Mathematical Explanation for Logistic Regression Implementation

Introduction

This document explains the mathematical foundation for a Logistic Regression model, which is a classification algorithm that predicts probabilities using a sigmoid function and optimizes its weights via gradient descent.

Key Components

Sigmoid Function

The sigmoid function maps any real-valued number to a probability range between 0 and 1. It is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

where:

- z: The linear combination of input features and weights, given by $z = Xw + b$.

Optimization Process

The model is optimized using gradient descent by updating the weights and bias iteratively.

Gradient of the Weights w

The gradient with respect to the weights is:

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{1}{n} X^T (\hat{y} - y)$$

where:

- X: Matrix of input features.
- $\hat{y}$: Predicted probabilities.
- y: True labels.

Gradient of the Bias b

The gradient with respect to the bias is:

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)$$

Weight and Bias Updates

The weights and bias are updated as follows:

$$w \leftarrow w - \alpha \frac{\partial \mathcal{L}}{\partial w}$$
$$b \leftarrow b - \alpha \frac{\partial \mathcal{L}}{\partial b}$$

where α is the learning rate.

Prediction

After training, predictions are made using the sigmoid function:

$$\hat{y} = \sigma(Xw + b)$$

The final predicted class labels are assigned based on a threshold (commonly 0.5):

$$y_{\text{pred}} = \begin{cases} 1 & \text{if } \hat{y} > 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

Conclusion

This implementation of Logistic Regression employs the sigmoid function to produce probabilities and iteratively optimizes the model parameters using gradient descent. The resulting model can classify data effectively based on the learned decision boundary.