Comparative Analysis of Distributed and Parallel Machine Learning: Principles, Models, and Applications

Jiatian Li

College of Letters and Science, University of California, Santa Barbara, the United States

jiatian@ucsb.edu

Abstract. Contemporarily, the explosive growth of technology and the growth of the complexity of machine learning models facilitated the advancement of parallel and distributed machine learning. This study compared and analyzed the two paradigms from three perspectives: principles, models, and applications. This research analyzed the basic principle and some important models of parallel and distributed machine learning, mainly focusing on their operating mechanism, scalability, and performance metrics. The analysis includes how these methods can help to deal with large datasets and complex models. According to the evaluations, distributed machine learning is better in managing massive data and achieving model convergence, while parallel machine learning performs well in processing complicated models quickly. These results provided opinions on each method's advantages and disadvantages and suggestions on how to choose between the two methods in different situations. This research also stressed the importance of understanding these differences in advancing machine learning and promoting innovation in data-intensive fields.

Keywords: Parallel machine learning, distributed machine learning, complex models.

1. Introduction

Machine learning started from the fields of artificial intelligence and statistics. It can be traced back to the middle of the 20th century. Early development of machine learning was marked by automate data learning and became popular with the advent of electronic computers. The earliest studies mainly focused on symbolic reasoning and systems based on rules, as the checkers-playing program in 1959 [1]. In the early periods, cybernetics played an important role. The fundamental learning algorithms and studies of convergence rate both originated from this field [2]. The development of machine experienced three major waves. The first one was the exploration of knowledge-based systems between the 1950s and 1970s. The Bregman method introduced in 1964 was later widely used in machine learning, and the stochastic approaches first emerged during this period [2]. The second wave was related to the statistical learning methods from the 1980s to the 21st century. The backpropagation learning algorithm was invented, and it rekindled people's interest of machine learning [2]. The third one is pushed by the improvement of deep learning and the applications of big data [3] and is also what are now experiencing.

These years, the scope and scale of the study and application of machine learning were gradually expanding, which largely benefited from the rapid growth of available data and the large improvement of computational ability of modern hardware. This improvement made the exploration of new computational paradigms to process the vast amount of data and learning algorithms in machine learning necessary [3]. In these paradigms, parallel and distributed machine learning already became key approaches to scaling machine learning algorithms across multiple computing resources. Parallel machine learning involves dividing data across different nodes, with each node carrying out the computation individually and synchronously [4]. In comparison, distributed machine learning focus on crossing multiple processors or executing many operations in the cores to accelerate the computation [5]. The growth of parallel and distributed machine learning made implementing complex models and algorithms in different fields possible. In recent years, one of the most important applications is on natural language processing, in which models like GPT-3 and BERT used techniques of distributed machine learning to train on massive data on thousands of GPU. This approach allowed large scale of data processing and complex model architectures, thereby increase ability of language understanding and task generation significantly [6]. In the field of computer vision, parallel machine learning played an important role in the training of deep convolutional neural networks. It was applied in image classification, object detection, and segmentation. Using GPU largely shortened the training time, making training models on large image dataset like ImageNet possible [7]. Besides, distributed machine learning largely influenced the development of recommendation system by handling massive datasets in real time. For instance, major companies like Google and Facebook used distributed learning algorithms to manage large amount of user data [8]. This highly efficient analysis and the ability to use large-scale data improved the accuracy of the recommendation system and raised the users' experience on these platforms.

Understanding the difference between parallel and distributed machine learning was important to researchers and practitioners. Considering the application of machine learning in contemporary data-intensive environments, people can choose the most appropriate method to accomplish certain tasks. This paper aims to provide a comprehensive analysis of these two paradigms, focusing on introducing the fundamental principles, applicable models, and applications in real life. Beginning with the descriptions of parallel and distributed machine learning, this paper discussed their concepts separately, including how they improve the efficiency of model and data training. After this, the paper will shift the focus to the analysis and comparison of the two methods, using specific cases and data to stress their advantages and challenges. At last, topics such as the influences of these methods on the future and their potential limitations were discussed thoroughly, hoping to offer useful insights for people working and studying in the field of machine learning.

2. Descriptions of parallel machine learning

Parallel machine learning is a technique that separate the main task into multiple smaller subtasks. The subtasks can be carried out simultaneously on multiple processors or cores, thus accelerating the whole computing process. Parallel learning, the machine learning frame, solved the challenges of low data-process efficiency and lack of general theories used on analyzing and executing complicated learning systems. Its core is parallel system, consist of two interconnected systems: real system and artificial system. It also used external memory to store and reason the knowledge, but it went beyond memory story and provided an impact system that operates asynchronously with the real system [9].

Considering a task T, by parallel machine learning method, it should be split to n smaller, non-overlapping subtasks T1, T2, …, Tn. Each subtask was sent to operate on different processors P1, P2, …, Pn. As shown in Fig. 1, it is assumed there was no communication cost, then the total time to complete task T is about T = max (T1, T2, …, Tn). In parallel machine learning, the data can be distributed to multiple processors. Each processor uses the data it gets to carry out its subtask. Under the circumstance of training learning models, this process usually involves calculating gradients of the sub-datasets and share those gradients with the central server or aggregator to update the model parameters. The gradients represent the direction and magnitude of the changes optimizing model parameters needed [9]. After that, the updated parameters were shared with the processors, and the update process repeats again and again. In mathematics, this computing model can be described as the synchronous stochastic gradient descendent method, in which updates are parallel on the processors in each iteration, and aggregates before the weight update is applied.

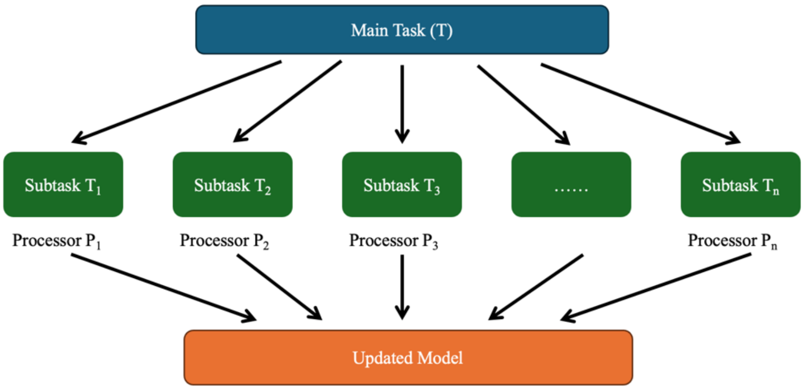

Figure 1. Schematic graph of parallel machine learning (Photo/Picture credit: Original).

There are some models that are suitable for parallel machine learning. Those include some independent tasks. Examples include k-means clustering, in which each cluster can be handled separately, and ensemble methods like random forest, in which every tree can train parallelly [10]. Deep learning model is also benefited from parallel learning frame, especially those models applied in computer vision and natural language processing, since their matrix operation is highly parallelizable [11]. Moreover, modern GPU frame aims to perform many simple operations parallelly, making it an ideal choice for tasks involving large scale neural networks. These neural networks could include millions to billions of parameters. Parallel machine learning significantly reduced the training time and remained the model's accuracy.

3. Description of distributed machine learning

Unlike parallel machine learning, distributed machine learning focuses on distributing data and computing both to the nodes in the network. Each node trains the model on its own. This method is especially efficient when processing large datasets that exceeds the memory capacity of one single machine [4]. In the system of distributed machine learning, data are divided and stored in different nodes, with every node executing the computing task to its local data subsets [12]. Distributed machine learning usually involves a distributed optimization framework. It is assumed have a dataset X. According to the distributed method, one divides it into m subsets X1, X2, …, Xm. As shown in Fig. 2, each node, namely each worker, receive their specific part of data and execute local training by using the machine learning model running on this subset. It is assumed the central model is F parametrized by a. The object is to minimize the loss function $L = \sum_{i=1}^{m} L(X_i, a)$. After the data were distributed to the workers, each node i will calculate the gradient, which is $\nabla L(X_i, a)$ in this case. The next step is for the workers to share their model parameters, including weight or gradient, with the central coordinator or with each other [4]. The gradients are usually gathered through a parameter server or ring-reduction method to update the model parameter [10]. The central model will then carry out the global model update by combining the parameters and adjust the model to show the result of the distributed model training. The updated model will be shared with all nodes to proceed further training or adjusting. This iterative process continues until the model converges and produce a final model.

Distributed machine learning is useful for models that have large amount of data, such as the recommendation system, search engine, and social network analysis [13]. For instance, some popular frameworks include Google's MapReduce and Apache Hadoop. They distribute data and computation across clusters of machines to be able to handle the processing of petabyte-scale datasets [11]. Another example is the distributed model training using frameworks like TensorFlow and PyTorch to conduct training of deep neural network. These frameworks allow model parallelism and data parallelism across multiple GPU or TPU. By using multiple GPU or TPU, the training time can be largely shortened but with the accuracy kept, making it one of the most optimal methods for complicated and deep learning tasks [14]. Training models on large datasets can be expensive and time-consuming. However, with the ability stated above, distributed machine learning method can reduce the difficulties in real world situations, like image identification and natural language processing [13].

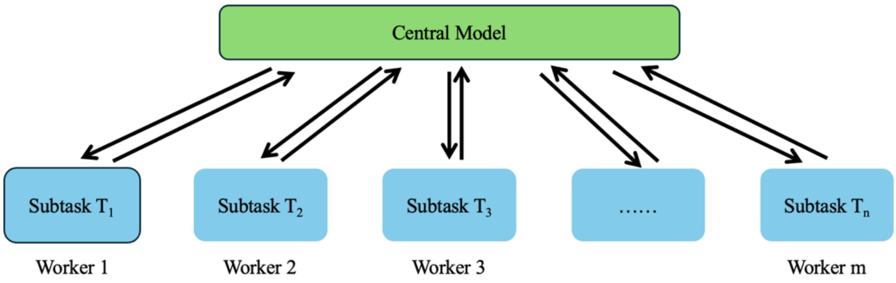

Figure 2. Schematic graph of distributed machine learning (Photo/Picture credit: Original).

4. Applications and comparisons

In recent years, the application of parallel and distributed machine learning both developed significantly. Each of them showed different strengths, mainly depending on the field of problems and the systematic framework of the models. Latest studies showed that deep learning models, especially, benefited a lot from distributed machine learning frameworks, yet parallel machine learning were more effective for tasks categorized as independent operations [15]. This section discussed the application of the two methods, and compared their performances in different cases. Since the explosive spread of the internet and the rapid growth of data, the need for data analysis and processing continue to increase. Machine learning algorithms usually require processing large scale datasets to predict or make decisions on unseen datasets. This demand led to failure of traditional single-core processors to meet the needs of machine learning tasks' computation. Therefore, parallel computing became an important technique in machine learning.

Parallel computing can break large questions down to multiple smaller problems and run at the same time on many processors. This method can reduce computing time significantly and raise the efficiency of the algorithm. In machine learning, parallel algorithm can be applied in fields like data analysis and model evaluation. Parallel machine learning is largely used in model training processes, especially in splitting models to independent tasks. One common application is ensemble methods, like random forests and gradient-boosted trees, in which each decision tree can be trained independent from any other decision trees. For instance, in random forest, multiple decision trees are trained parallelly. Each tree is trained by different data subsets, and all the results from the decision trees are collected to make the final prediction. This method is especially effective in reducing training time, and it will not sacrifice accuracy, especially in missions like categorizing and regressing [10].

Another important application of parallel machine learning is matrix operations, especially in deep learning models. Since this computing process can be spread on multiple processors, parallel machine learning framework can reduce the training times significantly. Therefore, deep learning relies heavily on matrix multiplication for backpropagation. A case involving ResNet model, a model that is used a lot in image identification tasks, showed that, comparing to a sequential approach, training models on parallel frameworks can decrease the training time by over fifty percent [16]. Parallelization plays a key role in distributed training systems. Take the system that PyTorch and TensorFlow use as an example. Its gradient calculation and model update can proceed on multiple processors at the same time. Fig. 3 showed how the time of training ResNet model can be reduced by expanding the number of GPU [17]. It proves that parallel machine learning can be scalable for deep learning models.

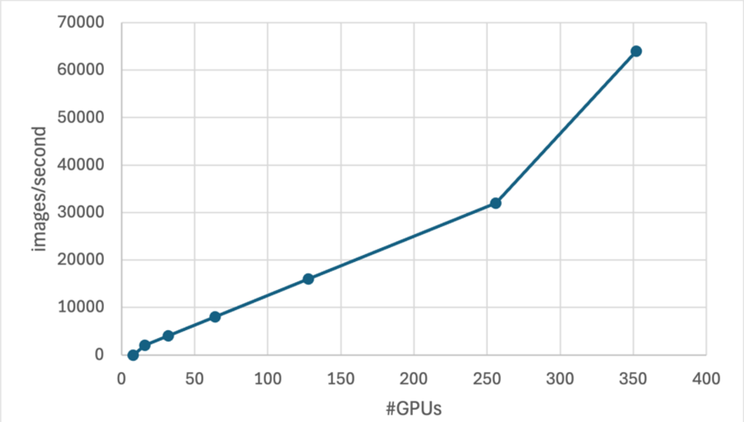

Figure 3. The run time performance of parallel machine learning algorithm [17].

Recommendation system is a key application field of distributed machine learning. It is one of the core businesses of modern internet enterprises. It analyzes users' behaviors, interests, and demands, and recommend relative commodities, services, or contents for them. As data scale increases, traditional recommendation algorithms cannot fulfill the actual demands, and thus distributed machine learning technique became a necessary approach in this field. Companies like Netflix and Amazon all rely on large data scale to recommend personalized contents. By putting the computing tasks on the same group of machines, the systems can effectively handle the data and select the mass data needed when making real decisions [18]. As shown in Fig. 4, when the number of nodes increases, the training time in recommendation system decreases [19]. Likewise, distributed machine learning is useful and shows great potential when being applied in natural language processing tasks such as the transformer-based models [6].

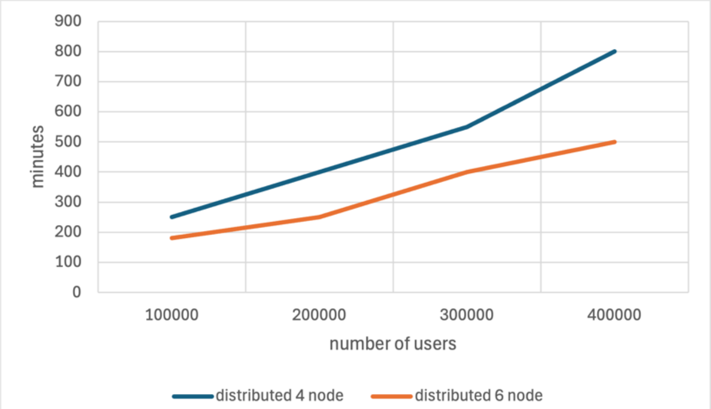

Figure 4. Run time of distributed method in recommendation system [19].

Although parallel and distributed machine learning both focus on raising efficiency and scalability, they are suitable for different missions. One of the main differences in their performance is cost of communication. In parallel machine learning, there are little communication between the processors because most tasks can be done independently. However, distributed machine learning needs to communicate between the machines frequently, especially during model update, possibly leading to delay. It can be seen from deep learning models like ResNet that, when the task can be distributed on the same machine, parallel machine learning is more efficient [20]. Another major difference between these two learning algorithms is fault tolerance. Distributed machine learning system is designed to have built-in fault tolerance. Even if some machines in the cluster is out of function, the system can still run

[4]. In comparison, parallel machine learning system that run on a single machine usually don't have the same level of fault tolerance. Thus, it is less resilient to hardware failures [5].

5. Implications, limitations and prospects

The comparison between the theories and applications of the parallel and distributed machine learning methods highlighted some important implications in this field. One of them is that distributed machine learning is relatively more scalable compared to parallel machine learning method. Distributed method can utilize large number of computational resources spread on different machines, making it perfect for large datasets in fields like medical insurance, finance, and automatic drive [20]. Although parallel machine learning is effective for smaller tasks and local computation, it is limited in scalability since it relies on shared memory system. When the data size is too large, parallel method can be slow. This means that distributed machine learning is more suitable for high computational requirement situations. Secondly, the two methods have disadvantages. For parallel machine learning, the challenge comes from limitation in storage. When dealing with complex models like deep neural network, the cost of synchronization can surpass the benefit that its speed brings. On the other hand, distributed machine learning has difficulties in communication delay and fault tolerance. Distributed machine learning transfer data between nodes frequently, which can cause delay and lower the efficiency, and it might break down when one node is out of function.

Nevertheless, parallel and distributed machine learning both have bright futures. Parallel method can be used in smaller applications such as real-time video processing and local learning environment improvement. The memory-sharing framework performs well in these situations [5]. Moreover, distributed machine learning can be applied on tasks with large data size, like natural language processing and large-scale recommendation system [20]. Future studies can also explore the combination of the two methods. Possible ways include using parallel machine learning's computation of local data to increase speed and adding distributed machine learning's scalability [6]. These studies can be helpful for meeting the growing demand of the industries.

6. Conclusion

To sum up, parallel and distributed machine learning are important methods of crossing multiple computational resources, and they both have their unique strength and weakness. Parallel machine learning method performs well in scenarios with moderate data size and benefits from local processing, while distributed machine learning is better when processing large-scale data and in decentralized systems. Although there are limitations in parallel and distributed machine learning methods, they are both of great use in certain areas. Hybrid models' exploration in the future can combine their strengths and provide solutions for more machine learning problems. Parallel machine learning is used in ensemble methods and matrix operations, and distributed machine learning method is applied on recommendation systems and language processing tasks. In the future, combining both of their strengths can fulfill the need for higher machine learning abilities.

References

[1] Samuel A L 1959 Some Studies in Machine Learning Using the Game of Checkers BM Journal of Research and Development vol 3 p 3

[2] Fradkov A L 2020 Early History of Machine Learning IFAC PapersOnLine vol 53 p 2

[3] Bengio Y 2015 Introduction In Historical Trends in Deep Learning Nature vol 193 pp 11–25

[4] Verbraeken J, Wolting M, Katzy J, Kloppenburg J, Verbelen T and Rellermeyer J S 2020 A Survey on Distributed Machine Learning ACM Computing Surveys vol 53 p 2

[5] Upadhyaya S R 2012 Parallel approaches to machine learning—A comprehensive survey Journal of Parallel and Distributed Computing vol 73 p 3

[6] Shoeybi M, Patwary M, Puri R, LeGresley P, Casper J and Catanzaro B 2019 Megatron-LM: Training multi-billion parameter language models using model parallelism arXiv:190908053

[7] Krizhevsky A, Sutskever I and Hinton G E 2017 ImageNet classification with deep convolutional neural networks Communications of the ACM vol 60(6) pp 84-90

[8] Zhou G, Mou N, Fan Y, Pi Q, Bian W, Zhou C and Gai K 2018 Deep Interest Evolution Network for Click-Through Rate Prediction Proceedings of the AAAI Conference on Artificial Intelligence vol 33(1) pp 5941-5948

[9] Li L, Lin Y, Zheng N and Wang F Y 2020 Parallel learning: a perspective and a framework IEEE/CAA Journal of Automatica Sinica vol 4 p 3

[10] Zinkevich M, Weimer M, Smola A J and Li L 2010 Parallelized stochastic gradient descent Advances in Neural Information Processing Systems vol 23 pp 2595-2603

[11] Dean J and Ghemawat S 2008 MapReduce: Simplified data processing on large clusters Communications of the ACM vol 51 p 1

[12] Xing E P, Ho Q, Xie P and Wei D 2016 Strategies and Principles of Distributed Machine Learning on Big Data Engineering vol 2 p 2

[13] Agarwal A, Chapelle O, Dudik M and Langford J 2014 A Reliable Effective Terascale Linear Learning System The Journal of Machine Learning Research vol 15 p 32

[14] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J and Zheng X 2016 TensorFlow: A system for large-scale machine learning 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI) pp 265-283

[15] Scutari G, Facchinei F, Lampariello L, Sardellitti S and Song P 2017 Parallel and distributed methods for constrained nonconvex optimization-part II: Applications in communications and machine learning IEEE Transactions on Signal Processing vol 65(8) pp 1945–1960

[16] Nazir A, Mir R N and Qureshi S 2020 Exploring compression and parallelization techniques for distribution of deep neural networks over Edge–Fog continuum – a review International Journal of Intelligent Computing and Cybernetics vol 13 p 3

[17] Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A and He K 2017 Accurate Large Minibatch SGD: Training ImageNet in 1 Hour arXiv:170602677

[18] Ko H, Lee S, Park Y and Choi A 2022 A Survey of Recommendation Systems: Recommendation Models Techniques and Application Fields Electronics vol 11 p 1

[19] Dicky T, Erwin A and Ipung H P 2020 Developing a Scalable and Accurate Job Recommendation System with Distributed Cluster System using Machine Learning Algorithm Journal of Applied of Information Communication and Technology vol 7 p 2

[20] Hegde V and Usmani S (2016) Parallel and distributed deep learning vol 5(31) pp 1-8.