NYCU Introduction to Machine Learning, Homework 1

[111550113], [謝詠晴]

Part. 1, Coding (60%):

(10%) Linear Regression Model - Closed-form Solution

1. (10%) Show the weights and intercepts of your linear model.

```
2024-10-02 14:47:07.710 | INFO     | __main__:main:85 - LR_CF.weights=array([2.8491883 , 1.0188675 , 0.48562739, 0.1937254 ]),
 LR_CF.intercept=-33.8223
```

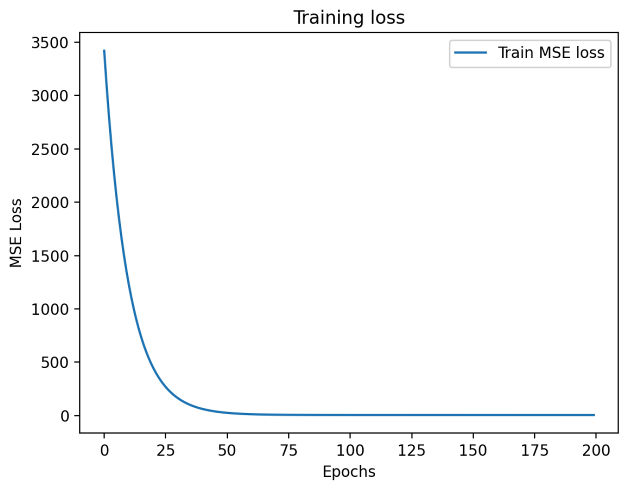
(40%) Linear Regression Model - Gradient Descent Solution

2. (10%)
   - Show the hyperparameters of your setting (e.g., learning rate, number of epochs, batch size, etc.).
   - Show the weights and intercepts of your linear model.

```
losses = LR_GD.fit(train_x, train_y, learning_rate=0.025, epochs=200)
```

```
2024-10-02 14:47:09.262 | INFO     | __main__:main:90 - LR_GD.weights=array([2.84908167, 1.01883135, 0.4856209 , 0.19374889]),
 LR_GD.intercept=-33.8212
```

3. (10%) Plot the learning curve. (x-axis=epoch, y-axis=training loss)



4. (20%) Show your MSE.cf, MSE.gd, and error rate between your closed-form solution and the gradient descent solution.

```
2024-10-02 14:47:09.271 | INFO     | __main__:main:99 - Mean prediction difference: 0.0019
2024-10-02 14:47:09.276 | INFO     | __main__:main:104 - mse_cf=4.1997, mse_gd=4.1999. Difference: 0.003%
```

(10%) Code Check and Verification

5. (10%) Lint the code and show the PyTest results.

```
(ml) PS C:\Users\yungching\Desktop\ML\HW1\release> pytest ./test_main.py -s
>>
==================================================== test session starts ====================================================
platform win32 -- Python 3.11.9, pytest-7.4.4, pluggy-1.0.0
rootdir: C:\Users\yungching\Desktop\ML\HW1\release
collected 2 items

test_main.py 2024-10-02 14:51:33.706 | INFO     | test_main:test_regression_cf:27 - model.weights=array([[3.]]), model.interce
pt=array([4.])
.2024-10-02 14:51:34.657 | INFO     | test_main:test_regression_gd:39 - model.weights=array([2.99999751]), model.intercept=3.9
99996678539041
.

===================================================== 2 passed in 1.52s =====================================================
```

Part. 2, Questions (40%):

1. (10%) How does the presence of outliers affect the performance of a linear regression model? How should outliers be handled? <u>List at least two methods</u>.

   Outliers can greatly affect the regression outcome, such as distorting the regression line, altering the slope and intercept, and thus providing a misleading representation of the relationship between variables.

   1) The simplest way to handle outliers is to remove them from the dataset if they are errors or irrelevant to the analysis.
   2) Another way is to perform transformations such as logarithmic to minimize the influence of outliers and make the data distribute more normally.
   3) Moreover, we can use robust regression methods like RANSAC (Random Sample Consensus) or Huber regression.
      a) RANSAC iteratively selects random subsets of data to fit a model, focusing on the inliers and ignoring outliers.
      b) Huber regression blends linear regression and robust regression by using least squares for small errors and switching to absolute error for larger deviations (when the error exceeds the threshold).

2. (15%) How do different values of learning rate (too large, too small…) affect the convergence of optimization? <u>Please explain in detail</u>.

   The learning rate controls the extent to which we adjust the weights according to gradient descent. When the learning rate is too large, the adjustment will be too much each time, which may potentially prevent convergence to optimal value. Instead, an excessively large learning rate can cause the weights to continually follow the latest gradient, leading to divergence instead of convergence. On the other hand, when the learning rate is too small, the convergence will be really slow and thus require many iterations to reach the intended outcome.

3. (15%)

- What is the prior, likelihood, and posterior in Bayesian linear regression. [Explain the concept in detail rather than writing out the mathematical formula.]

- What is the difference between Maximum Likelihood Estimation (MLE) and Maximum A Posteriori Estimation (MAP)? (Analyze the assumptions and the results.)

1) Prior, likelihood, and posterior:
    - Prior means our initial belief about the parameters before observing any data. Therefore, it's irrelevant to the data. It's more like an estimation based on our previous experiences such as analysis based on historical records.
    - Likelihood is the probability of observing the true data given a set of parameters. If the probability is higher, it means that the estimation is closer to the true data.
    - Posterior probability indicates the probability of the parameters θ occurring given a specific dataset D and hyperparameters m. It combines the prior and the likelihood to provide a comprehensive view of the parameter based on both prior knowledge and evidence from the data.
2) The most significant difference between MLE and MAP is that MLE only focuses on maximizing the likelihood of the observed data without considering prior. On the other hand, MAP considers the combination of likelihood and prior knowledge, and thus provides a more robust estimate.
    [MLE]
    - Assumptions: MLE assumes that the primary source of information for estimating is the data and that all model weight parameters have an equal probability of occurrence.
    - Results: It yields parameters that maximize the likelihood of the observed data but it performs badly when the data is scarce, noisy or contains outliers. It seeks to find a set of θ that increases the probability of the model generating the data.

    [MAP]
    - Assumptions: MAP assumes that the prior provides information that can result in better estimation.
    - Results: It maximizes the posterior distribution and provides more robust estimation especially when the data is limited. MAP directly asks which values of θ maximize the probability given certain conditions according to Bayes' theorem.