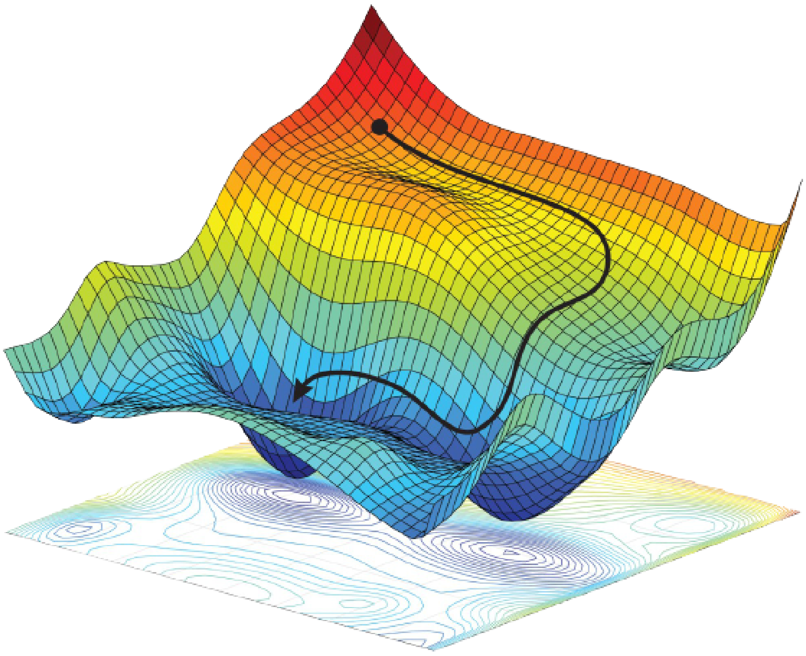

Exploring Optimisation Algorithms and Applications

Project UID : 30

Report

Mentee: **Aboli Ganesh Malshikare**

February 16, 2025

Abstract

Optimisation algorithms are fundamental in solving complex problems across various domains, from engineering and finance to artificial intelligence and logistics. This project delves into different optimisation techniques. The study focuses on understanding their efficiency, applicability, and limitations in real-world scenarios. Through practical implementations and comparative analysis, the project aims to identify the strengths of each algorithm and provide insights into choosing the most effective approach for different problem statements. The findings offer a comprehensive perspective on optimisation strategies and their practical impact..

Contents

1

Details

In this chapter, I have provided my personal and project-related details.

Name:	Aboli Ganesh Malshikare
Roll Number:	23B1211
Department:	Electrical Engg. (B.Tech)
Mentor:	Vishesh Jain (22B2467)
Project:	Exploring Optimisation Algorithms and Applications (UID=30)

Table 1.1: Personal and Project Details

2

Introduction

2.1 Project Overview

Optimization in machine learning involves selecting the best parameters for a model to minimize a given loss function. The efficiency of an optimization algo- rithm affects the model's training time, accuracy, and generalization to unseen data. This report consolidates the core techniques explored in seven labs, focus- ing on theoretical underpinnings, implementation, and practical implications.

2.2 Objectives

The main objectives of the project were:

- To develop a deeper understanding of optimization techniques
- To investigate the effect of learning rate, regularization, and stopping criteria on optimization efficiency.
- To compare accuracy, decision boundaries, and computational efficiency across different optimization methods.
- To demonstrate how optimization techniques improve classification in real-world datasets.
- To learn and overcome the challenges involved in real-world applications.

3

Optimization Techniques and Analysis

3.0.1 Gradient-based Methods

1. Batch Gradient Descent (BGD)

- Uses the entire dataset to compute gradients and update parameters.
- Ensures stable and smooth convergence but is computationally expensive.
- Suitable for convex optimization problems.

2. Stochastic Gradient Descent (SGD)

- Updates parameters after each training example, making it faster than BGD.
- Introduces noise due to frequent updates, leading to fluctuations in convergence.
- Effective for large datasets but requires careful tuning of the learning rate.

3. Mini-batch Gradient Descent (MGD)

- A trade-off between BGD and SGD, updating parameters using small batches.
- Reduces variance while maintaining computational efficiency.
- Commonly used in deep learning applications.

3.0.2 Second-order Optimization Methods

4. Newton's Method

- Uses second-order derivatives (Hessian matrix) for more accurate step sizes.
- Achieves rapid convergence compared to first-order methods.
- Computationally expensive, limiting its use in high-dimensional problems.

5. Limited-memory BFGS (L-BFGS)

- Stores only a few past gradient evaluations to reduce memory usage.
- Maintains efficiency while providing fast convergence in high-dimensional spaces.
- Used in logistic regression and other machine learning models.

3.0.3 Adaptive Learning Rate Methods

6. Adam Optimizer

- Combines momentum-based updates and adaptive learning rates.
- Adapts learning rates dynamically for each parameter.
- Outperforms standard gradient descent in deep learning applications.

3.0.4 Evolutionary and Probabilistic Methods

7. Bayesian Optimization

- Models the objective function using Gaussian Processes.
- Balances exploration and exploitation for efficient hyperparameter tuning.
- Outperforms grid search and random search in complex optimization tasks.

3.1 Comparative Analysis

- **Gradient-based methods** are effective for smooth functions and provide stability, but require tuning learning rates.
- **Second-order methods** converge faster but are computationally expensive.
- **Adaptive learning rate methods** such as Adam improve stability in deep learning applications.
- **Evolutionary and probabilistic approaches** are robust for hyperparameter tuning but can be slow.

4

Experiments and Analysis

The experiments explored a variety of optimization techniques, beginning with gradient-based methods and extending to evolutionary and probabilistic approaches. Gradient descent methods, including Batch Gradient Descent (BGD), Stochastic Gradient Descent (SGD), and Mini-batch Gradient Descent (MGD), were implemented to analyze their effectiveness in minimizing loss functions. While SGD showed faster convergence, it introduced more noise, whereas MGD balanced speed and stability. Newton's method, leveraging second-order derivatives, achieved faster convergence but was computationally expensive.

Advanced optimization techniques such as the L-BFGS algorithm demonstrated efficiency in optimizing logistic regression while requiring less memory. The Adam optimizer proved to be highly effective for deep learning models, surpassing SGD and RMSProp in adaptability and convergence speed. Beyond gradient-based methods, genetic algorithms were employed for hyperparameter tuning in an SVM classifier, effectively exploring the search space but with slower execution. Bayesian optimization using Gaussian Processes was also applied for hyperparameter tuning in a Random Forest model, striking a balance between exploration and exploitation for improved parameter selection.

5

Results and Conclusions

5.1 Results and Observations

Experiments demonstrated the impact of optimization techniques on model performance. Key takeaways include:

- Adaptive learning rate methods generally outperform standard gradient descent in deep learning.
- While gradient-based methods are effective for smooth functions, more advanced methods like L- BFGS and Adam improve convergence efficiency
- Bayesian optimization is superior for hyperparameter tuning due to its efficiency.
- Constraint handling is essential in SVMs and real-world problems with predefined limits.

5.2 Conclusion

This report consolidates insights from seven labs, demonstrating that optimization is a crucial component of machine learning. Different techniques cater to different problem settings, and choosing the right approach significantly influences model convergence, accuracy, and computational efficiency. Future work can explore hybrid methods combining multiple optimization strategies for enhanced performance.