**Machine Learning - Homework 2**  
- Logistic Regression -

ChangYoon Lee

Dankook University, Korea Republic

32183641@gmail.com

**Abstract.** Linear regression is one of the simplest prediction models in machine learning algorithms. The following model fits the best model by changing the coefficients and the intercept variables of the linear model. Two main approaches are used to implement the linear model, and these methods are called gradient descent and least square. Also, the error of the model is evaluated by the error function, which is the mean squared error (MSE). In this paper, we will explain the concepts that are used in the linear regression model, and then present the implemented code for tasks 1 and task 2. Task 1 represents the gradient descent method, and task 2 represents the least square method. After explaining the implemented codes, we will compare the generated prediction models with the built-in libraries.

**Keywords:** Linear Regression, Gradient Descent, Normal Equation, Least Square, Mean Squared Error, MSE

# Introduction

Linear regression is used to predict the value of a variable based on the value of another variable. The variable that we want to predict is called the dependent variable. The variable that we are using to predict the other variables are called the independent variable (About linear regression).

In this form of operation, independent variables are chosen that best predict the value of the dependent variable, based on the coefficients of the linear equation. It minimizes discrepancies between the predicted and actual output values by fitting a straight line or surface. To perform the linear regression, we use two main concepts used in building the linear regression model: gradient descent and least square. Also, the error of the prediction by using the following prediction model, which is called cost, is calculated by using the mean squared error (MSE) formula.

In this paper, we will discuss the basic concepts that are used to implement the linear regression model: gradient descent and least square. Then, we will discuss how we evaluate the performance of the prediction model by evaluating the error: mean squared error (MSE). After discussing all the concepts, we will present the implemented python code for task 1 and task 2, which represents each model that applies gradient descent and the least square method of the linear regression model. In the result, we will show the prediction graph according to the given data set and the coefficients for the predicted model. Also, there will be comparisons between the built-in linear regression models and the implemented models.

# Concepts

## Logistic Regression

In the linear regression model, the cost function is usually calculated through the mean squared error (MSE). Figure 1 presents the following cost function. The main goal of implementing the linear regression model is to find the best coefficients that minimize the value of the cost function. The approaches to finding these coefficients end up in the method, such as gradient descent and the least square method.

## Sigmoid Function

The least square method is a powerful and simple method to get the coefficients of the linear regression model. By simply processing the matrix operations, we can get the best coefficients and intercepts for the following model. However, the least square method is not always available to get the prediction model. If the matrix of the given data set is not invertible, such situations where redundant features are in the matrix, or the size of the rows is smaller than the size of the columns, we cannot apply the least square method to implement the linear regression model.

## Logistic, Hyperbolic Functions

The last consideration is the local minimum. The local minimum becomes a problem when the gradient recognizes some minimum point is the optimal point, which is not, and stops the operation so that it cannot reach the global minimum. This problem is affected by the starting point of the operation, too small a learning rate, and the batch size. Therefore, by setting the right direction for the operation, proper step size, and using SGD, we can resolve the following consideration.

## Decision Boundary

The last consideration is the local minimum. The local minimum becomes a problem when the gradient recognizes some minimum point is the optimal point, which is not, and stops the operation so that it cannot reach the global minimum. This problem is affected by the starting point of the operation, too small a learning rate, and the batch size. Therefore, by setting the right direction for the operation, proper step size, and using SGD, we can resolve the following consideration.

## Cross-Entropy Function

The last consideration is the local minimum. The local minimum becomes a problem when the gradient recognizes some minimum point is the optimal point, which is not, and stops the operation so that it cannot reach the global minimum. This problem is affected by the starting point of the operation, too small a learning rate, and the batch size. Therefore, by setting the right direction for the operation, proper step size, and using SGD, we can resolve the following consideration.

# Implementation

## Representation of Given Data Set

Each step of the code follows the formula operation which is presented in section 2.3. By using the following code, we can predict with the linear regression model for the given data set by applying the gradient descent method.

## Task 1: Train a Classifier with the Train Data Set

Each step of the code follows the formula operation which is presented in section 2.3. By using the following code, we can predict with the linear regression model for the given data set by applying the gradient descent method.

## Task 2: Classify the Data in the Test Data Set using the Trained Model

Each step of the code follows the formula operation which is presented in section 2.2. By using the following code, we can predict with the linear regression model for the given data set by applying the least square method.

## Extra: Classification with the Polynomial Logistic Regression Model

Each step of the code follows the formula operation which is presented in section 2.2. By using the following code, we can predict with the linear regression model for the given data set by applying the least square method.

# Build Environment

The following build environments are required to execute the implemented code for task 1 and task 2.

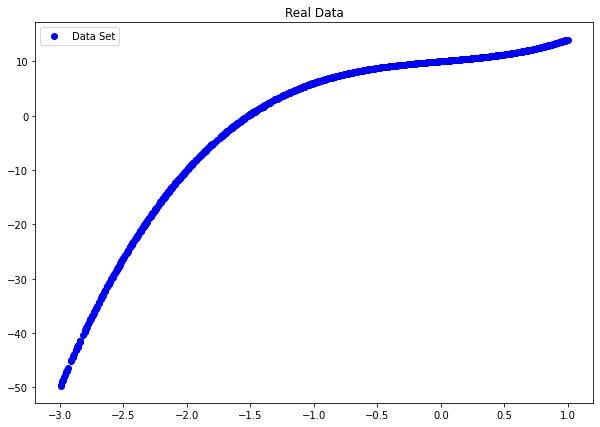
Building Environment:

* The environment that can open the ipython notebook: COLAB, Jupyter Notebook
* Each cell in the notebook should be executed sequentially.

File Upload: Before cell execution, the given data set must be uploaded first.

# Results

* Visualization of the original given data set of x and y



# Evaluation

Figures 18 and 19 show the implemented linear regression model of task 1 and task 2, which are each implemented by our code and the sci-kit learn library. As we can see, the coefficients and the intercept are the same, which means that the implemented linear regression models are successfully generated. Also, the linear regression model of task 2 shows a better performance than the model of task 1. It means that as the number of coefficients increases in the linear regression model, the performance gets better. In short, we generated the fine-granted linear regression model, and the model of task 2, which is polynomial, shows better performance than the model of task 1, which is linear.

# Conclusion

By understanding this paper, we can understand the basic concepts of the linear regression model and the formulas that are used to implement the following model: gradient descent, least square, and the mean square error (MSE). Also, we can understand the considerations that occur by applying the following concept, which is mainly about gradient descent.

# Citations

[1] *About linear regression*. IBM. (n.d.). Retrieved September 20, 2022, from [https://www.ibm.com/topics/linear-regression](https://www.ibm.com/topics/linear-regression%20)

[2] Lee, K. H. (n.d.). [Slide] 02. Linear Regression.

[3] *Sklearn.linear\_model.linearregression*. scikit. (n.d.). Retrieved September 21, 2022, from [https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LinearRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html%20)