Under the Hood: Linear Regression using Gradient Descent

Tejinder Singh Hunjan, Mohit Kamkhalia, Sashank Mishra

Introduction

Linear Regression is one of the simplest yet most powerful techniques in supervised learning. Instead of using built-in libraries, we explore how it works mathematically by implementing Gradient Descent from scratch.

1. Hypothesis Function

The goal is to fit a line:

$$\hat{y} = mx + b$$

2. Loss Function (Mean Squared Error)

The Mean Squared Error is used to measure the quality of predictions:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))^2$$

3. Partial Derivatives for Gradient Descent

To minimize the MSE, we compute partial derivatives with respect to the parameters m and b:

$$\frac{\partial \text{MSE}}{\partial m} = -\frac{2}{n}\sum_{i=1}^{n} x_i(y_i - (mx_i + b))$$

$$\frac{\partial \text{MSE}}{\partial b} = -\frac{2}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))$$

4. Update Rules

Using a learning rate α, we update the parameters iteratively:

$$m := m - \alpha \cdot \frac{\partial \text{MSE}}{\partial m}$$

$$b := b - \alpha \cdot \frac{\partial \text{MSE}}{\partial b}$$

5. Intuition

With each update, the model slowly converges towards values of m and b that minimize the total error, creating the best-fit line.

Conclusion

Building Linear Regression from scratch with Gradient Descent gives deeper insights into how ML models learn. It's not just about fitting lines—it's about mastering optimization.