Chapter 1

Linear Regression and Gradient Descent

Linear regression models can be fitted using an optimization algorithm known as gradient descent. This is especially useful when the number of features is large, making the normal equation computationally expensive.

In gradient descent, we start with an initial guess for the model parameters and iteratively update these parameters to minimize the cost function, which is usually the mean squared error (MSE) for linear regression. For a linear regression model with parameters θ, the update rule is given by

$$\theta := \theta - \alpha \nabla J(\theta)$$

where α is the learning rate and $\nabla J(\theta)$ is the gradient of the cost function evaluated at θ. For MSE, the gradient is given by

$$\nabla J(\theta) = \frac{2}{n}\mathbf{X}^T(\mathbf{X}\theta - \mathbf{y})$$

where $\mathbf{X}$ is the feature matrix, $\mathbf{y}$ is the vector of target values, and n is the number of observations.

1.1 Standardizing Inputs

Standardizing inputs can improve the performance of gradient descent. By ensuring all features have a similar scale, we can avoid a situation where the cost function has a very elongated shape, causing gradient descent to take a long time to converge.

More specifically, standardization transforms the features so they have a mean of 0 and a standard deviation of 1. This is done by subtracting the mean and dividing by the standard deviation for each feature:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i}$$

where x_i is a feature vector, and μ_i and σ_i are its mean and standard deviation, respectively.

By standardizing the inputs, each feature contributes approximately proportionately to the final distance, helping the gradient descent algorithm converge more quickly and efficiently.

Appendix A

Answers to Exercises

Index