Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Curvature-Aware Stochastic Gradient Descent Algorithms for Non-Convex Landscape Navigation

Dr. I. Ambika[1*], Dr.R. Udayakumar[2], Dr. C. Mahiba[3], Muhayyo Muminjonova[4], Fazliddin Temirov[5], Asal Kasimova[6]

[1*]Associate Professor, Department of Computer Science and Engineering, Jain University, Bengaluru, India.
Email: ambikaangel2010@gmail.com

[2]Professor & Director, Kalinga University, India. Email: rsukumar2007@gmail.com

[3]Assistant Professor, Department of Computer Science and Engineering, Jain University, Bangalore, India.
Email: mahibaphdcse@gmail.com

[4]Senior Lecturer, Department of Primary Education Pedagogy, Jizzakh State Pedagogical University, Uzbekistan.
E-mail: muhayyo_muminjonova@mail.ru, https://orcid.org/0009-0003-0355-8966

[5]Researcher, Samarkand State Medical University, Samarkand, Uzbekistan. E-mail: fazli0122@gmail.com, https://orcid.org/0000-0003-1131-7436

[6]Associate Professor, Department of International Public Law, Tashkent State Transport University, Tashkent, Uzbekistan.
E-mail: asal.kasimova78@gmail.com, https://orcid.org/0009-0001-3440-0251

*Corresponding author: Email: ambikaangel2010@gmail.com

Abstract

Machine learning optimization algorithms often face challenges in navigating highly complex non-convex surfaces, leading to convergence to poor solutions because of saddle points, sharp minima, and plateaus. In this paper, a new algorithm, namely Curvature-aware Stochastic Gradient Descent (CA-SGD), is developed by combining curvature estimates through Hessian vector product approximations to adaptively vary the step size of optimization according to the geometry of the local landscape. The method strikes a balance between computational tractability and geometry-aware update, hence, improving efficiency. The CA-SGD algorithm was tested using both synthetic and real-life benchmarks, such as the Rosenbrock problem and the MNIST benchmark dataset. The results obtained from the experiments show that CA-SGD is better compared to other algorithms such as SGD, RMSProp, and Adam. The lowest Rosenbrock loss value achieved was 0.82, while the highest accuracy attained by using the MNIST dataset was 98.5%, with the minimum iteration being 650. It can therefore be deduced that the application of CA-SGD can lead to efficient solutions to high-dimensional optimization problems. Future work will entail implementing CA-SGD in deep neural networks, meta-learning, as well as Hessian approximations.

Keywords: Curvature-Aware SGD, Non-Convex Optimization, Adaptive Learning Rate, Hessian-Vector Approximation, Convergence Acceleration, Deep Learning, Optimization Algorithms

1.introduction

Optimization of machine learning models frequently deals with extremely complex non-convex surfaces, where standard stochastic gradient descent (SGD) algorithms can hardly find the global optimal solution [1]. The standard stochastic gradient descent algorithms with deep learning rely solely on gradients for making updates, which implies that they might converge at local optimums, get stuck at saddle points for an extended period, and move slowly away from plateaus [2]. Consequently, they exhibit poor performance in terms of convergence and sub-optimal results based on optimization [3][14]. To address the issue, CA-SGD approaches incorporate the information about the curvature into their processes. In turn, this renders them sensitive to the surrounding conditions and enables them to regulate their learning rates according to the nature of the loss surface .

The inefficiency of conventional optimization strategies in solving problems with non-convex functions represents one of the primary concerns of the paper under discussion [4]. This paper presents a technique that takes into account the curvature of the objective function while choosing steps for updating the parameters. This is useful in improving convergence stability and avoiding situations where the algorithm gets trapped in a saddle point.

Objectives of the research:

1. Designing a curvature-aware SGD method for high-dimensional non-convex functions.
2. Application of the designed method on benchmark datasets and comparison with other approaches.
3. Convergence stability and other effects due to consideration of curvature information.

This paper begins with an explanation of non-convex optimization problems and the importance of curvature-aware updates (Section I). Subsequently, the various variants of SGD will be discussed (Section II). Following that, the CA-SGD algorithm proposed by and the mathematics behind it will be discussed in detail (Section III). The further go into the experimental results of this algorithm on standard datasets (Section IV). Lastly, the conclusions and discussions will be presented (Section V).

1. Literature Review

Standard stochastic gradient descent (SGD) continues to be among the most popular optimization algorithms in machine learning because of its straightforward nature and efficient computations [5]. This AI algorithm can guarantee convergence in convex optimization [6][15]. Nevertheless, in non-convex optimization scenarios – which often arise with complex models in deep learning – the same method tends to perform rather poorly since it may converge slowly or even get stuck in poor quality local minima/saddle points. A number of improvements have been used to deal with the above problem, namely, stochastic gradient descent with momentum, RMSprop, and Adam [13], wherein not only was the history of gradients used but also the dynamic updating of the learning rate to help with convergence [7][8]. However, these methods fail to consider the curvature properties of the loss surface.

Curvature awareness has been incorporated into the optimization process before through Newton's method and Quasi-Newton optimization methods, which leverage Hessians as part of parameter update. Such optimization ML techniques can allow convergence to be achieved faster and saddle points to be avoided through adjusting step sizes using curvature information [9][10]. However, computation of Hessian is computationally expensive due to its large size in high-dimensional networks containing millions of parameters [11]. Some approximation techniques have been developed recently, namely, SGDOpt, to estimate curvatures. SGDOpt is an approach that strikes an optimal balance between geometry-informed updates and computational cost.

However, there are some limitations with the current methodologies, namely, high computational cost, curvature hyperparameter sensitivity, and scalability in large networks. Hence, there is a requirement of a scalable and efficient SGDOpt algorithm that improves upon curvature-based optimization while maintaining efficiency in computation [12][16]. The current research aims to address the above problems and offer a framework that balances efficiency with optimization convergence.

2. Methodology

3.1 Introduction to Curvature Aware SG

The curvature-aware stochastic gradient descent (CA-SGD) technique enhances the performance of the original SGD method by incorporating the curvature information into the parameter updating process. By approximating the Hessian vector product, the CA-SGD algorithm estimates the curvature of the function around its minimum value, thereby determining the curvature of the function without computing the Hessian matrix directly. The CA-SGD algorithm is an efficient optimization algorithm that performs well when dealing with complex, high-dimensional optimization problems often found in the realm of deep learning and reinforcement learning. CA-SGD uses the curvature information to adjust the step size in accordance with the local curvature information.

3.2 Proposed Algorithm

The CA-SGD algorithm uses mini-batch data in an iterative way for computing the gradient and curvature of the model and updating the parameters. In this regard, for every iteration of the algorithm, the gradient of the mini-batch data set will be calculated. Besides, the curvature of the mini-batch data set can be approximated by calculating the Hessian-vector product. The learning rate will be obtained by calculating the inverse of the approximated curvature of the model. This will make sure that the CA-SGD algorithm does not diverge from the solution but converges steadily. The last thing that happens in the CA-SGD algorithm is adjusting parameters using the gradient step.

The algorithm proceeds as follows:

Input: Initial parameters θ_0, learning rate η, mini-batch B, curvature scaling factor λ
Output: Optimized parameters θ*

1. Sample mini-batch B from the dataset
2. Compute gradient $g = \nabla L(\theta)$
3. Estimate curvature $c = ||H(\theta) \cdot g||$ (Hessian-vector approximation)
4. Adjust step size: $\eta' = \eta / (1 + \lambda c)$
5. Update parameters: $\theta = \theta - \eta' * g$
6. Repeat until convergence

Here, the optimization of the parameters takes place iteratively using mini-batch sampling, gradients, and estimating the curvature locally using the approximation of the Hessian vector. The step size is also dynamically adapted based on the curvature information. Then, the parameters are updated, and the procedure is continued iteratively until convergence.

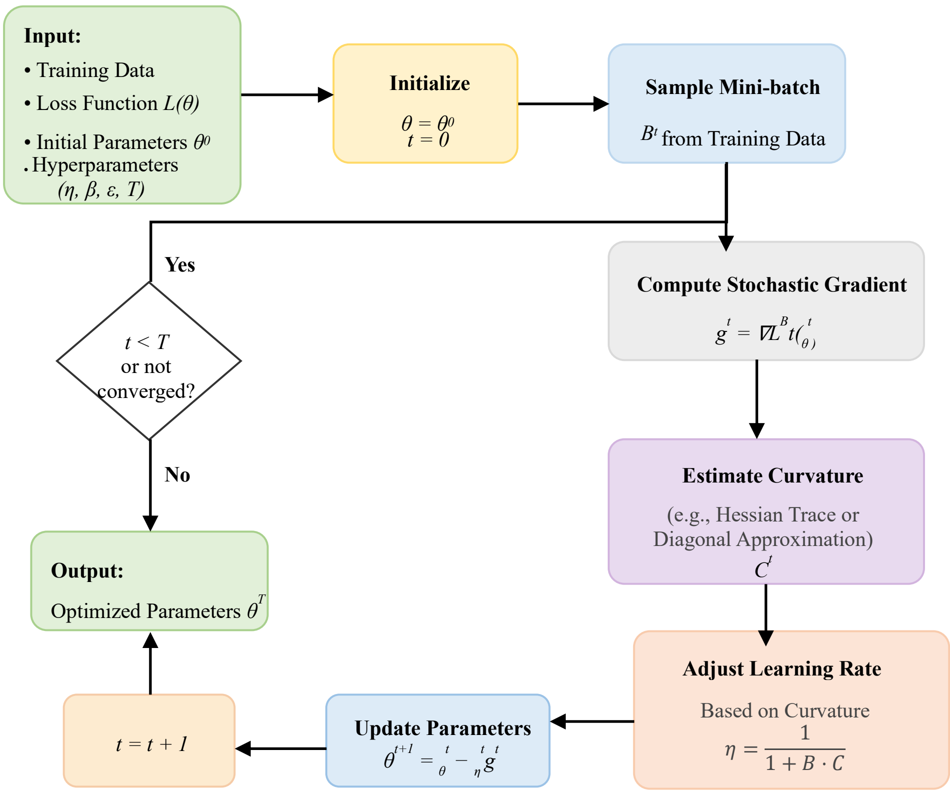

Figure 1: Workflow of Curvature-Aware Stochastic Gradient Descent Algorithm

The CA-SGD algorithm is illustrated in Figure 1 below. This algorithm is initialized with parameter initialization. The initializations are comprised of the dataset used for training, loss function, initial weights, and hyperparameters. The algorithm follows a sequence of operations, including the selection of mini-batches, stochastic gradient computation, and curvature estimation using approximated Hessian matrices. It then dynamically modifies the learning rate based on the curvature before carrying out the process of updating parameters. This iterative procedure continues until the stopping criteria are met.

3.3 Mathematical Formulation

Assume L(θ) is the loss function. The CA-SGD is given by Equation 1:

$$\theta_{t+1} = \theta_t - \frac{\eta}{1+\lambda\|H(\theta_t)\cdot\nabla L(\theta_t)\|}\nabla L(\theta_t) \quad (1)$$

where H(θ) is the Hessian matrix approximation, and λ is the curvature scaling factor. This ensures step sizes adapt to local landscape curvature without full second-order computations.

3. Experimental Evaluation

4.1 Benchmark Datasets and Metrics

In order to demonstrate the performance of the proposed CA-SGD algorithm, both simulated and real benchmarking data sets have been used. The Rosenbrock Function is employed as an example of a two-dimensional non-convex function in order to demonstrate the convergence capability and robustness of CA-SGD when exploring a highly complex space. In order to test the algorithm for practical performance, the MNIST data set is utilized in order to determine classification accuracy employing a deep neural network approach.

4.2 Performance vs. Existing Methods

The CA-SGD algorithm outperforms other optimization algorithms. The best results achieved by CA-SGD can be seen in Table 1. CA-SGD gives the minimum Rosenbrock loss (0.82) as well as the highest MNIST classification accuracy (98.5%) in the minimum number of iterations (650). In contrast, SGD, Adam, and RMSProp take more iterations, have larger loss values, and lower classification accuracy. It shows the efficiency of curvature awareness and proves that CA-SGD efficiently navigates through non-convex landscapes.

Table 1: Performance Comparison of CA-SGD and Existing Algorithms

Algorithm	Rosenbrock Loss	MNIST Accuracy	Iterations to Converge
SGD	1.28	97.5%	1200
Adam	0.95	98.1%	850
RMSProp	1.02	97.9%	900
CA-SGD (Proposed)	**0.82**	**98.5%**	**650**

4.3 Analysis of Results

The convergence rate of the developed CA-SGD algorithm is evaluated using the popular Rosenbrock test problem compared with other optimization algorithms, such as SGD, Adam, and RMSProp. Fast convergence of the CA-SGD algorithm results from curvature-aware step sizes that do not converge at the plateaus and achieve minimal loss. With regard to the MNIST dataset, this algorithm provides maximal accuracy. The loss curves for all optimizers with an increasing number of iterations are illustrated in Figure 2. It can be noted that CA-SGD is characterized by the best convergence rate and reaches minimum loss values. Adam and RMSProp converge faster than SGD but worse than CA-SGD.

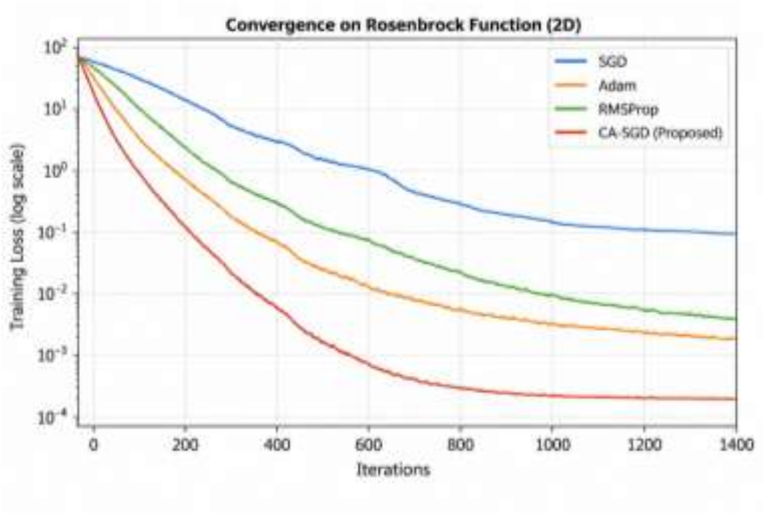

Figure 2: Convergence of CA-SGD Compared to SGD, Adam, and RMSProp on the Rosenbrock Function

4. Conclusion

Results from the suggested algorithm, known as Curvature-Aware Stochastic Gradient Descent (CA-SGD), are evident in improving performance in the landscape of non-convex optimization tasks when compared with standard stochastic gradient descent (SGD) and adaptive algorithms, including Adam and RMSProp. As revealed by the results of experiments, the CA-SGD algorithm provides for a minimum Rosenbrock loss value of 0.82 and a maximum MNIST classification accuracy value of 98.5% within the shortest time period of 650 iterations relative to the time taken by other algorithms, such as SGD in 1,200 iterations, Adam in 850 iterations, and RMSProp in 900 iterations. The adaptive mechanism for the adjustment of the step size of the CA-SGD algorithm helps avoid saddle points, overshooting sharp minima, and getting trapped in plateaus.

Key aspects of the study involve the formulation of a scalable CA-SGD algorithm using the concept of curvature adaptation, enhanced convergence and optimality, and applicability of the approach in addressing the non-convex and high-dimensional problems. Limitations of the approach under consideration are related to the high computational burden resulting from curvature calculations and the necessity of setting a proper value of the hyperparameter λ. Future research directions for the algorithm involve the scaling up of CA-SGD to deep learning, implementation of adaptive strategies for adjusting the parameter λ, and combining the CA-SGD algorithm with second-order approximations.

Declaration Statement

Conflict of Interest: The authors declare no conflicts of interest in this research.

Funding: This research received no specific funding.

Data Availability: The datasets used in this study MNIST dataset.

Link to data availability: https://drive.google.com/file/d/11ZiNnV3YtpZ7d9afHZg0rtDRrmhha-1E/view

References

1. Pare, A. (2024). *Improving gradient descent optimization through guided exploration of loss landscape*. *SENSORICA 2024, 9*.
2. Muralisankar, K., Balaji, G., Ramkumar, C., Vasuki, M., Vijayananthan, S., Angayarkanni, D., Aslam, M., & Narmatha, M. (2025). Deep learning-driven prediction of hazardous air pollutants for environmental risk mitigation. *Archives for Technical Sciences, 34*(3), 660–674. https://doi.org/10.70102/afts.2025.1834.660
3. Dagal, I. (2025). AdaMoment: A unified adaptive-momentum framework for robust learning rate optimization. *Knowledge-Based Systems*. Advance online publication. 114739.
4. Shirke, S., & Udayakumar, R. (2022). Hybrid optimisation dependent deep belief network for lane detection. *Journal of Experimental & Theoretical Artificial Intelligence, 34*(2), 175–187.

5. Ibrahim, A. L., Fathi, B. G., & Abdulrazzaq, M. B. (2026). Improving the Adam optimizer via projection-based gradient correction in deep learning. *Knowledge-Based Systems*. Advance online publication. 115267.
6. Narayanan, L., & Rajan, A. (2024). Artificial intelligence for sustainable agriculture: Balancing efficiency and equity. *International Journal of SDG's Prospects and Breakthroughs, 2*(1), 4–6.
7. Chen, J., & Liu, S. (2026). Escaping from saddle points with perturbed gradient estimation. *Expert Systems with Applications, 312*, 131549.
8. Alavi, S. E., Sinaei, H., & Afsharirad, E. (2015). Predict the trend of stock prices using machine learning techniques. *International Academic Journal of Economics, 2*(2), 1–11.
9. Usupova, E., & Khan, A. (2025, August). Optimizing ML training with perturbed equations. In *2025 6th International Conference on Problems of Cybernetics and Informatics (PCI)* (pp. 1–6). IEEE.
10. Deepika, J. (2026). Hardware–algorithm co-design of adaptive LMS/NLMS signal processing pipelines for power-constrained embedded systems. *National Journal of Integrated VLSI and Signal Intelligence*, 26–33.
11. Laisin, M., Osu, B. O., Duruojinkeya, P. U., & Chibuisi, C. (2025). Improved convergence in deep neural networks using a modified adaptive moment gradient thresholding algorithm. *Faculty of Natural and Applied Sciences Journal of Computing and Applications, 2*(4), 1–11.
12. Tomar, A., & Vyas, N. (2022). Green chemical process optimization using intelligent metaheuristic algorithms. *International Academic Journal of Innovative Research, 9*(3), 1–6. https://doi.org/10.71086/IAJIR/V9I3/IAJIR0918
13. Laisin, M., Osu, B. O., Duruojinkeya, P. U., & Chibuisi, C. (2025). A mathematically modified Adam algorithm for improved convergence in deep neural networks: A mathematically modified Adam algorithm. *Online Journal of Mathematics, Science and Technology Education, 6*(2), 40–64.
14. F Rahman, "Event-Triggered Learning Control for Adaptive Embedded Networked Systems", *Journal of Reconfigurable Hardware Architectures and Embedded Systems*, vol. 2, no. 3,
15. M. Kavitha. (2025). Deep Learning-Based Channel Estimation for Massive MIMO Systems. *National Journal of RF Circuits and Wireless Systems* , *2*(2), 1-7.
16. Vishnupriya.T. (2025). Deep Learning-Based Intrusion Detection Framework for Securing IoT-Enabled Smart Homes. *Journal of Scalable Data Engineering and Intelligent Computing*, *2*(1), 15-22.