HYPERPARAMETER OPTIMIZATION USING VARIOUS OPTIMIZERS

|  |  |  |
| --- | --- | --- |
| ***Keshav Kumar1***  *Department of Computer Science and Technology*  *Galgotias University*  *Greater Noida, UP*  [*Keshav.kr2825@gmail.com*](mailto:Keshav.kr2825@gmail.com) | ***Jayant Sikarwar2***  *Department of Computer Science and Technology*  *Galgotias University*  *Greater Noida, UP*  [*jayantsikarwar205.js@gmail.com*](mailto:jayantsikarwar205.js@gmail.com) | ***Sugan Patel3***  *Department of Computer Science and Technology*  *Galgotias University*  *Greater Noida, UP*  *techsugan@gmail.com* |

***ABSTRACT*— Technology-dependent models reach their maximum performance by using hyperparameter optimization for users to adjust model parameters. This research investigates performance costs between Adam, RMSprop, SGD and Adagrad as common optimization techniques during their utilization in MNIST handwritten digits classification with a Convolutional Neural Network (CNN). The study employed grid search for analyzing parameters that consisted of learning rate and batch size with dense layers number and neuron layer count.**

**The experimental results indicate that each optimizer produces unique features which determine training performance as well as accreditation achievement along with minimum loss benefits. The outcome showed RMSprop and Adam achieving the quickest convergence time together with achieving superior accuracy results compared to other optimizer implementations. (egtion of training length through SGD and Adagrad demonstrated important impacts on their performance until the optimizers obtained equal results in given system setups. Researchers use this analysis method to determine which optimizers optimized through specific parameters perform best for classification problems with similar structures.**

**Machine learning optimization research benefits from the research method which demonstrates advanced hyperparameter tuning protocols to generate superior results.**

***Keywords*—**

1. INTRODUCTION

The operations of numerous application domains make use of ML models through recommendation systems and computer vision and natural language processing and user behavior analytics. [1] Every application type requires distinct ML algorithm selection.

Developing robust machine-learning models requires an elongated duration since users need to select their algorithms correctly and optimize their hyperparameters until they achieve the best model structure. [2] ML models have parameter components which serve two purposes since initializations and updates from data affect specific parameter types but other types lack direct data-based estimation.

Multiple research teams seek to create better techniques for optimizing hyper-parameter since current procedures have no proven method yet.

I.1 Identification of Relevant Contemporary Issue

The fast-growing quantity of information and continuously developing complex machine-learning systems require optimal model performance for different applications. [4]Data scientists and machine learning engineers now face the modern challenge of selecting the appropriate optimizer among other hyperparameters through the process of hyperparameter optimization. Model convergence speed together with final accuracy and generalization depend strongly on optimizers which stand as vital components for machine learning project development. [5]

I.2. Identification of Problem

The selection process of hyperparameters together with the optimizer choice for machine learning models remains complex to advance. The improper choice of hyperparameters results in training slowness together with overfitting issues or poor generalization. The current industry practice of performing manual optimizer selection together with trial and error methods proves inefficient and time-consuming. [6]The proposed work conducts hyperparameter optimization by examining numerous optimizers to determine which ones produce the best outcomes during implementation of Grid search algorithm.

I.3. Identification of Tasks

* The study examined different optimizers in machine learning through research and review activities.
* Selection of appropriate datasets for experimentation.
* A performance metrics evaluation was conducted on various optimizers for the purpose of comparison.
* The study presents final findings along with recommendations which stem from research results.

Timeline of the Reported Problem

* Since the first machine learning advances researchers studied hyperparameter optimization then its relevance grew substantially because of deep learning development. Since the 1950s SGD optimizers have been in use yet Adam (2014) and subsequent adaptive versions expanded optimization capabilities although adding new complexities. Machine learning models now require more efficient hyperparameter tuning because they present complex and resource-heavy capabilities.[8]
* Technical experts can choose among different hyperparameter optimization approaches which consist of manual tuning alongside grid search and random search. The popularity of modern automated hyperparameter tuning systems includes Bayesian optimization together with evolutionary algorithms. The selection of proper optimizers stands as a vital process because the performance of models depends entirely on which optimizers are chosen. [9]

I.4 Existing Solutions

Numerous studies have explored the benefits and limitations of different optimizers:

* **SGD**: Despite its simplicity, it is highly sensitive to the learning rate and tends to converge slowly, especially in complex tasks.[9]
* **Momentum and Nesterov Accelerated Gradient (NAG)**: Enhancements to SGD, these optimizers add momentum to accelerate convergence, particularly in scenarios where SGD struggles with local minima.
* **Adam**: Known for its adaptive learning rate and momentum, Adam is often the optimizer of choice for most practitioners. However, recent research has raised concerns about its generalization ability compared to SGD.[6]
* **RMSProp and AdaGrad**: These optimizers also adapt the learning rate, making them more suitable for sparse data problems.

I.5. Problem in Existing Solutions

Numerous tasks and datasets yield continuous variations between Adam and RMSProp controllers while these optimizers remain highly accepted throughout the community. The research analyzes performance assessments of optimizers between different tasks since experimental comparison studies remain inadequate in current publications.

1. METHODOLOGY

II.1. Data Collection

We used two datasets in this study:

* MNIST: A widely used image classification dataset consisting of 70,000 28x28 grayscale images of handwritten digits [0-9] with corresponding labels.

II.2. Model Architectures

For each dataset, we implemented specific models:

* MNIST: A simple Convolutional Neural Network (CNN) was used, comprising two convolutional layers followed by two fully connected layers.
* Boston Housing: A Linear Regression model and a Neural Network with three hidden layers were implemented.

II.3. Optimization Technique used

* Grid Search Optimization Technique stands as a common method to evaluate hyperparameter configuration spaces according to [13]. The method assesses each combination of hyperparameters in a grid structure during an exhaustive evaluation process which is also known as a brute-force method. The GS methodology produces its results by evaluating the Cartesian product of every value provided by users according to their specifications [14]. The search capabilities of GS do not automatically focus on the well-performing area of hyperparameter space. The identification of global optima follows a specific manual procedure as described in [13].
* Launch from an extensive research domain and large-hyperparameter value ranges to start
* The search should begin with basic findings of well-receiving parameter sets followed by a narrowing down of intervals and space.
* The process should be repeated through step 2 multiple times to spot the optimal solution in the global domain.
* The main drawback of grid search exists in its poor functionality when operating in large hyperparameter dimensional spaces. The evaluation count increases with exponential speed as more hyperparameters enter the model. The exponential growth problem known as the curse of dimensionality produces computational complexity at O(nk)O(n^k)O(nk) due to kkk hyperparameters and nnn possible parameter values [16]. GS works well as an HPO approach however its effectiveness decreases when the hyperparameter space has numerous parameters.
* The scikit-learn library includes GridSearchCV as a tool for simple grid search implementation which identifies efficient optimal hyperparameters [16]. The prevalence of GS for a number of reasons exists even though researchers have studied advanced global optimization methods and alternative HPO techniques over many years.
* It is easy to implement and automate.
* It consistently identifies better hyperparameters (λ\lambdaλ) compared to manual sequential optimization.
* It is reliable and effective in low-dimensional hyperparameter spaces.

II.4. Optimizers Evaluated

The following optimizers were evaluated:

* SGD: SGD with a different learning rate.
* Adam: Known for its adaptive learning rate and momentum.
* RMSProp: Designed to handle noisy data by adjusting the learning rate based on recent gradients.
* AdaGrad: Adapts learning rates individually for each parameter.

1. Experimental Setup

All experiments were conducted using the TensorFlow framework. Each model was trained for 10 epochs, and the following hyperparameters were tuned:

* Learning rate: Various learning rates [0.001, 0.01] were tested for each optimizer to ensure fair comparison.
* Batch size: Batch sizes of 32, 64 were used across all models.
* Dense Layers: Various Numbers of layers, which were 1, 2 were used across all models
* Neurons per Layer: The number of perceptron or neurons used in each layer were 64, 128, 256, ensuring that each number of neurons are used for all possible combinations, which were 96 in total.

III.1. Evaluation Metrics

We used the following metrics to evaluate the performance of the optimizers:

* Accuracy for classification tasks (MNIST).
* Convergence speed, measured by the number of epochs required to reach a certain accuracy or loss threshold.
* Generalization: Tested by evaluating the model on a validation set and recording any overfitting tendencies.

1. RESULTS AND ANALYSIS

IV.1. Classification Task: MNIST

For the **MNIST dataset**, Adam and RMSprop showed the fastest convergence, but among the two the former came first, reaching almost 98% accuracy with combining it with rest hyperparameters options decided, learning rate 0.001, batch size 32, dense layers 1, Neurons/Layer 256, Training Time 389.25s. **SGD** and **Adagrad** exhibited more fluctuations.

* **RMSProp** performed similarly to Adam in terms of convergence speed but was slightly more stable during training.
* **AdaGrad**, while effective initially, exhibited diminishing returns due to its continuously shrinking learning rate.

**A graph with a line

Description automatically generated**

**Fig. 4.1.1. Accuracy of Adam**

**A graph with a line

Description automatically generated**

**Fig. 4.1.2. Loss by Adam**

The total run time to check all the possible combinations, which were 96, was 4 hrs 52 min 29 sec.

This test was performed on cloud platform, mainly Google Colab, and on IDE of the user’s PC, on a 64 bit system with 2.9 GHz AMD Ryzen 7 4800H processor, 16GB of memory and 4GB of VRAM.

# CONCLUSION AND FUTURE WORK

## V.1. CONCLUSION

## The results indicates 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.
* Regression dataset, like Boston housing dataset, should be tested to increase the performance of the model.

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. 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.
4. Negrin I, Roose D, Chagoyén E, Leuven KU. PARAMETER TUNING STRATEGIES FOR METAHEURISTIC METHODS APPLIED TO DISCRETE OPTIMIZATION OF STRUCTURAL DESIGN.
5. Wilson AC, Roelofs R, Stern M, Srebro N, Recht B. The Marginal Value of Adaptive Gradient Methods in Machine Learning.
6. Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for Hyper-Parameter Optimization.
7. Por R, Luque Á, Tutorizado L, Rubio EL, García González J. Hyperparameter optimization for the radiance field reconstruction model DirectVoxGo.
8. Hazan E, Klivans A, Yuan Y. Hyperparameter Optimization: A Spectral Approach. 2017 Jun 2.
9. Yang L, Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing. 2020 Nov 20.
10. Andonie R. Hyperparameter optimization in learning systems. Journal of Membrane Computing. 2019 Dec 1.
11. Zoller M-A, Huber MF. Benchmark and survey of automated machine learning frameworks, arXiv preprint arXiv: 1904.12054, 2019.
12. Hutter F, Kottho L, Vanschoren J.Automated machine learning: methods,systems, challenges. Springer Nat. 2019;1:219.
13. Claesen M, Simm J, Popovic D, et al. Easy hyperparameter search using optunity, arXiv preprint arXiv: 1412.1114, 2014.
14. Daniel Mesafint Belete & Manjaiah D. Huchaiah (2021): Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results.