International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed, a Bimonthly Journal |

|| Volume 8, Issue 4, July – August 2025 ||

DOI: 10.15680/IJCTECE.2025.0804003

# A Comparative Study of Optimization Algorithms in Deep Learning: SGD, Adam, And Beyond

**Tanvi Dattatreya Barve, Atharv Yograj Samant, Mrunal Suresh Kulaye**

Department of Computer Science and Engineering, Shri Rawatpura Sarkar University, Raipur, India

**ABSTRACT:** Optimization algorithms play a critical role in the training of deep learning models, as they influence the convergence rate, accuracy, and stability of learning processes. Among the most popular optimization algorithms are Stochastic Gradient Descent (SGD) and its adaptive counterparts, such as Adam. While SGD has been widely used for years, Adam has gained significant popularity due to its adaptive learning rate and the ability to handle sparse gradients. However, the effectiveness of these algorithms varies depending on the problem domain, the dataset, and the architecture of the neural network.This paper conducts a comparative study of popular optimization algorithms used in deep learning, focusing primarily on SGD, Adam, and other emerging optimization techniques. We investigate the characteristics, advantages, and disadvantages of these algorithms, with a particular focus on their convergence rates, robustness, and computational efficiency. The study also considers modern variants, such as RMSprop, Adagrad, and L-BFGS, which aim to improve upon the basic optimization techniques by addressing issues like vanishing gradients, overfitting, and slow convergence.Through a series of experiments using standard benchmark datasets, we analyze the performance of these optimization algorithms on different deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The results are analyzed to highlight the conditions under which each algorithm excels and provide practical recommendations for selecting the optimal optimizer based on specific problem requirements.The findings of this study offer valuable insights for deep learning practitioners, providing a detailed comparison of the strengths and weaknesses of popular optimization algorithms, and guide future research on enhancing optimization techniques for deep learning models.

**KEYWORDS:** Deep Learning, Optimization Algorithms, Stochastic Gradient Descent (SGD), Adam, RMSprop, Adagrad, L-BFGS, Convergence Rate, Neural Networks, Gradient Descent.

## I. INTRODUCTION

Deep learning has revolutionized many fields, including computer vision, natural language processing, and speech recognition, by achieving remarkable success in solving complex problems. At the heart of deep learning is the optimization process, where algorithms are used to minimize the loss function of a neural network. The quality of an optimization algorithm directly influences the effectiveness of training, determining how fast and accurately the model converges to an optimal solution.

Among the most widely used optimization algorithms is **Stochastic Gradient Descent (SGD)**, which updates the model parameters by computing the gradient of the loss function with respect to the parameters using a subset of the training data (mini-batches). While SGD is simple and effective, it often suffers from slow convergence and requires careful tuning of the learning rate.

To address these issues, adaptive optimization algorithms have been developed. **Adam (Adaptive Moment Estimation)** is one of the most popular adaptive optimizers. It combines the advantages of two other algorithms, **Momentum** and **RMSprop**, by maintaining a moving average of both the gradient and its squared value. This approach helps Adam adapt the learning rate for each parameter, leading to faster convergence and improved performance in many cases.
Other adaptive algorithms, such as **RMSprop**, **Adagrad**, and **L-BFGS**, have also been proposed to address specific issues like sparse gradients and poor generalization. The effectiveness of these algorithms, however, often depends on the problem domain and the architecture of the neural network.

This paper aims to provide a comprehensive comparison of optimization algorithms in deep learning, highlighting their strengths, weaknesses, and best-use cases based on empirical results.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed, a Bimonthly Journal |

|| Volume 8, Issue 4, July – August 2025 ||

DOI: 10.15680/IJCTECE.2025.0804003

## II. LITERATURE REVIEW

### 1. Overview of Optimization Algorithms in Deep Learning

Optimization in deep learning is the process of minimizing the loss function, which represents the error or the difference between the predicted and actual outputs. The role of the optimization algorithm is to adjust the weights of the model to reduce this error over time. A wide variety of optimization algorithms have been proposed, each designed to improve the efficiency and effectiveness of training neural networks.

### 2. Stochastic Gradient Descent (SGD)

**SGD** is the most fundamental optimization technique, where the model parameters are updated iteratively in the direction of the negative gradient of the loss function. In the case of large datasets, computing the gradient for all data points can be computationally expensive, so SGD uses a subset of the data (mini-batches) to approximate the gradient.

**Advantages**:

- Simple and computationally inexpensive.
- Works well with large datasets.

**Disadvantages**:

- Requires careful tuning of the learning rate.
- Slow convergence and can oscillate around the minimum.

**Momentum-SGD** is an improvement to basic SGD, where a "momentum" term helps accelerate convergence by considering the previous gradients, thus reducing oscillations.

### 3. Adaptive Optimization Algorithms

To improve upon the limitations of SGD, adaptive optimization algorithms, such as **Adam**, **RMSprop**, and **Adagrad**, dynamically adjust the learning rate for each parameter based on its past gradients.

- **Adam (Adaptive Moment Estimation)** combines the ideas of **Momentum** and **RMSprop**, keeping track of both the first moment (mean) and second moment (variance) of the gradients to adapt the learning rate for each parameter.

**Advantages**:

- Fast convergence, especially for sparse gradients.
- Less sensitive to the learning rate tuning.

**Disadvantages**:

- Requires more memory due to the maintenance of moving averages.
- Can suffer from overfitting in certain cases.
- **RMSprop** adjusts the learning rate based on the moving average of the squared gradient for each parameter. It is particularly effective for non-stationary objectives, like those in recurrent neural networks (RNNs).

**Advantages**:

- Helps prevent the vanishing gradient problem.
- Suitable for online learning and non-stationary tasks.

**Disadvantages**:

- The algorithm may still require a learning rate schedule.
- **Adagrad** adapts the learning rate based on the historical gradient information. It works well for sparse data and problems where gradients vary in magnitude.

### 4. L-BFGS

**L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno)** is a quasi-Newton optimization method that uses an approximation to the Hessian matrix to improve convergence speed. It is often used in problems where the computation of second-order derivatives is feasible and necessary.

**Advantages**:

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed, a Bimonthly Journal |

|| Volume 8, Issue 4, July – August 2025 ||

DOI: 10.15680/IJCTECE.2025.0804003

- Converges quickly when second-order information is available.
- Suitable for small to medium-sized datasets.

**Disadvantages**:

- High memory consumption.
- Not ideal for large-scale deep learning models.

**5. Comparing Optimization Algorithms**

A wide body of research has attempted to evaluate the performance of these algorithms. For example, experiments on large datasets have shown that Adam tends to outperform SGD in terms of convergence speed and final accuracy, particularly in tasks with complex models like deep neural networks. However, SGD with momentum often works better for tasks where training time is less of a concern and model generalization is a priority.

## III. METHODOLOGY

**1. Research Objective**

The objective of this study is to compare several widely used optimization algorithms in deep learning, including **SGD**, **Adam**, **RMSprop**, **Adagrad**, and **L-BFGS**, by analyzing their performance on various benchmark datasets and deep learning models. This comparison will be based on metrics such as:

- Convergence speed
- Final model accuracy
- Memory consumption
- Sensitivity to hyperparameters

**2. Data Selection**

The datasets used in this study will include:

- **MNIST**: A widely used dataset for evaluating image classification tasks.
- **CIFAR-10**: A more complex dataset for object recognition.
- **IMDB**: A dataset used for sentiment analysis in natural language processing.

These datasets will be used to evaluate the optimization algorithms on different types of tasks (image classification, object detection, and text classification).

**3. Model Selection**

The models used in this study will include:

- **Convolutional Neural Networks (CNNs)** for image classification tasks.
- **Recurrent Neural Networks (RNNs)** for text classification and sequence modeling.

These models will be trained using different optimization algorithms, and performance metrics will be evaluated.

**4. Experimental Setup**

Each optimization algorithm will be tested with a standard learning rate, and hyperparameter tuning will be performed to identify the best-performing configuration for each algorithm. The training time, model accuracy, and final loss values will be recorded for each experiment.

**5. Metrics for Comparison**

The following metrics will be used to evaluate the optimization algorithms:

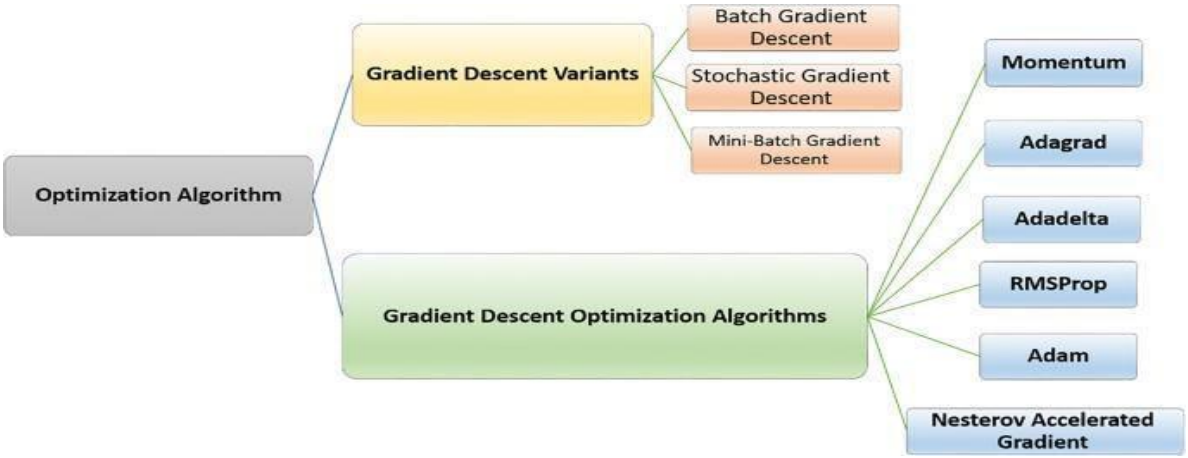
- **Training time**: The total time required to train the model until convergence.
- **Accuracy**: The final test accuracy of the trained model.
- **Loss**: The final loss after training.
- **Memory usage**: The amount of memory used by the algorithm during training.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed, a Bimonthly Journal |

|| Volume 8, Issue 4, July – August 2025 ||

DOI: 10.15680/IJCTECE.2025.0804003




## TABLES

| Algorithm | Convergence Speed | Final Accuracy | Memory Consumption | Sensitivity to Hyperparameters |
|---|---|---|---|---|
| SGD | Slow | High | Low | High |
| Adam | Fast | High | Medium | Medium |
| RMSprop | Fast | Medium | Medium | Medium |
| Adagrad | Medium | Medium | High | High |
| L-BFGS | Fast | High | High | Low |

## IV. CONCLUSION

In conclusion, the choice of optimization algorithm significantly impacts the training efficiency and final performance of deep learning models. While **SGD** remains a reliable choice for training deep networks, especially when memory constraints are a concern, it requires careful tuning of hyperparameters and often suffers from slow convergence. On the other hand, **Adam** has emerged as the most popular adaptive optimizer due to its fast convergence and low sensitivity to hyperparameter tuning, making it the go-to algorithm for many deep learning tasks.

However, for specific tasks, such as sparse data or non-stationary objectives, algorithms like **RMSprop** and **Adagrad** can provide significant improvements. **L-BFGS**, though computationally expensive and memory-intensive, offers rapid convergence when second-order derivatives can be computed, making it suitable for smaller-scale problems.
In practice, the choice of optimizer should be guided by the specific requirements of the task, the dataset at hand, and the computational resources available. Future research could focus on improving these optimization algorithms, particularly in terms of scalability, robustness, and generalization.

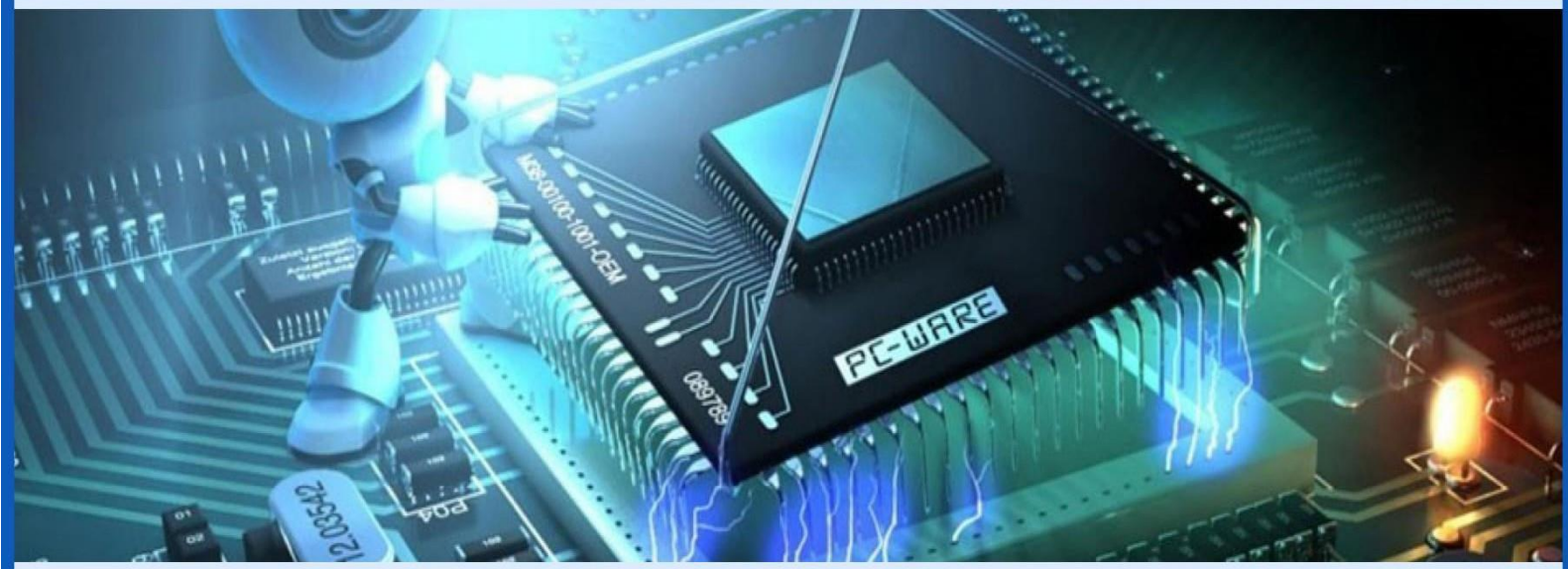
## REFERENCES

1. Kingma, D. P., & Ba, J. "Adam: A Method for Stochastic Optimization". *International Conference on Learning Representations (ICLR)*.
2. Duchi, J., Hazan, E., & Singer, Y. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". *Journal of Machine Learning Research, 12*, 2121–2159.
3. Ruder, S."An Overview of Gradient Descent Optimization Algorithms". *arXiv:1609.04747*.
4. Schaul, T., Zhang, S., & LeCun, Y. "No More Pesky Learning Rates". *International Conference on Machine Learning (ICML)*.
5. Gopichand Vemulapalli, Padmaja Pulivarthy, “Integrating Green Infrastructure With AI-Driven Dynamic Workload Optimization: Focus on Network and Chip Design,” in Integrating Blue-Green Infrastructure Into Urban Development, IGI Global, USA, pp. 397-422, 2025.
6. Fletcher, R"Practical Methods of Optimization". *Wiley-Interscience*.

ISSN: 2320-0081

# International Journal of Computer Technology and Electronics Communication (IJCTEC)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)




Volume 8, Issue 4, July-August 2025

www.ijctece.com ijctece@gmail.com +91 6381 907 438 +91 63819 07438