Gradient Descent for Linear Regression: Problem, Solution, and Implementation

Raghda Al Taei

2024

1 Introduction

In machine learning, linear regression is a foundational technique used to predict a dependent variable based on one or more independent variables. However, finding the optimal parameters for the model can be challenging, particularly when dealing with large datasets. This is where the Gradient Descent algorithm comes into play. In this report, we will explore the problem of linear regression, how Gradient Descent provides a solution, and break down the implementation of this approach in MATLAB.

2 The Problem: Linear Regression

Linear regression aims to model the relationship between input features (independent variables) and a target variable (dependent variable). The primary goal is to find the best-fitting line that minimizes the difference between the predicted and actual values of the target variable.

For example, consider a simple linear regression model represented by the equation:

$$y = \theta_0 + \theta_1 x \tag{1}$$

Where:

- y is the predicted value.
- θ_0 is the intercept (bias term).
- θ_1 is the slope (coefficient).
- x is the input feature.

To assess how well our model performs, we use a cost function, often the Mean Squared Error (MSE), defined as follows:

$$J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{2}$$

Where:

- m is the number of training examples.
- $h_\theta(x^{(i)})$ is the hypothesis function (predicted value).
- $y^{(i)}$ is the actual target value.

The goal is to find the parameters θ_0 and θ_1 that minimize the cost function J.

3 Solution: Gradient Descent

Gradient Descent is an optimization algorithm used to minimize the cost function by iteratively adjusting the parameters in the direction of the steepest descent. The update rule for the parameters is as follows:

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} \tag{3}$$

Where:

- α is the learning rate, determining the size of the steps taken towards the minimum.
- $\frac{\partial J(\theta)}{\partial \theta_j}$ is the gradient of the cost function with respect to the parameter θ_j.

The process continues until the algorithm converges, meaning the changes in the cost function become negligible or a predetermined number of iterations is reached.

4 Implementation in MATLAB

The following MATLAB code implements Gradient Descent for linear regression:

4.1 Code Explanation

- The code begins by loading the dataset, which contains the independent variable x and the dependent variable y.
- It sets the learning rate and the number of iterations for the Gradient Descent algorithm.
- The parameters θ are initialized to zero.

- In each iteration, the hypothesis function h is computed, and the parameters are updated using the Gradient Descent update rules.
- Finally, the estimated parameters are displayed.
- GitHub Repository Code

5 Conclusion

Gradient Descent is an effective optimization technique for training linear regression models. By iteratively adjusting the parameters based on the gradients of the cost function, we can find the optimal values that minimize the prediction error. The provided MATLAB code serves as a practical implementation of this approach, demonstrating how to solve linear regression problems through Gradient Descent.