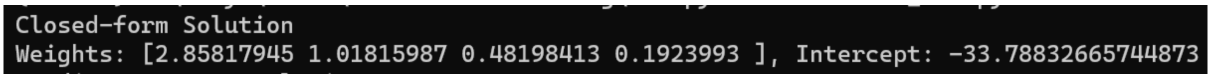
NYCU Introduction to Machine Learning, Homework 1

110550167, 侯博軒

Part. 1, Coding (50%):

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

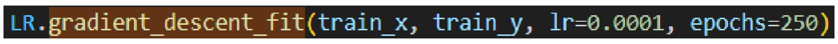
1. (10%) Show the weights and intercepts of your linear model.



```
Closed-form Solution
Weights: [2.85817945 1.01815987 0.48198413 0.1923993 ], Intercept: -33.78832665744873
```

(40%) Linear Regression Model - Gradient Descent Solution

2. (0%) Show the learning rate and epoch (and batch size if you implement mini-batch gradient descent) you choose.

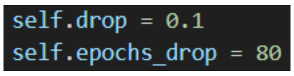


```
LR.gradient_descent_fit(train_x, train_y, lr=0.0001, epochs=250)
```

Learning Rate: 0.0001

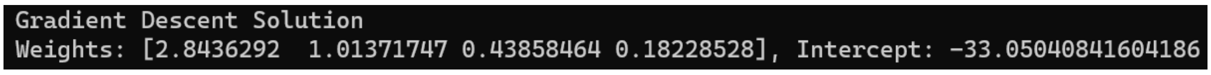
Epochs: 250

Also, I applied step decay on learning rate scheduling with parameters:



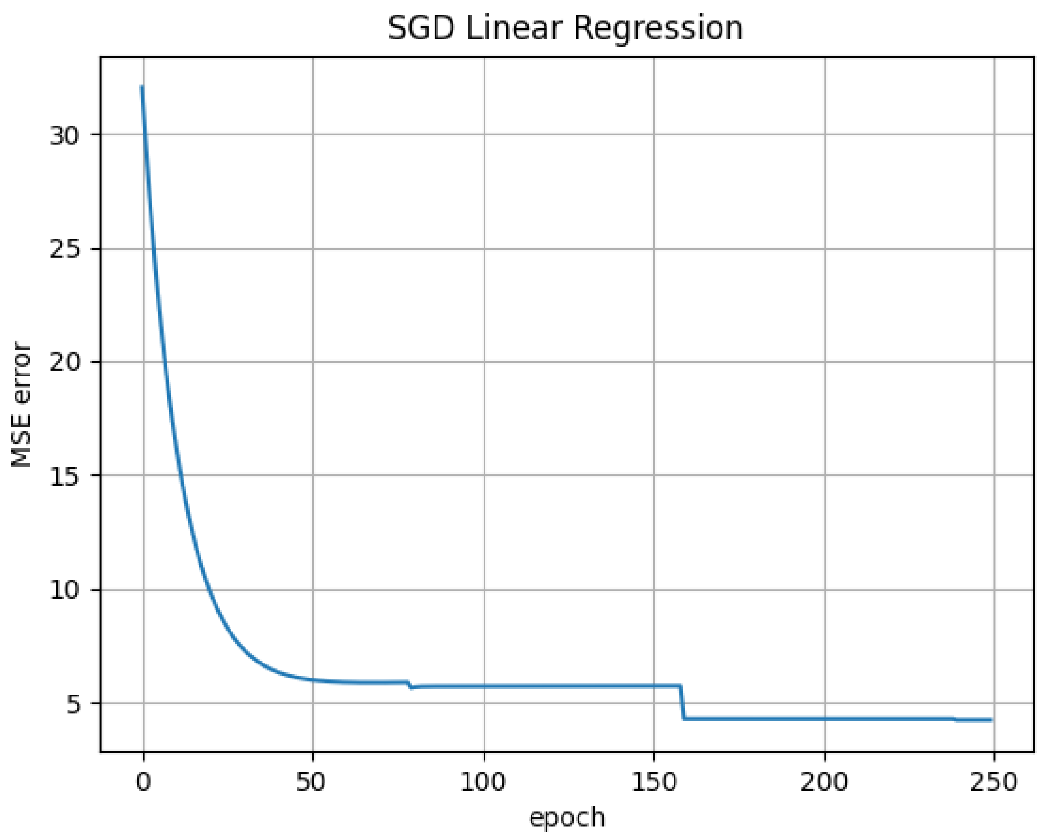
```
self.drop = 0.1
self.epochs_drop = 80
```

3. (10%) Show the weights and intercepts of your linear model.

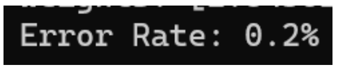


```
Gradient Descent Solution
Weights: [2.8436292  1.01371747 0.43858464 0.18228528], Intercept: -33.05040841604186
```

4. (10%) Plot the learning curve. (x-axis=epoch, y-axis=training loss)



5. (20%) Show your error rate between your closed-form solution and the gradient descent solution.



```
Error Rate: 0.2%
```

Part. 2, Questions (50%):

1. (10%) How does the value of learning rate impact the training process in gradient descent? Please explain in detail.

    Learning rate controls the step taken in the Gradient Descent process. In other words, by controlling the learning rate parameter we can control the speed of convergence of the iterative method, in this case Gradient Descent. With a large learning rate, the models takes large steps in the opposite direction of the gradient, potentially leading us faster to model convergence. With a small learning rate, the model takes small steps in the iterative process, leading to a costly convergence process but with smaller error term from the lost function.

2. (10%) There are some cases where gradient descent may fail to converge. Please provide at least two scenarios and explain in detail.

   There are 2 parameters that we can manipulate in the SGD method, the below 2 cases are both related to bad parameter choice:

   small epoch + small learning rate: With a small learning rate the iteration to convergence must be large enough for the parameter vector to converge, thus when the epoch is smaller than the iterations needed for the model to converge under the certain learning rate gradient descent will fail to converge.
   Large learning rate: When the learning rate is too large, gradient descent can suffer from divergence. This means that weights increase exponentially, resulting in exploding gradients which can cause problems such as instabilities and overly high loss values.

3. (15%) Is mean square error (MSE) the optimal selection when modeling a simple linear regression model? Describe why MSE is effective for resolving most linear regression problems and list scenarios where MSE may be inappropriate for data modeling, proposing alternative loss functions suitable for linear regression modeling in those cases.

   "Yes, MSE is the optimal choice when modeling a simple linear regression model. When we do not have specific information about the training data, we typically assume that errors follow a normal distribution with a mean of 0 and constant variance. Under these conditions, minimizing MSE is equivalent to maximizing the likelihood of the training data. MSE also offers benefits such as computational simplicity, differentiability, and convexity, which make it well-suited for optimization methods like Gradient Descent.

   There are two cases in which MSE may not be the optimal choice. First, when the observed data contains outlier data points that could significantly influence the model, Huber loss may be a better alternative to MSE. Huber loss strikes a balance between absolute loss and MSE, making the model less sensitive to extreme values. Second, when the assumption of constant error variance across all data points is violated, MSE may not be the best choice (as it assumes constant variance). In such cases, using weighted least squares, where observations are assigned weights based on their estimated variances, may be a more suitable choice for the loss function."

4. (15%) In the lecture, we learned that there is a regularization method for linear regression models to boost the model's performance. (p18 in linear_regression.pdf)

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

4.1. (5%) Will the use of the regularization term always enhance the model's performance? Choose one of the following options: "Yes, it will always improve," "No, it will always worsen," or "Not necessarily always better or worse."

Not necessarily always better or worse. Regularization is primarily used to prevent overfitting by adding a penalty term to the loss function, which discourages the model from fitting noise in the data. However, if the model is under fit, adding too much regularization can lead to underperformance, as it restricts the model's ability to fit the training data well.

4.2. We know that λ is a parameter that should be carefully tuned. Discuss the following situations: (both in 100 words)

4.2.1. (5%) Discuss how the model's performance may be affected when λ is set too small. For example, λ=10^(-100) or λ=0

4.2.2. (5%) Discuss how the model's performance may be affected when λ is set too large. For example, λ=1000000 or λ=10^100

(1) λ will not be able to play an important enough factor in the error term. Reducing the regularized result to a result similar to the un-regularized counterpart.

(2) λ will lead the gradient too much further towards minimizing the norm of the weight vector. Instead of fitting the data points, the parameters will tend to converge to 0.