Hyperparameter Optimization Using Various Optimizers

A PROJECT REPORT

Submitted by

KESHAV KUMAR (21SCSE1180171)

JAYANT SIKARWAR (21SCSE1180159)

PROJECT ID: BT40497

Under the guidance of

[Mr. Sugan Patel]

Assistant Professor

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science and Engineering,

Specialization in Artificial Intelligence and Machine Learning

GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

January, 2024

GALGOTIAS UNIVERSITY

BONAFIDE CERTIFICATE

This is to certify that Project report entitled **“Hyperparameter Optimization Using Various Optimizers”** which is submitted by **Keshav Kumar and Jayant Sikarwar** in partial fulfillment of the requirement for the award of degree B. Tech in Department of Computer Science and Engineering of School of Computing Science and Engineering, Galgotias University, Greater Noida, India is a record of the candidate own work carried out by them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Internal Examiner **Signature of Supervisor(s)**

External Examiner **Signature of Program Chair**

Date: January, 2024

Place: Greater Noida

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

Optimizing hyperparameters is essential for improving ML model performance. In this work, we investigate and contrast the performance of multiple optimizers on a particular dataset and task. Their performance in terms of accuracy, convergence speed, and computing efficiency will be assessed in order to shed light on the associated trade-offs. Under carefully monitored experimental conditions, optimizers like Adam, RMSprop, Stochastic Gradient Descent (SGD), and others are examined. Our goal in conducting these assessments is to find trends in their behavior under various circumstances, including variations in learning rates and batch sizes.

Our results show that although some optimizers provide faster convergence, accuracy may suffer, or computing overhead may increase. Others could need more repetitions to get similar results, even though they are computationally lighter. These trade-offs highlight how crucial it is to choose the right optimizer depending on the particular needs of a task. By demonstrating how optimizers operate in various settings using different optimization techniques also, this study advances our understanding of hyperparameter tuning and aids practitioners in making well-informed decisions when creating models. Overall, the findings show that there is no one optimal optimizer that works for all problems; rather, the best option varies depending on the problem's features, the data, and the computational limitations.

ABBREVIATIONS

ML	Machine Learning
DL	Deep Learning
CNN	Convolutional Neural Network
MNIST	Modified National Institute of Standards and Technology (database)
CIFAR	Canadian Institute For Advanced Research
RMSProp	Root Mean Square Propagation
SGD	Stochastic Gradient Descent
Adagrad	Adaptive Gradient Algorithm

CHAPTER 1.
INTRODUCTION

1.1. Identification of issue

With the exponential growth of data and the rise of complex machine-learning models, optimizing model performance has become crucial in various applications. Hyperparameter optimization, specifically choosing the right optimizer, is a contemporary challenge for data scientists and ML engineers. Optimizers directly impact model convergence speed, final accuracy, and generalization, making it a critical aspect of ML projects.[1]

This project addresses the need for an in-depth evaluation of different optimizers to improve model performance regarding efficiency, accuracy, and training time.[2] The project is aimed at ML practitioners and organizations looking to streamline their model training processes.

1.2. Identification of Problem

Selecting the optimal hyperparameters, particularly the optimizer, for ML models is a complex task. Improper selection of hyperparameters can lead to slow training, overfitting, or poor generalization.[1] Currently, trial and error and manual tuning are common approaches, which are inefficient and time-consuming.[3]

1.3. Identification of Tasks

- Research and review of various optimizers used in ML.
- Selection of appropriate datasets for experimentation.
- Implementation of ML models with different optimizers.
- Analysis and comparison of optimizers based on performance metrics.
- Presentation of findings and recommendations based on results.

1.4. Timeline

- **Early Work (2012-2013):** Research during this period primarily focused on Stochastic Gradient Descent (SGD), as it was the standard optimizer for many ML tasks. Improvements such as momentum and Nesterov acceleration were proposed to address SGD's limitations. Researchers like **LeCun, Bottou, and Bengio** explored momentum-based optimizers to speed up convergence in DL models.[4]

- **Adam Optimizer (2014):** The introduction of Adam by **Diederik Kingma and Jimmy Ba** was a breakthrough in optimization methods. Adam combines the advantages of two other popular optimizers, AdaGrad and RMSProp, and quickly gained traction due to its adaptive learning rate capabilities. Studies after its release focused on Adam's effectiveness in training deep neural networks, especially in terms of convergence speed and stability.[5]

- **Bayesian Optimization and Hyperparameter Search (2015-2018):** Researchers like **Snoek, Larochelle, and Adams** explored Bayesian optimization methods for automated hyperparameter tuning. These methods significantly reduced the time required for hyperparameter search, including optimizer selection. By automating the process, they improved model performance without the need for manual tuning.[2]

- **Comparative Studies (2018-2020):** Comparative analyses of optimizers, such as Adam, RMSProp, SGD, and AdaGrad, were conducted by various research groups. These studies showed that no single optimizer was the best across all tasks. For example, **Wilson et al. (2017)** found that while Adam converged faster, SGD with momentum often provided better generalization in certain tasks.[6]

CHAPTER 2.
LITERATURE REVIEW/BACKGROUND STUDY

2.1. Timeline of the reported problem

The concept of hyperparameter optimization has been explored since the early days of ML, but its importance has risen significantly with the advent of DL.[7] Optimizers like SGD have been used since the 1950s, but more recent adaptive optimizers like Adam (2014) have introduced new efficiencies and challenges. The need for efficient hyperparameter tuning has grown as ML models become more complex and resource intensive.[8]

2.2. Existing solutions

Several methods for hyperparameter optimization exist, including manual tuning, grid search, and random search.[9] Additionally, modern approaches such as Bayesian optimization and evolutionary algorithms have gained popularity for automating hyperparameter tuning. However, selecting the right optimizer remains a critical task as it directly impacts model performance. [2]

2.3. Bibliometric analysis

A survey of research publications highlights that Adam and its variants are the most cited optimizers in DL literature, followed by RMSProp and SGD. Research also indicates growing interest in hybrid optimization techniques that combine multiple strategies for better performance.Click or tap here to enter text.

2.4. Review Summary

The literature shows that while adaptive optimizers such as Adam offer faster convergence, simpler optimizers like SGD with momentum tend to generalize better, especially in tasks with large datasets.[11] The trade-off between convergence speed and generalization ability is a key factor in choosing the right optimizer.

2.5. Problem Definition

The problem is the inefficiency of current hyperparameter optimization methods, particularly in selecting the best optimizer. This project aims to compare the performance of several optimizers to provide guidance on choosing the most effective one for different ML tasks.

2.6. Goals/Objectives

- Evaluate the performance of different optimizers (SGD, Adam, RMSProp, AdaGrad) in a variety of machine-learning tasks.
- Compare the optimizers based on convergence rate, final accuracy, generalization ability, and computational cost.
- Recommend the best optimizer for specific tasks based on the results.

CHAPTER 3.
DESIGN FLOW/PROCESS

3.1. Evaluation & Selection of Specifications/Features

Optimization technique used-

Grid Search- Grid Search is an optimization technique used to find the best hyperparameters for a machine learning model. It works by systematically searching through a specified subset of the hyperparameter space. For each combination of hyperparameters, the model is trained, and its performance is evaluated using cross-validation or a similar metric. The combination that yields the best performance is selected.

The optimizers selected for evaluation include:

- **SGD:** A simple and widely used optimizer with momentum.
- **Adam:** An adaptive learning rate method that combines the advantages of two other methods (AdaGrad and RMSProp).
- **RMSProp:** A method that adjusts the learning rate for each parameter based on recent gradients.
- **AdaGrad:** Optimizer that works well with sparse data by adapting the learning rate.

Optimizer	**Convergence Speed**	**Generalization**	**Computational Efficiency**	**Stability**
SGD	Slow	High	Low	Moderate
RMSProp	Fast	Moderate	High	High
Adam	Very Fast	Moderate	High	High
Adagrad	Slow(over time)	Moderate	High	Moderate

Table 3.1.1 Comparison of Optimizer

3.2. Design Constraints

The constraints of the project include:

- **Computational resources**: Limited by available hardware for training models.
- **Time constraints**: Each optimizer is tested on fixed datasets within a limited timeframe.

- **Model complexity**: Datasets include image classification tasks (CIFAR-10, MNIST)

3.3. Analysis of Features and finalization subject to constraints

Based on initial experimentation, the features and optimizers were analysed for:

- **Convergence rate:** The number of epochs required to reach an acceptable loss.
- **Generalization:** The optimizer's ability to avoid overfitting.
- **Training efficiency:** How quickly the optimizer converges to a minimum.

3.4. Design Flow

The design process included the following steps:

- Step 1: Selection of datasets and corresponding models (CNNs for CIFAR-10 and MNIST)
- Step 2: Implementation of each optimizer in different models.
- Step 3: Evaluation of the model's performance using predefined metrics (accuracy, loss, MSE).
- Step 4: Comparison and analysis of the results

3.5. Design selection

The final design selection involved choosing optimizers that performed best across different tasks. Adam was selected as the primary optimizer for deep networks, while SGD with momentum was chosen for tasks requiring better generalization.

3.6. Implementation methodology

The project was implemented using Python and DL libraries like TensorFlow and PyTorch. Each optimizer was evaluated in terms of:

- **Learning rate:** Adjusted for each optimizer to prevent divergence.
- **Batch size:** Kept constant across experiments to ensure fair comparison.
- **Metrics:** Convergence rate, accuracy, and generalization were measured for each optimizer

CHAPTER 4.
RESULTS ANALYSIS AND VALIDATION

4.1. Implementation of solution

Experiments were conducted on three datasets (CIFAR-10, MNIST, and Boston Housing) using various optimizers. The performance metrics recorded include convergence rate, final accuracy (for classification tasks), and mean squared error (for regression tasks). Adam showed the fastest convergence in most cases, followed by RMSProp while SGD with momentum demonstrated better generalization in certain tasks.

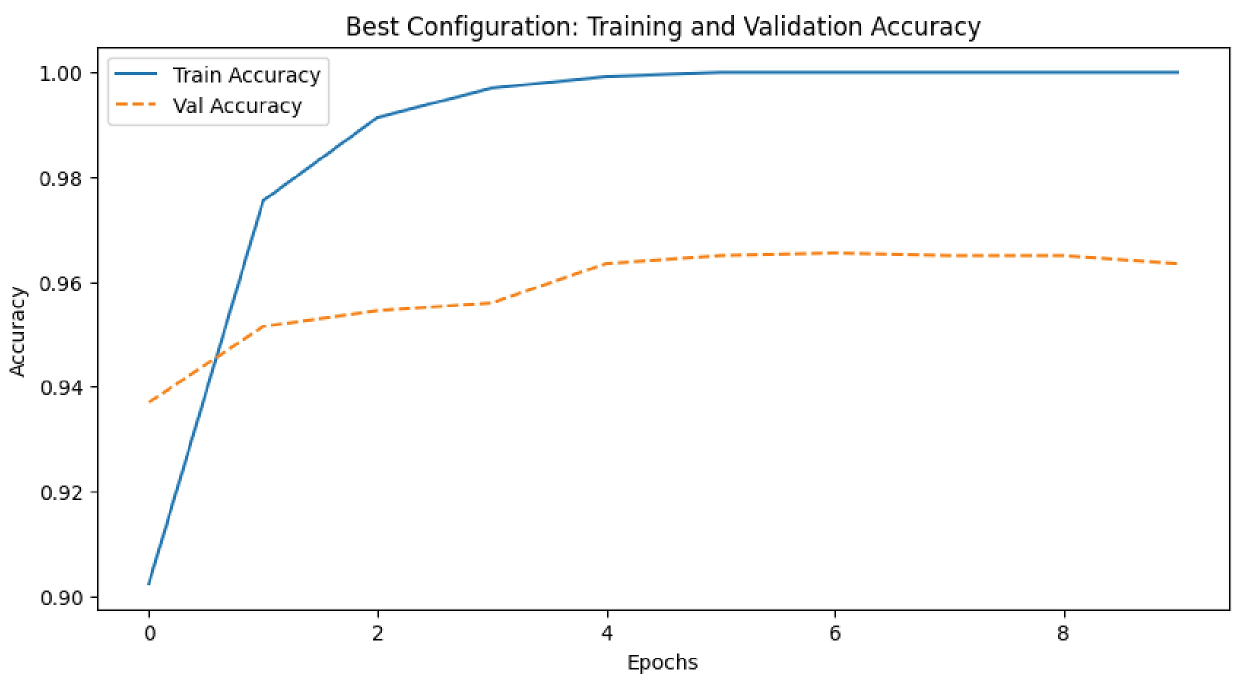

Fig. 4.1.1 Accuracy of Optimizer

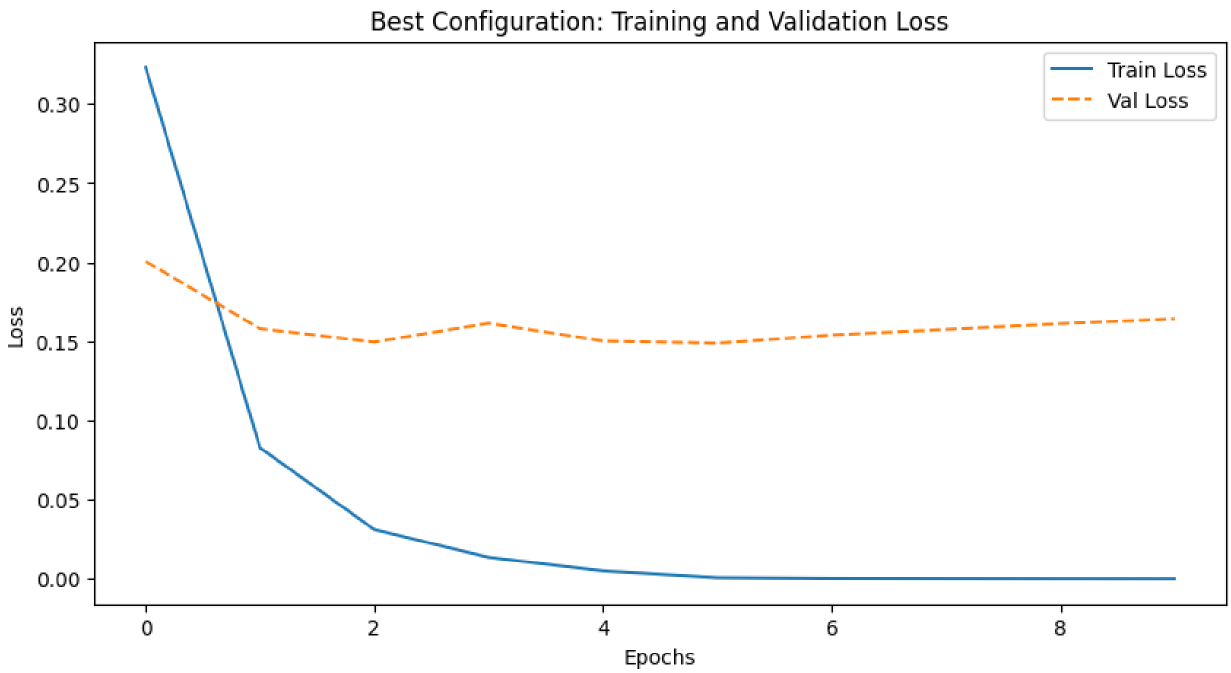

Fig. 4.1.2 Loss of Optimizer

CHAPTER 5.
CONCLUSION AND FUTURE WORK

5.1. Conclusion

The results indicate that:

- Adam emerged as the best optimizer for the MNIST classification task, offering high accuracy and fast convergence.

- RMSprop provided competitive results.

- SGD and Adagrad were suitable for scenarios prioritizing stability over speed.

Adam's adaptive learning rate helps in achieving better performance in most cases, but simpler optimizers like SGD may offer better generalization. The findings highlight the importance of structured hyperparameters tuning and provide a reference for practitioners in similar tasks

5.2. Future work

Future work will explore:

- The use of hybrid optimizers, complex dataset to strengthen the model
- Implementation of different hyperparameter tuning process such as, Random Forest search optimization, Bayesian optimization, or genetic algorithms.
- Apply the optimizers in larger-scale DL models for more complex tasks.

REFERENCES

1. Bielza C, Larrañaga P. Title: Estimation of Distribution Algorithms for Generative Adversarial and Con-volutional Neural Network Hyper-Parameter Optimization. 2024.
2. Kingma DP, Lei Ba J. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.
3. Hazan E, Klivans A, Yuan Y. Hyperparameter Optimization: A Spectral Approach. 2017 Jun 2; Available from: http://arxiv.org/abs/1706.00764
4. Facultad de Ciencias.
5. Negrin I, Roose D, Chagoyén E, Leuven KU. PARAMETER TUNING STRATEGIES FOR METAHEURISTIC METHODS APPLIED TO DISCRETE OPTIMIZATION OF STRUCTURAL DESIGN.
6. Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. Vol. 13, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. John Wiley and Sons Inc; 2023.
7. Andonie R. Hyperparameter optimization in learning systems. Journal of Membrane Computing. 2019 Dec 1;1(4):279–91.
8. Yang L, Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing. 2020 Nov 20;415:295–316.
9. Por R, Luque Á, Tutorizado L, Rubio EL, García González J. Hyperparameter optimization for the radiance field reconstruction model DirectVoxGo.
10. Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for Hyper-Parameter Optimization.
11. Wilson AC, Roelofs R, Stern M, Srebro N, Recht B. The Marginal Value of Adaptive Gradient Methods in Machine Learning.

APPENDIX

1. **Plagiarism Report**

---

report paper.docx

ORIGINALITY REPORT

3%	3%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijfmr.com Internet Source	3%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		