A Comprehensive Analysis of Optimization Algorithms for Large-Scale Machine Learning

Aman Bhitkar[1], Sujal Bohara[2], Tanishk kulkarni[3], Prof. Dipti Pandit[4]
[1,2,3,4]Department of Electronics and Telecommunication Engineering,
Vishwakarma Institute of Information Technology, Pune, India
Email: [1]aman.22210836@viit.ac.in , [2]sujal.22211555@viit.ac.in , [3]tanishk.22210337@viit.ac.in , [4]dipti.pandit@viit.ac.in .

***Abstract*—Optimization algorithms determine the best model parameters that tend to offer prediction accuracy in complex machine learning (ML) problems in big data. In fact, they are very crucial while handling large-scale data because old ways of handling datasets will be computationally very inefficient. This paper provides an in-depth comparison of the two most widely used optimization techniques in machine learning: SGD, and BGD. Our simulated results indicate that though the SGD technique indicates very fast initial convergence, its efficiency eventually tends to degrade with the increasing iteration of the algorithm. On the other hand, the BGD approach might take slow initiation but is relatively consistent in a long run. We then further probe how the variations in learning rate affect its performance in the case of both the methods. Our analysis shows that adaptive learning rates drastically accelerate convergence. Finally, we show that the computational efficiency of the SGD method makes it a better choice since gradients can be computed on a per sample basis, which makes the method better for scaling.**

***Keywords*—Optimization Algorithms, Stochastic gradient descent, Batch gradient descent, Machine learning, Large-scale data, and Adaptive learning rate.**

• INTRODUCTION

The fast-paced scenario of embedded systems demands the utmost energy efficiency, especially for devices running on resource-constrained environments e.g., Internet-of-Things (IoT) and wireless sensor networks. In this paper power optimizing techniques in embedded systems are explored, especially for real-time operating system (RTOS) and their effective use to the low-power designs. Strategies from the literature such as Dynamic Voltage and Frequency Scaling (DVFS), Dynamic Power Management (DPM), sleep modes, and task scheduling algorithms are described in order to preserve energy consumption while providing real-time performance support. The paper aims, through this review, to provide insights into current energy optimization techniques and their relevance to the design of energy-efficient RTOS for embedded systems.

I. BACKGROUND AND MOTIVATION

This has been done so that others may know which of these algorithms is preferable considering the strengths and weaknesses of its capabilities with large-scale tasks of machine learning. Optimization methods like development of machine learning algorithms are traditionally designed to minimize a loss function that measures the amount of disagreement between predictions and actual values. These methods play an important role in evaluating the ability of the model to generalize well to unseen data. However, as large-scale datasets have become prominent, time complexity associated with the traditional algorithms turned out to be a serious limitation. They require optimization methodologies so that good results are achieved while using sufficient amount of time. LMLs have surfaced to overcome such problems: large- scale algorithms more computationally feasible without losing performance.

The paper is a detailed survey of optimization techniques as they evolve into scalable machine learning models where the maximization of computational efficiency will be accompanied by the precision of the model.

II. LITERATURE REVIEW AND RELATED WORK

Some rather good studies have been carried out on ma- chine learning using optimization algorithms. Wang et al.'s systematic survey [3] have emphasized the requirements of the scalability of the machine learning algorithms by study- ing existing methods as well as their relevant computational efficiency along with limitations. On similar lines, Bottou et al. [5] designed optimization methods for large-scale machine learning focusing on stochastic gradient descent and how it could take the advantages of large amounts of data.

This work by Robbins and Monro [4] essentially ushered in stochastic approximation, which also forms the basis of modern stochastic gradient algorithms. More recently, El Hanchi et al. [6] have proposed a method of reweighted gradient to less than the variance in gradients' estimators for efficient time in running times SGD in large scale applications. Zhou et al. [7] also generalized gradient-based methods to multi-task learning and gave improvements about stability of convergence.

This paper extends the material already existing in the literature by providing a more detailed comparison between SGD and BGD concerning convergence speed, computational efficiency, and adaptability to large-scale data.An importance-sampling-based algorithm called SRG (stochastic reweighted gradient) for reduction of the variance of gradient estimator has been proposed in [6]. The authors also proposed an extension of the SRG (SRG+) for the variance reduction

through an important sampling method. This article Analyse the convergence of SRG in the strongly convex and smooth case and conclude that the performance of SRG is better than SGD (Stochastic Gradient Descent). In article [7], authors proposed a novel ML approach based on a multi gradient descent method for finding optimal solutions and a gradient surgery based gradient descent approach for finding stable optimal solutions.

In article [8], authors proposed a new stochastic regu- larized damped Broyden–Fletcher–Goldfarb–Shanno method. This method contains a new gradient difference scheme and a novel damped parameter for solving a non-convex optimization problem. A comparative analysis of stochastic (SG) and batch methods for optimization of objective function has been presented in article [9]. Authors also Analyzed and Studied visual classification in a large number of classes. Paper Conclude that stochastic training suits our large-scale setting well, and stochastic-based method can work as well as a batch technique at a fraction of their cost. The article [10], analyzes and highlights optimization algorithms from an ML viewpoint. The article [11], presented a computationally efficient gradient based optimization method for the optimization of the stochastic cost function with a large data set and high dimension parameters machine learning problems.

Although much of the related work has been carried out in the literature to optimize the machine learning problem, we restrict our discussion to the most suitable algorithms for the large-scale machine learning problem, i.e. stochastic gradient descent algorithm.

III. OPTIMIZATION METHODS

The numerical computation of the system design param- eter or specification of the designed learning training model is optimization. Optimization is one of the vital pillars of machine learning. Machine learning comput- ing methods design and train a model in the objective function to perform a specific task using given data. Many researchers in various communities have been inspired to design widely applicable new methods with the recent success outcomes of LML optimization methods. The effectiveness of the optimization methods will improve the efficiency and performance of the machine learning training models. The performance of the optimization algorithm is usually measured via loss or pre- diction function. The optimization issues in machine learning appeared through prediction and loss functions. The main aim of these machine learning optimization algorithms is to minimize the loss function or prediction function.

A. *Gradient Descent Method*

An optimisation technique for training machine learning models is gradient descent. A convex function serves as the foundation for the gradient descent algorithm. This approach iteratively updates a model's weights in the opposite direction of the loss/cost function's gradients. The goal of these updates is to progressively converge to the optimal point of the objective function. In order to compute the objective function's step size and the number of iterations required to achieve the point of convergence or fact of minima, the learning rate, or , is utilized. Reducing or minimizing the loss/cost function (h(x;), y) (error of projected and actual output) is the aim of gradient descent. The convex function's(local or global minimum) lowest point on its curve can be found using gradient descent known as the point of convergence.

B. *Batch optimization method*

The Batch or Full Gradient Method (FG) is another name for this comprehensive gradient descent technique. Instead of computing a single sample, this approach computes all the samples in an iteration. Therefore, compared to the previously reported stochastic gradient descent approach, it is more costly and computationally inefficient. Instead of computing a single sample on the complete data set at each iteration, the algorithm in this method computes all samples at once randomly.

C. *Stochastic optimization method*

In other words, stochastic gradient descent (SGD) Com- putes the gradient primarily based on a single randomly selected sample from the dataset at every iteration. This leads to faster initial updates, making it computationally efficient for large-scale datasets. However, SGD can in- troduce noise within the gradient estimates, which may also result in fluctuations across the top-rated solution as opposed to convergence to a unmarried factor. $_{t+1} \leftarrow_t \ _t f_j t\ (_t)$ (9) t T := 1, 2, 3, 4... , index j_t is selected randomly from 1, 2....., n . $_t$ is a positive step size. $f_j t\ (_t)$ corresponding to computation of single sample. Therefore, the stochastic estimates of the gradient are computationally more efficient than a gradient based on the entire training set [5], [14]. The main objective of the stochastic gradient descent method is minimizing a strongly convex function. The objective function F : Rdy R is strongly convex function, there exists a constant $a < 0$ such that function f is called strongly convex with parameter.

IV. RESULTS AND DISCUSSION

This chapter provides a comprehensive analysis of the experimental results obtained in performance comparing Stochastic Gradient Descent (SGD) versus Batch Gradi- ent Descent (BGD) over different learning rate strategies and data scales. Our experiment on these measurements compares computation efficiency, convergence speed, and the overall effectiveness of these algorithms on the large- scale machine learning task. Testing both will provide an evaluation of the influence of fixed and adaptive learning rates on the convergence behavior of each optimization method.

Key Observation:

The BGD learns much slower at the initialization but steadily moves toward the optimum and has less variance, hence reliable for tasks that require high precision. SGD, on the other hand, learns much faster in the early phases but as it is getting

close to the optimal solution, the stability becomes problematic. Variable learning rate improves the performance of both SGD and BGD. For SGD, this damps out the oscillations around the minimum seen before, whereas for BGD, this accelerates convergence in general. It shows that adaptive learning rates are indeed crucial in realising the trade-off between convergence speed and convergence accuracy, especially in all large-scale optimization tasks. he results summarize quite important trade-offs between SGD and BGD:

1) Speed of convergence vs. stability: SGD makes fast, initial progress but can easily be unsta- ble near the optimum without careful learning rate tuning. BGD is slower at first but gives a way to more stable and consistent path to convergence, particularly if the learning rate is adaptive.

2)Computational Efficiency: SGD is far more cost-efficient in terms of the computation per itera- tion cost and thus is preferable for large-scale applications when memory and time constraints are critical. Though BGD is computationally heavier, it might be preferred for those applications where high precision or stability is more important than speed.

3)Learning Rate Effect: Both the algorithms depend heavily on the learning rate. A constant learning rate might work well on some instances of problems but poorly on others, in particular on the case of SGD, and convergence is not even guaranteed. This adaptive learning rate prevents over-shooting in both the techniques and allows for a faster convergence towards the final stages of the latter.

D. Future Scope

Even though this paper has provided some idea about the performance and efficiency of SGD and BGD, the open scope for further research is many:
Hybrid Optimization Algorithms:
Hybrid Optimization Methods In the future, one could try to create hybrid optimization methods that combine the strength of both SGD and BGD. More specifically, such optimization methods could switch dynamically from one to the other according to the size of the dataset, the stage of training, or the complexity of the optimization landscape. For example, starting with mini-batch or stochastic updates in the early iteration of training and moving eventually to full- batch updates as optimization converges toward the minimum might balance speed and accuracy. Advanced Learning Rate Schedules That impact on the convergence of the learning rates is quite important as well, and further work in more advanced learning rate schedules could provide even greater gains in performance. Techniques even like cyclical learning rates or cosine annealing have even shown some promise in being able to speed up convergence without paying a penalty in precision. Looking at the use of reinforcement learning to achieve dynamic learning rate variations during training is likely an additional exciting avenue of study. Optimization for Non-Convex Problems
Non-convex optimization is still one of the biggest chal- lenges, especially in deep learning. Further research into more robust methods to navigate such complex landscapes of non- convex problems might potentially lead to some optimization algorithms being far more efficient. Application and Scalabil- ity in Real Life: Techniques that should be studied for large- scale non-convex problems, including higher order methods like Newton's methods or quasi-Newton methods like BFGS, might conceivably provide new insights into more efficient and precise optimization strategies. Finally, second direction for future work is testing the algorithms on specific large-scale problems in real-world ma- chine learning. Scaling optimization methods to distributed computing environments, such as cloud or edge computing, and analyzing how they perform in federated learning would be very valuable. 5.4 Final Words In conclusion, optimization is indeed an important part of machine learning and is only set to increase as datasets become larger and models begin to get more complex. This paper has put under the light some trade- offs between computational efficiency, convergence speed, and stability that appear in SGD and BGD. Although SGD is an immensely effective approach for large-scale machine learning, in particular when combined with adaptive learning rates, BGD still remains a safe and reliable choice for small-scale, high-precision tasks. Mini-batch methods seem the most promising middle ground, especially when deep learning is concerned. Further advancement in hybrid optimization tech- niques as well as more advanced learning rate scheduling strategies would directly enable the professional machine learning expert to still further optimize the performance and scalability of his models in the future.

E. CONCLUSION

This paper brings in an all-round analysis of two of the most widely used optimisation algorithms applied to huge scale ma- chine learning applications, which are SGD and BGD. Indeed, the increasing size of data and their complexity increasingly require scalable and efficient optimization techniques. This article aims to analyze the strengths and weaknesses and relative effectiveness of SGD and BGD in terms of their computation efficiency, speed, and convergence stability in solving large datasets.

A thorough comparative analysis of the large-scale machine learning optimization setting of SGD and BGD is shown. It is established that although BGD converges stably and accurately, it has computation impracticality with growing data at each step owing to processing the entire dataset at each step. Whereas SGD is computationally efficient for updating model parameters with individual samples or small batches, it has a erratic behavior near the optimum since it is based on some inherent randomness due to its subjectiveness.

Adaptive learning rates proved effective, with time-variant rates, accelerating convergence while inhibiting overshooting past the optimal point. Overall, SGD will be preferred for such large-scale applications where the computational requirement is the most critical one, whereas BGD remains useful for smaller scales and high stability requiring situations. Further work can be on developing hybrid methods combining the efficiency of SGD with stability of BGD; further advanced learning rate techniques can be evaluated to boost convergence on larger complex datasets.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, "Title of paper if known," unpublished. R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
[7] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, arXiv:1312.6114. [Online]. Available: https://arxiv.org/abs/1312.6114
[8] S. Liu, "Wi-Fi Energy Detection Testbed (12MTC)," 2023, GitHub repository. [Online]. Available: https://github.com/liustone99/Wi-Fi-Energy-
[9] Detection-Testbed-12MTC
[10] "Treatment episode data set: discharges (TEDS-D): concatenated, 2006 to 2009." U.S. Department of Health and Human Services, Substance Abuse and Mental Health Services Administration, Office of Applied Studies, August, 2013, DOI:10.3886/ICPSR30122.v2
[11] K. Eves and J. Valasek, "Adaptive control for singularly perturbed systems examples," Code Ocean, Aug. 2023. [Online]. Available: https://codeocean.com/capsule/4989235/tree