

Numerical Optimization and Machine Learning: Exploring Techniques Underlying Gradient Descent

Badger Code

July 26, 2023

1 Introduction

Numerical optimization is a critical component of many machine learning algorithms. It involves finding the optimal parameters of a model that minimize or maximize an objective function. This paper examines numerical optimization techniques that underlie various machine learning algorithms, with a specific focus on gradient descent. We will delve into the principles behind gradient descent, explore its variants, and discuss its applications in training machine learning models.

2 Numerical Optimization Basics

Numerical optimization aims to find the minimum or maximum of a function, subject to certain constraints. In the context of machine learning, the objective function typically represents a loss or cost function that quantifies the model's performance.

2.1 Optimization Problem Formulation

Given an objective function $f(\mathbf{w})$, where \mathbf{w} is a vector of model parameters, the optimization problem can be formulated as follows:

$$\min_{\mathbf{w}} f(\mathbf{w})$$

subject to optional constraints.

2.2 Gradient Descent Intuition

Gradient descent is an iterative optimization algorithm that uses the gradient (derivative) of the objective function to update the model parameters in the direction that reduces the function's value. The intuition is to move in the steepest descent direction to reach the local or global minimum.

3 Gradient Descent Algorithm

The basic gradient descent algorithm involves the following steps:

1. Initialize the model parameters \mathbf{w} with random values.
2. Calculate the gradient of the objective function with respect to \mathbf{w} : $\nabla f(\mathbf{w})$.
3. Update the parameters using the gradient and a learning rate α : $\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla f(\mathbf{w})$.
4. Repeat steps 2 and 3 until convergence or a maximum number of iterations is reached.

4 Variants of Gradient Descent

Several variants of gradient descent have been proposed to address certain challenges and improve convergence speed.

4.1 Stochastic Gradient Descent (SGD)

SGD is a variant of gradient descent that updates the model parameters after processing each training example, rather than using the entire dataset at once. This reduces the computational cost and can lead to faster convergence.

4.2 Mini-Batch Gradient Descent

Mini-batch gradient descent is a compromise between batch gradient descent and SGD. It divides the dataset into small batches and updates the model parameters after processing each batch.

4.3 Momentum

Momentum is a technique that accelerates the convergence of gradient descent by introducing a momentum term. It helps the optimization algorithm move more quickly through regions with small gradients.

5 Applications in Machine Learning

Gradient descent is a fundamental optimization technique used in various machine learning algorithms, including:

5.1 Linear Regression

In linear regression, gradient descent is used to find the optimal coefficients that best fit the data to a linear model.

5.2 Neural Networks

Gradient descent is the backbone of training neural networks. It updates the weights and biases to minimize the loss function and make the network learn the desired mapping.

6 Conclusion

Numerical optimization techniques, such as gradient descent, play a crucial role in machine learning algorithms. These techniques allow models to learn from data by adjusting their parameters to minimize the objective function. Gradient descent, with its variants, has proven to be effective in training various machine learning models, from simple linear regression to complex deep neural networks. Understanding numerical optimization is essential for developing efficient and accurate machine learning algorithms and contributes to the continuous advancement of the field.