Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/2025.TJ24498

Analysis of Optimizing Distributed Machine Learning Techniques for Large-Scale Data Applications

Lin Xu

School of Software, Shandong University, Jinan, China
202200300122@mail.sdu.edu.cn

Abstract. This study investigates the optimization and application of Distributed Machine Learning (DML) techniques, which are essential for addressing the computational challenges posed by large-scale datasets in modern applications such as autonomous driving and natural language processing. The primary objective is to evaluate the performance and characteristics of key DML methods, including MapReduce, Parallelized Stochastic Gradient Descent (PSGD), and Federated Learning (FL). A comprehensive analytical framework is employed to explore the underlying mechanisms of each method. For MapReduce, the study examines data partitioning, mapping, shuffling, and reducing processes to assess its efficiency in managing large datasets. In PSGD, the research analyzes the methods of data allocation, local gradient computation, and parameter aggregation to highlight its parallelization benefits and identify potential bottlenecks. FL focuses on its privacy-preserving model training process, which includes client-server communication, local updates, and global aggregation. Experimental results show that while MapReduce offers fault tolerance, PSGD accelerates training on large datasets, and FL ensures data privacy. However, limitations such as communication overhead and convergence issues remain, underscoring the trade-offs between efficiency, scalability, and privacy in DML applications.

Keywords: Distributed Machine Learning, MapReduce, Parallelized Stochastic Gradient Descent, Federated Learning.

1. Introduction

With the development of Machine Learning (ML), its application areas expand continuously. People's daily lives and work generate massive amounts of data. Hence, the ability to process this data is crucial for applying machine learning.

For instance, autonomous driving and natural language processing applications require substantial training data [1,2]. In centralized machine learning, the existing hardware performance is challenging to meet the increased computational demands of the large data volume. Facing the challenges posed by big data, distributed machine learning (DML) offers a solution. DML combines distributed computing with machine learning, distributing the computational workload across multiple devices to handle the increasing computation [3]. Furthermore, some distributed data, which is stored on different devices and is challenging to transfer, cannot be applied to traditional

DOI: 10.54254/2755-2721/2025.TJ24498

centralized systems, but can be processed by distributed systems [4]. DML Optimization can improve the efficiency of model training and enhance the scalability of the models.

Much research has been conducted in machine learning, resulting in the development of numerous optimization algorithms. For example, the Stochastic Gradient Descent (SGD) algorithm deals with large datasets [5]. Convolutional neural networks (CNNs) also perform well in processing high-dimensional data and are used in image and natural language processing [6]. Besides the centralized methods, optimization algorithms in DML are continuously evolving. From Parallel Stochastic Gradient Descent (PSGD) to Federated Learning (FL), they all enable the completion of machine learning tasks using multiple processors [7, 8]. These distributed optimization algorithms have expanded the ways to process data in machine learning. Some DML models require artificially allocated data, while in other models, the data is naturally distributed and cannot be transferred between devices. These algorithms can play a significant role in specific application scenarios based on their characteristics. For instance, FL is applied in fields such as finance and healthcare because of its capability to protect client privacy [8]. There have been several DML frameworks to support distributed training [9, 10]. As technology advances and big data influences more domains, further research and exploration will be required for DML optimization techniques.

The primary objective of this study is to systematically organize and summarize the key concepts and background of DML, while examining its applications from a developmental perspective. This chapter emphasizes the significance of research in DML. Chapter 2 provides a classification of the main technologies in DML and a discussion of their fundamental principles. This paper will focus on tracing the evolution of optimization techniques, with particular attention to the analysis of cutting-edge, mainstream technologies. Chapter 3 presents a detailed examination of the performance of these technologies across various application domains. This section gives special attention to the relationship between the characteristics of DML optimization methods and their corresponding applications, along with an analysis of the limitations inherent in different algorithms. Finally, Chapter 4 offers a comprehensive summary of the study, discussing the future development prospects of DML algorithms and anticipating their potential applications.

2. Methodology

2.1. Dataset description

When evaluating the performance of the MapReduce framework, the main benchmarks used are Word Count (WC), Distributed Sort (DS), Log Statistics (LS), and Inverted Index (II) [11,12]. The datasets processed include various system log files and document collections. ImageNet Dataset: The ImageNet dataset is a widely used image recognition and classification dataset created by researchers at Stanford University. This dataset includes over 14 million URLs linking to approximately 1 million labeled images. These images are divided into 22,000 categories, each containing about 500 photos. When evaluating the performance of PSGD on image recognition tasks, the ImageNet dataset was used [10]. Gboard user log data: This data is used to train the Coupled Input-Forget Gate (CIFG) next-word prediction model based on server-side training. The text input by anonymous Gboard users who agree to share text fragments while typing in Google apps constitutes this dataset [13].

DOI: 10.54254/2755-2721/2025.TJ24498

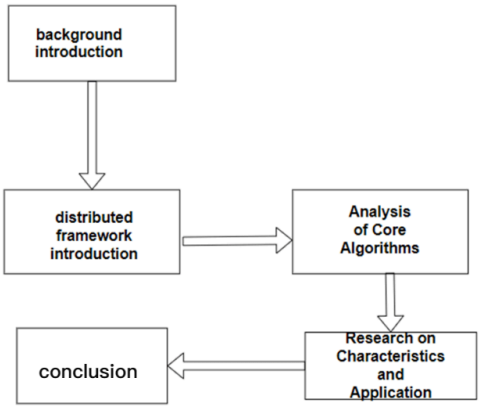

Figure 1: The pipeline of the study (picture credit: original)

2.2. Proposed approach

This chapter aims to research the key technologies and basic principles of various DML methods (shown in Figure 1). The following sections will focus on analyzing the technical processes of these optimization algorithms, which involve a comprehensive look at how these algorithms function under the hood and show their specific mechanisms for optimizing the learning process. Furthermore, taking the performance of the algorithms in applications as examples, the connections and differences among them will be analyzed. The introduction will start with the Distributed Computing Framework and extend to technologies such as PSGD and FL and their applications. MapReduce will be used as a case study for distributed computing frameworks to analyze how it supports multi-node data processing and optimizes performance through distribution. Next, the mathematical principles, implementation details, and the efficiency and scalability of PSGD will be discussed. The application of PSGD in various types of machine learning tasks will also be analyzed. FL, as an emerging technology, is attracting increasing attention. Its basic concepts, key technical challenges (such as data heterogeneity and communication efficiency), and applications in various fields will be introduced. Through specific experiments and case studies, the effectiveness of PSGD and FL in different scenarios will be demonstrated, such as in processing large-scale datasets, improving model training efficiency, and protecting data privacy.

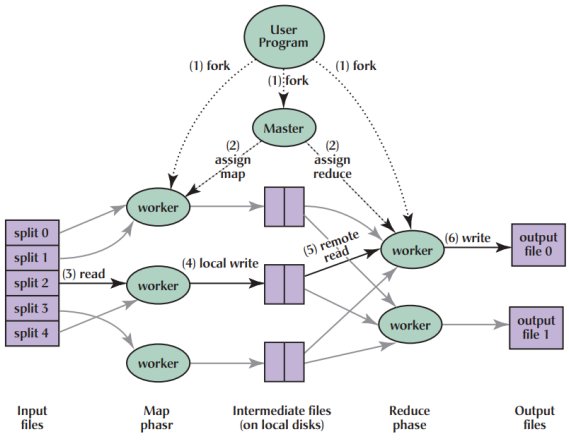

Figure 2: Execution overview of MapReduce [12]

DOI: 10.54254/2755-2721/2025.TJ24498

2.2.1. MapReduce

MapReduce is an efficient distributed computing model for processing and generating large datasets. The MapReduce framework is mainly composed of two parts: map and reduce. The user program creates a master node, which coordinates the work by assigning Map or Reduce tasks to the worker nodes. The Map function receives input key-value pairs, typically divided into multiple splits. Input data in the form of key-value pairs becomes new key-value pairs (intermediate results) after being processed by the Map function. These intermediate results will be stored on the local disks. Subsequently, the master node will coordinate the transfer of intermediate files, routing intermediate results with the same key to the same Reduce node and sorting the intermediate results for processing. Finally, the Reduce function performs a merge operation on all values for each key, generates the final results, and writes them to the output files. The above covers the basic principles of how MapReduce processes data (shown in Figure 2).

Due to its distributed structure, which facilitates scalability and high efficiency, MapReduce has been widely applied in scientific computing [9]. However, there are also some unresolved issues with MapReduce. For example, work has been done to decrease the significant communication overhead in MapReduce [5].

2.2.2. Parallelized Stochastic Gradient Descent (PSGD)

PSGD is a parallelized version of the traditional SGD algorithm, allowing each node to compute the gradient on its dataset independently. The process of the PSGD algorithm mainly includes several steps: data allocation, local SGD, and result aggregation [7]. First, each node receives a randomly assigned subset of the data. Then, during the training process, each node randomly selects samples from its assigned data subset, computes the gradients, and updates the parameters using a fixed learning rate. Repeating this process T times, the local update is completed. When the local updates are completed, the nodes send the updated parametersv_i to the master node. The global parameter is the average of all the parameters received:

$$v = \frac{1}{k}\sum_{i=1}^{k} v_i \tag{1}$$

The PSGD algorithm improves training efficiency and reduces communication overhead, making it suitable for large-scale DML. For example, Downpour SGD, a variant of PSGD that executes asynchronously, can be applied to speech and image recognition tasks and significantly enhances model performance [10]. However, PSGD also has its limitations. For example, in traditional synchronous PSGD, nodes may experience waiting times due to performance differences. As the number of nodes increases, the overhead caused by waiting also grows larger [7].

2.2.3. Federated Learning (FL)

FL is also a kind of DML method. Unlike PSGD, FL can train models without centralized raw data [7,8]. The Federated Average (FedAvg) algorithm is a key technology in FL, and its specific algorithm principle is as follows. Firstly, the server update: Initialize the model parametersx_t. During the T rounds of communication, in each round, randomly select m out of the K clients uniformly to form a setS_t. Then, the client receives the current global model and performs E iterations of updates on the local data to calculate the local parameters. In each round of iteration, a mini-batch ξj is sampled from the local datasetD_i. Finally, the server receives the updated models sent by all clients:

$$x_{t+1} = \sum_{i \in S_t} p_i x_t^{(i)} \tag{2}$$

DOI: 10.54254/2755-2721/2025.TJ24498

FL can be categorized into Horizontal FL and Vertical FL based on the distribution and features of data [8]. Horizontal FL is suitable for collecting samples with the same feature space but different sample spaces. At the same time, Vertical FL applies to data with the same sample space but different feature spaces. FL can conduct training in a distributed system while ensuring data security and privacy. Therefore, FL is particularly suitable for the next-word prediction tasks without exporting users' private data and performs better than other methods [13].

3. Discussion

MapReduce has many advantages when dealing with large-scale datasets, such as high fault tolerance. When a node fails, the master node can reassign the task to other nodes. Moreover, MapReduce abstracts away the underlying parallel and distributed processing details, allowing users to focus solely on the transformation and computation logic of the data. However, the drawback of MapReduce lies in its high latency caused by the high communication overhead, which affects MapReduce's performance in real-time computing tasks. Fortunately, there exist methods to deal with the issues above. Tiled-MapReduce decomposes a large MapReduce job into multiple smaller sub-jobs through a "tiling strategy" and iteratively processes these sub-jobs one by one [5].

The primary advantage of PSGD lies in its ability to accelerate model training on large-scale datasets significantly. By dividing the data into smaller chunks and computing gradients in parallel across multiple nodes, PSGD can fully leverage distributed computing resources, thereby significantly reducing training time [7]. However, PSGD also has some evident drawbacks. After each node computes the local gradient, it needs to send the gradient information to other nodes for aggregation, a process that may become a performance bottleneck. Moreover, suppose the data is not evenly distributed across different nodes. In that case, it may lead to some nodes being overloaded with computation while others remain idle, thereby reducing the overall training efficiency.

FL's primary advantage lies in its data privacy and security protection. Since data remains on local devices and is not uploaded to the cloud or a central server, the risk of data leakage is significantly reduced. However, FL also faces some challenges. If specific devices have smaller data volumes or lower-quality data, it may hurt the global model. And because the worldwide model needs to be iteratively updated among multiple devices, the convergence speed of FL may be slower. About the future research directions of FL, the main tasks include improving communication efficiency, optimizing model convergence speed, and addressing the issue of non-i.i.d. (non-independent and identically distributed) data. FL has broad application prospects in healthcare, transportation, and industry. For example, vehicles can act as FL clients in intelligent transportation systems and collaborate with roadside units (RSUs) for data learning without transmitting data to remote servers [11]. In addition, FL can also be applied to many other fields, such as Smart Cities and Health Monitoring [11].

4. Conclusion

This study explores the optimization and application of DML techniques, proposing a comprehensive analytical framework to evaluate the performance and characteristics of key methods, including MapReduce, PSGD, and FL. For MapReduce, data partitioning, mapping, shuffling, and reducing processes are analyzed to assess their efficiency in managing large-scale datasets. In the case of PSGD, the study delves into data allocation, local gradient computation, and parameter aggregation, highlighting its advantages in parallelization while identifying potential bottlenecks. For FL, the focus is on the privacy-preserving aspects of model training, specifically

DOI: 10.54254/2755-2721/2025.TJ24498

client-server communication, local model updates, and global model aggregation. The evaluation of these methods reveals that while MapReduce offers high fault tolerance, PSGD significantly accelerates training on large-scale datasets, and FL excels in preserving data privacy. However, each method has its limitations, such as communication overhead and convergence speed. The following research phase will focus on improving communication efficiency within FL. Specifically, attention will be given to analyzing the impact of non-i.i.d. data distribution on model convergence and exploring strategies to enhance the overall performance of DML systems.

References

[1] Abhishek-Gupta, A., Anpalagan, A., Guan, L. (2021). Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. Array, 10, 100057.
[2] Chowdhary, K.R. (2020). Natural Language Processing. In Fundamentals of Artificial Intelligence. Springer, New
[3] Joost-Verbraeken, M., Matthijs Wolting, M., Jonathan Katzy, J. (2020). A Survey on Distributed Machine Learning. ACM Comput. Surv., 53(2), 30.
[4] Emara, T.Z., & Huang, J.Z. (2020). Distributed Data Strategies to Support Large-Scale Data Analysis Across Geo-Distributed Data Centers. IEEE Access, 8, 178526-178538.
[5] Chen, R., & Chen, H. (2013). Tiled-MapReduce: Efficient and Flexible MapReduce Processing on Multicore with Tiling. ACM Trans. Archit. Code Optim., 10(1), 3.
[6] Li, Z., Liu, F., Yang, W. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. IEEE Transactions on Neural Networks and Learning Systems, 33(12), 6999-7019.
[7] Filho, C. P., Marques Jr, E., Chang, V. (2022). A systematic literature review on distributed machine learning in edge computing. Sensors, 22(7), 2665.
[8] Liu, J., Huang, J., Zhou, Y., et al. (2022). From distributed machine learning to federated learning: a survey. Knowl Inf Syst, 64, 885–917.
[9] Abualigah, L., & Masri, B. A. (2021). Advances in MapReduce Big Data Processing: Platform, Tools, and Algorithms. In Artificial Intelligence and IoT. Studies in Big Data, 85.
[10] Dean, J., Corrado, G. S., Monga, R., et al. (2012). Large scale distributed deep networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, 1223–1231.
[11] Nguyen, D. C., Ding, M., Pathirana, P. N. (2021). Federated Learning for Internet of Things: A Comprehensive Survey. IEEE Communications Surveys & Tutorials, 23(3), 1622-1658.
[12] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Commun. ACM, 51(1), 107–113.
[13] Liu, M., Ho, S., Wang, M. (2021). Federated learning meets natural language processing: A survey. arXiv preprint arXiv: 2107.12603.