Mathematics Behind Linear Regression

Ankit Shrestha

17 September 2025

1 Problem Setup

Linear regression is one of the simplest and most widely used machine learning models. The main idea is to describe the relationship between the input variables (also called features) and the output variable (also called the target) using a straight line.

For example, suppose we want to predict a house price based on its size. A simple linear regression model would try to fit a straight line through the data points such that:

$$\hat{y} = \theta_0 + \theta_1 x$$

Here:

- $\hat{y}$ is the predicted value (e.g., predicted house price),
- θ_0 is the intercept (the value of $\hat{y}$ when $x = 0$),
- θ_1 is the slope, which tells us how much y changes for a unit change in x.

When we have more than one feature, we call it multiple linear regression. The equation extends to:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

This means that the prediction is now a weighted sum of all the input features, plus the intercept.

2 Cost Function

The goal of linear regression is to find the line (or hyperplane) that best fits the data. But how do we define "best"? We measure it by how close the predictions $\hat{y}$ are to the actual values y.

The difference between $\hat{y}$ and y is called the error or residual. To measure overall performance, we square these errors and take the average. This gives us the **Mean Squared Error (MSE)**:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

where:

- m is the number of training examples,
- $y^{(i)}$ is the actual output of the i^{th} training example,
- $\hat{y}^{(i)}$ is the predicted value for the i^{th} training example.

The lower the cost function $J(\theta)$, the better the model fits the data.

3 Optimization

Now that we have defined the cost function, the next step is to minimize it, i.e., find the values of $\theta_0, \theta_1, \ldots, \theta_n$ that give the smallest error. There are two main ways to do this:

a) Normal Equation

The Normal Equation provides a direct mathematical formula for finding the best values of θ:

$$\theta = (X^T X)^{-1} X^T y$$

Here, X is the matrix of input features and y is the vector of target values. The equation gives us the exact solution without iteration. However, it can be computationally expensive for very large datasets because it involves inverting a matrix.

b) Gradient Descent

Gradient descent is an iterative optimization method. Instead of computing the solution directly, we start with some initial guess for θ and gradually improve it.

The rule for updating each parameter θ_j is:

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

Here:

- α is the learning rate, which controls the size of each step,
- $\frac{\partial}{\partial \theta_j} J(\theta)$ is the derivative of the cost function with respect to θ_j.

For linear regression, this derivative works out to:

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{2}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) x_j^{(i)}$$

So at each step, we adjust the parameters slightly in the direction that reduces the error. Over many iterations, the parameters converge to values that minimize the cost function.

4 Model Evaluation

Once the model is trained, we need to check how good it is. For this, we use evaluation metrics:

- **Mean Squared Error (MSE):** This is the same cost function we minimized, measuring average squared error.
- **Root Mean Squared Error (RMSE):** This is simply the square root of MSE. It has the advantage of being in the same units as the target variable, making it easier to interpret.
- **Coefficient of Determination (R^2):** This tells us how much of the variation in the target variable is explained by the model:

$$R^2 = 1 - \frac{\text{Sum of Squared Errors}}{\text{Total Sum of Squares}}$$

  An R^2 value close to 1 means the model explains most of the variation in the data, while a value close to 0 means it does not explain much.

5 Summary

In simple terms, linear regression is about drawing the best line through data points so that predictions are as close as possible to actual values. We measure closeness using a cost function (MSE), and we find the best line using optimization techniques like the Normal Equation or Gradient Descent. Finally, we check how well the model works using metrics such as MSE, RMSE, and R^2.