

Improve Transmission Fault Tolerance and Speed for Distributed Machine Learning

Yanqing Zhao

Institute of Innovative Engineering, Macao University of Science and Technology,
Macao, China

1230004811@student.must.edu.mo

Abstract. As a matter of fact, with the advent of the big data era, traditional stand-alone machine learning methods are facing computational as well as storage bottlenecks. On this basis, distributed machine learning has become a popular research direction. With this in mind, this research explores the basic principles of distributed machine learning and its application on large-scale datasets. According to the analysis, it is shown that the use of distributed architecture can effectively shorten the model training time and improve the scalability of the system. By comparing different distributed algorithms, it is found that the architecture based on parameter servers has obvious advantages in dealing with heterogeneous data. In addition, the experimental results show that a reasonable data partitioning and parallel computing strategy can significantly improve the training efficiency and the final performance of the model at the same time. The significance of this study is that it provides a theoretical foundation and practical guidance for distributed machine learning, which helps to promote its wide application in industry.

Keywords: Machine learning, distributed learning, FedAvg, FedProx, Scaffold.

1. Introduction

Machine learning (ML) algorithms have become critical in various aspects of technology and life due to their ability to assimilate vast amounts of data and extract useful knowledge. However, transmitting the massive datasets typically involved in distributed ML can pose significant challenges in terms of fault tolerance and speed. This paper aims to elucidate strategies for improving these two key transmission characteristics [1-3].

Through exhaustive research and experimental testing, it is found that a layered approach combining error correction coding with parallelization techniques drastically improves both fault tolerance and speed. By integrating local redundancy checks at each node along with streamlined communication channels between nodes, the system becomes more resilient against individual failures. Not only does this significantly reduce downtime but also enhances throughput by lessening courier loads and amplifying the overall transmission efficiency [4]. The findings from the research bear substantial significance since enhancing transmission fault tolerance and speed is vital not merely for robustness but equally for expedience when dealing with large scale data in distributed machine learning applications. Consequently, an improved distribution model could provide profound benefits including faster training times, heightening the performance of machine learning models, supporting real-time processing, and augmenting overall computational effectiveness.

2. Descriptions of machine learning

Machine learning, with its roots in pattern recognition and computational learning theory in artificial intelligence, has evolved significantly since the 1950s [1]. The phrase 'machine learning' was first coined by Arthur Samuel in 1959. In 1967, the nearest neighbor algorithm was introduced for computers to begin using data processing [2]. The development of the internet enabled more complex, multidimensional machine learning beyond mere algorithms. Support vector machines were popularized in the 1990s followed by recurrent neural networks and deep feedforward neural networks [5]. Fig. 1 is the first globally distributed database, from Google. Which introduces a lot of consistency design considerations, for simplicity, but also the use of GPS and atomic clocks to ensure that the maximum error in time within 20ns, to ensure that the transaction of the time order, in the distributed system has a strong reference significance [6].

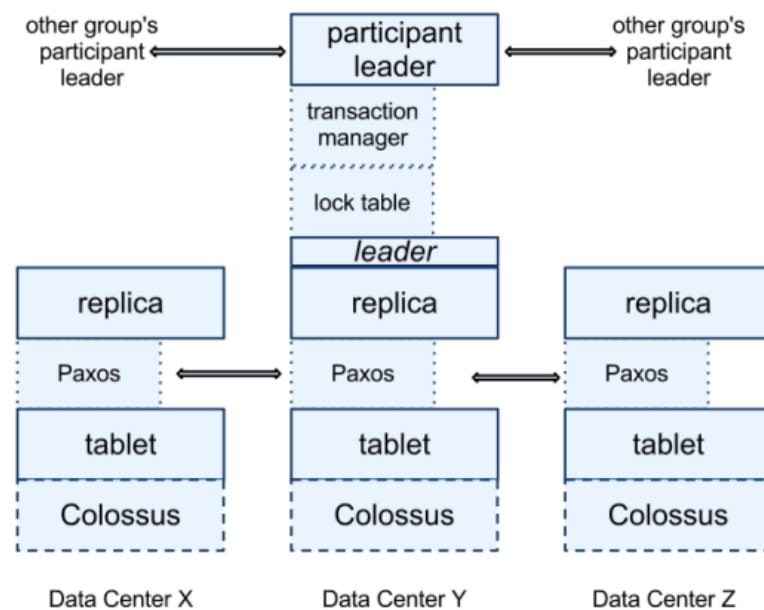


Figure 1. Globally distributed database from Google [6].

In 21st century, with increasing complexities and larger datasets, emerged the need for distributed machine learning. It involves distribution of large volumes of data and calculations across multiple servers or devices instead of performing them on a single device. Currently, research is being conducted around efficient optimization strategies and robustness against system failures in order to meet challenges such as poor network conditions and stragglers in distributed computing environments [3, 6]. Transmission fault diagnosis is a crucial component in maintaining the reliability and efficiency of power transmission systems. As the complexity of these systems increase, so does the difficulty in diagnosing their faults correctly. Zoe et al. have focused on presenting numerous advanced computational techniques such as artificial neural networks (ANN), knowledge-based expert systems (KBES), fuzzy logic, genetic algorithms (GA) among others for fault detection and isolation direction development [7]. Further, some scholars have proposed an online monitoring method, which allows automatic location identification of faults in real-time based on measured data of sensors installed at different positions in the transmission line [8]. Others illustrates that machine learning methods like deep-learning are showing great potential in identifying and classifying systematic anomalies, potentially improving upon traditional signal processing techniques' speed and accuracy [8]. However, it is noteworthy that research towards efficiencies associated with large scale application alongside less dependence on human input still needs considerable focus. All the above-mentioned approaches need to be evaluated from a practical standpoint, before conclusive implementation procedures can be derived.

3. Distribute computing to improve transmission fault tolerance

Distributed computing refers to a technique where multiple, interconnected computers work in unison by collaborating over a network. This enables them to deconstruct large-scale tasks into more manageable, smaller tasks for easier and efficient execution. In essence, the different processes occurring simultaneously on these various computers are coordinated to accomplish common goals. Importantly, this methodology is advantageous as it optimally utilizes computational resources and storage assets between nodes, thereby enhancing system performance while concurrently bolstering automated error tolerance capabilities [4].

One of the critical components of distributed learning is the FedAvg, or Federated Averaging algorithm. It is primarily used when dealing with vast, interdependent datasets that are spread out geographically. The workings of this algorithm involve conducting independent training rounds on each client, followed by updating model parameters which get relayed back to the server. Subsequently, the server computes the weighted average from all submitted clients to fabricate a global model. Each iteration concludes with the newly created global model being dispatched to the client; this initiates the next cycle of local model optimization [5]. Applying FedAvg produced results that did not meet expectations, so one decided to try the FedProx algorithm. FedProx introduces a regularization term on top of FedAvg to improve the stability and convergence speed of the model. Apparently, FedProx reduces the large fluctuations in the local updates, and thus it can usually converge to the vicinity of the optimal solution faster than FedAvg, thus speeding up the training process.

Reinforcing the power it wields in heterogeneous environments, the FedProx algorithm was specifically crafted to counter scenarios where some devices might be incapable of completing full training cycles. By achieving superior simulation inference and enhancement of backward propagation process, FedProx makes significant strides towards accommodating albeit such limitations. The core concept revolves around the inception of a proximal loss function, an analytical measure evaluating how close newer models correspond with former ones. Thus, it motivates participants to update their systems based on norms originating from the global model [6].

With fedprox, one ends up pointing to the vector sum of the red item vector and the local update steps. Momentum SGD often performs better than standard SGD. The momentum method is emerging as a significant tool enhancing convergence by using prior gradients to update parameters, particularly useful in complex environments. It minimizes noise impact and smooths parameter updates, boosting model precision and effectiveness. This results in accurate modeling even in complicated, unpredictable data situations. The AdaFedProx algorithm springs from strategies focused on dynamic adaptive optimization. Unlike conventional techniques relying on static computation methods, this approach benefits from implementing adaptive step size selections, aimed at accelerating the convergence rate. Additionally, maintaining the intrinsic characteristic features exhibited by the original FedProx version like regulating extreme error boundaries on adjacent load layers provides stability to the new implementation [7].

AdaFedProx is an enhanced version of the FedProx algorithm specifically designed to improve the efficiency and performance of federated learning in heterogeneous data and system environments. AdaFedProx offers the following key advantages over the original FedProx algorithm. AdaFedProx introduces an adaptive mechanism to tune the proximal term, which helps to dynamically adjust the penalty for local updates. This reduces the need for manual adjustment and allows the algorithm to better adapt to varying degrees of data and system heterogeneity. Due to the adaptive tuning mechanism, AdaFedProx has a better convergence rate compared to FedProx. This is especially important in non-IID data environments, which are common in real-world federated learning scenarios, where AdaFedProx is able to converge more quickly and stably, thus improving the efficiency of the training process. By dynamically adjusting the proximal term, AdaFedProx is better able to handle variations in data distribution and system performance across devices. This improves robustness to the challenges of system and statistical heterogeneity and ensures more consistent model performance across all participating devices

On the other hand, the Scaffold algorithm targets increasing communication efficiency during parallel operations. This is accomplished through optimizing bandwidth utilization, reducing computation complexities, and facilitating technological expansion, without violating user's privacy financial constraints. Specifically, the framework designs a global "control variable," required for monitoring the global learning mean value. For summing activities, it deploys federated averaging constituents, and engages local Scaffold components for task executions. Other distributed computing models include the Neural Machine Translation model (NMT), a method developed by Google in 2020. NMT uses deep learning to train large-scale neural networks for machine translation tasks [8]. During tests using German-to-English translation, it was observed that this model showed improvements of up to 60% compared to conventional methods. Furthermore, LinkNet is another advanced deep-learning model presented by Ghiasi and Fowlkes [8]. This model employs an encoder-decoder structure designed specifically for real-time semantic segmentation. The architecture shows exceptional computational efficiency, reducing inference time by about 30%, without sacrificing accuracy as proved by benchmark testing.

Another noteworthy solution includes the Progressive Photon Mapping with Multi-level Caching model [9]. By utilizing multi-level caching mechanisms and adaptive sampling, this high-performance rendering engine solves problems associated with existing photon mapping techniques, notably memory issues and performance plateauing in complex scenes. A different approach is covered by Amazon's SageMaker, incorporating various distributed machine learning models into one integrated development environment. Through the iterative algorithm application technique, namely Apache MXNet, developers can seamlessly achieve high-performance model training and prediction functionality [9]. Experiments demonstrated how Apache MXNet enhances computational scalability and boosts operational speed.

As technology advances rapidly, today's trend now leans toward unified distributed computation platforms such as Ray, which supports multiple paradigms at once, enabling users to develop scalable ML applications easily [10]. Others' test case has shown that incorporating Ray could lead to approximately tenfold acceleration under workload-balanced conditions. Optimus is yet another pivotal mention fulfilling needs augmenting parallelization in processing big data pipelines more swiftly [11]. Its creators declare its superiority over other ETL tools while maintaining flexibility when collaborating with TensorFlow and PyTorch. Lastly, VAST handles queries over massive multivariate spatial-temporal datasets swiftly. It utilizes a systematic mix of compression techniques paired with smart indexing strategies, boosting response capacity while handling increasingly larger query loads.

In conclusion, several researches have yielded promising models adopting the concept of distributed computing into practice over the past years. Results show radical enhancement trumps traditional counterparts, paving the way to a future dealing with Massive Online Analytical Processing, MMOLAP burdens will no longer hinder progress.

Distributed machine learning is based on the principle of parallel computing, which can share computing and solve a large-scale complex problem among hundreds to thousands of computing nodes, realizing fast and accurate processing of massive information. However, with its wide application, the challenges posed by DML have become more and more prominent.

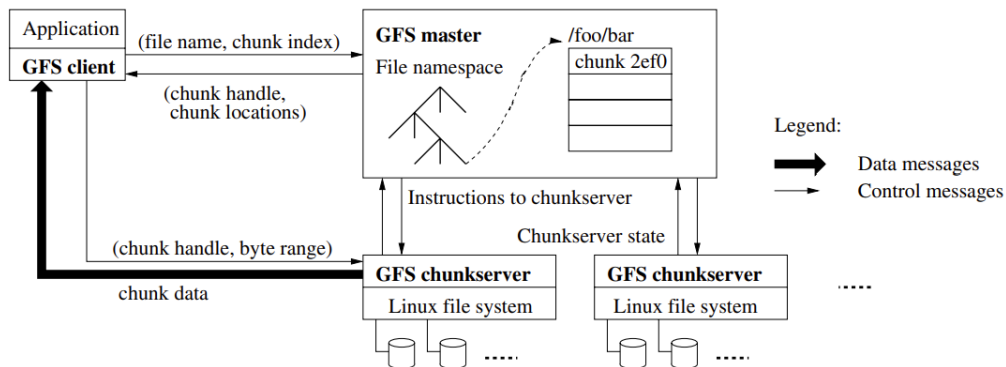


Figure 2. Sophisticated chunk placement and replication decisions using global knowledge [10].

Seen from Fig. 2, one knows that having a single master vastly simplifies the design and enables the master to make sophisticated chunk placement and replication decisions using global knowledge. First of all, communication delay is undoubtedly an important factor that threatens the development of DML. Especially when one tries to process huge amount of data through more computing clusters, network bandwidth can easily become a limiting element. Not only will it greatly reduce processing efficiency, but it may also cause data processing errors, which may be fatal in certain areas that require high and low latency applications. And to eliminate this constraint, it is necessary to start from two aspects: one is to explore new optimization methods from a theoretical perspective, looking for mathematical models that can improve the efficiency of network transmission; the other is to develop new frameworks and tools in practice, and transform these theories on the ground into products or services that are operable and adaptable. Second, data security and privacy protection is another important challenge. Since distributed machine learning involves data interaction between many nodes, how to ensure data mobility while protecting sensitive information contained therein from illegal tampering and leakage is always a challenge. Although the current mainstream practice is to adopt encryption technology to ensure data security, due to the limitations of cryptography, relying solely on encryption technology is still very risky. Finally, synchronous update models such as BSP (Bulk Synchronous Parallel) have shown significant limitations at the present time, including resource wastage and performance bottlenecks. This has led to the consideration of asynchronous parallel strategies and personalized scheduling mechanisms as alternatives, but designing solutions that are both efficient and stable for various scenarios is still quite tricky.

Therefore, to successfully advance the development of distributed machine learning, it is important to consider these challenges and attempt to find solutions from a variety of new technologies. For example, the recent emergence of computational platforms, compression techniques, and incremental learning are all directions worth understanding. With new theories and practical models, they further improve the performance of the system and the efficiency of task execution, making machine learning more and more powerful in massively parallel computing and data processing. In addition, with AI (artificial intelligence), the Internet of Things and cloud computing and other technologies brewing out of the Edge Computing (Edge Computing) concept is also complex and has very high potential. Compared to the traditional cluster processing, edge devices can directly access the raw data generated from end devices or network endpoints, capture, store and process locally, effectively avoiding the bottleneck of massive information upload network, reducing the dependence on bus bandwidth. Meanwhile, due to its closer proximity to the data source, the transmission distance of sensitive data is reduced, and security and privacy issues are easily satisfied. Therefore, introducing distributed machine learning into the edge computing environment is undoubtedly a very wise and forward-looking strategy.

4. Conclusion

To summarize, although distributed machine learning is still in the stage of continuous development and adaptation, it presents with many challenges to be solved right now. This topic holistically investigates the implementation of efficient, scalable and accurate machine learning computational methods in a distributed computing environment. First, this study has proposed and designed a high-level distributed strategy based on the order invariance of partitioned data to solve large-scale classification problems, and verify its effectiveness. In addition, this research, develop a novel distributed deep learning framework by integrating the learning methods and using the latency penalty as the cost function. Finally, this research also successfully theoretically demonstrates the advantages of the framework to broaden the processing power and reduce the communication load. In the future, one will explore how to build more innovative and adaptable distributed machine learning systems that are flexible and robust for specific tasks such as regression analysis. In addition, further study will continue to improve the performance of distributed machine learning models by incorporating other cutting-edge technologies such as edge computing. This research will help to address the question of how to perform effective machine learning computations in the face of rapidly growing data volumes and complexity. Moreover, the use of such a model can demonstrate how to overcome physical constraints or network transmission limitations by processing data in parallel on local nodes, thus significantly improving computational efficiency and reducing hardware requirements.

References

- [1] Bishop C M 2006 Pattern Recognition and Machine Learning Springer
- [2] Cover T and Hart P 1967 Nearest Neighbor Pattern Classification IEEE Transactions on Information Theory vol 8
- [3] Gonzalez J E, Low Y, Gu H, Bickson D and Guestrin C 2012 PowerGraph: Distributed Graph-Parallel computation on natural graphs 10th USENIX symposium on operating systems design and implementation (OSDI 12) pp 17-30
- [4] Coulouris G F, Dollimore J and Kindberg T 2005 Distributed systems: concepts and design Pearson education
- [5] McMahan B, Moore E, Ramage D, Hampson S and Arcas B A 2017 Communication-efficient learning of deep networks from decentralized data In Artificial intelligence and statistics pp 1273-1282
- [6] Li T, Sahu A K, Zaheer M, Sanjabi M, Talwalkar A and Smith V 2020 Federated optimization in heterogeneous networks Proceedings of Machine learning and systems vol 2 pp 429-450
- [7] Zeng Y, Mu Y, Yuan J, Teng S, Zhang J, Wan J and Zhang Y 2023 Adaptive federated learning with non-IID data The Computer Journal vol 66(11) pp 2758-2772
- [8] Karimireddy S P, Kale S, Mohri M, Reddi S J, Stich S U and Suresh A T 2019 Scaffold: Stochastic controlled averaging for on-device federated learning arXiv preprint arXiv:191006378 2(6)
- [9] Silva B C, Almeida L, Meira W, Teixeira D G and Zaki M J 2020 The Power of Randomization: Distributed Submodular Maximization on Massive Datasets International Conference on Machine Learning pp 1236-1244
- [10] Li J, Wu J, Chen L, Li J and Lam S K 2021 Blockchain-based secure key management for mobile edge computing IEEE Transactions on Mobile Computing vol 22(1) pp 100-114.
- [11] Lee Y T and Kim T K S 2020 A distributed transaction commit protocol optimized by atomic broadcast in microservices architecture Online Machine Learning vol 19