gradient vector will point in the direction where features with high values are decreasing as they cause the maximum decrease in loss. Hence, the features with values on higher scale will be given more importance comparative to other features. This is an unwanted process which can be avoided using normalization and hence better results can be acheived.