Distributed Machine Learning: A Survey

Naima Nazly
naima.nazly@g.bracu.ac.bd

Fahim Hasan Mehedi
fahimhasanmehedi@gmail.com

M S R Momen Sagor
msr.momen.sagor@g.bracu.ac.bd

Shanuma Afrin Meghla
shanuma.afrin.meghla@g.bracu.ac.bd

Md Sadaf Noor
md.sadaf.noor@g.bracu.ac.bd

Md Sabiir Hossain
md.sabbir.hossain1@g.bracu.ac.bd

Annajiat Alim Rasel
annajiat@gmail.com

***Abstract*— Machine learning has seen a huge increase in demand over the last decade. However, in order to increase prediction quality and make machine learning systems suitable for more complex applications, a substantial amount of training data is required. Simple machine learning models can be trained with small amounts of data, but the input for bigger models like neural networks rises exponentially as the number of parameters grows. The demand for training data processing has exceeded computing capability, forcing the distribution of machine learning workload over several computers and the conversion of a centralized to a distributed system. This survey covers the challenges of distributed machine learning, discussing the distributed machine ecosystem and algorithms.**

Keywords—distributed systems, machine learning, distributed ml, big data, Hadoop, TensorFlow.

I. INTRODUCTION

Multi-node machine learning methods and systems are referred to as distributed machine learning. They are designed to improve performance, boost accuracy, and scale to bigger input data quantities. For many algorithms, increasing the input data amount reduces the learning error and is often more effective than utilizing more complex methods. Companies, researchers, and people can use distributed machine learning to make decisions and derive useful conclusions from enormous amounts of data.

Distributed machine learning can be used to offload computationally-intensive gradient computations to multiple workers. So and Güler, in their research, proposed a fast and scalable method to train a logistic regression model for that [1]. It is promising in solving energy trading problems among enthusiasts of the respective field [2]. It can be used for various kinds of analysis and detection very efficiently. Iktishaf+, introduced by Alomari et. al is a very good example in this field. It is a software tool that uses big data and distributed machine learning methods to predict different kinds of detections and analyses based on Arabic tweets [3][4].

II. BACKGROUND

Because of multi-core processors and cloud computing platforms, parallel and distributed computing systems have lately become increasingly popular and approachable. As a result of this evolution, numerous distributed and parallel computing strategies have spawned a plethora of complexities that had previously been labeled as centralized and sequential systems. Machine learning (ML) can use distributed computing to manage larger volumes of data or to educate/learn data that is inherently distributed using this approach.

In this survey, we focus on data that is intrinsically distributed but cannot be moved across systems. Only a few strategies for constructing distributed algorithms are accessible in this mode.

A. Data Fragmentation into Distributed Databases

In practice, training data is a collection of examples, each of which has multiple features. Because of the distributed structure of the data set, there are two frequent types of data fragmentation.

i. Horizontal fragmentation, in which separate sites store subsets of instances.

ii. Vertical fragmentation, in which separate sites store portions of an instance's properties. The majority of data sets are horizontally fragmented because this is the most appropriate and natural call for the majority of applications.

B. Why Distributed Databases?

Your business's lifeblood is data, which is why you need a database at the heart of it all. However, not all databases are capable of meeting today's enterprises' rising data requirements. You'll need a distributed database system in particular, which will allow you to easily innovate and evolve.

C. How Distributed Database Works?

A distributed system is a collection of networked computers that appear to function as though they are one. A distributed database management system (DBMS), which is abstracted to the user as a single database, uses multiple nodes to store data of different systems. Applications do not need to know the actual node location where the data is kept since distributed databases provide location transparency. When a query is conducted on a distributed database, it is answered by a group of sites from several data centers working together.

D. Types of Distributed Databases

There are two types of distributed database systems, depending on the design homogeneous and heterogeneous. If consistency is important to you, a homogeneous architecture is the way to go. Physical resources, operating system, and database management system are all the same across all of the sites in this example. Database site setup and management become easier with homogeneous architectures.

If further customization is necessary, heterogeneous architecture is recommended. Various sites can have different attributes because to heterogeneous database architectures.

E. Storage Methods

Fragmentation & replication are the data storage methods for distributed database. Splitting data into smaller parts and

spreading those chunks across the many sites of a distributed database is known as fragmentation or partitioning.

F. Data Management

Replication makes copies of the data items that stores more than a location and can be access at any point of time. Replication does users to rely on it to make decisions and ensure availability when need. Distributed database designates one site as primary site and chronically syncs other sites with primary.

G. Benefits

We are in a world where data is everything and essential part of lives, distributed database lie at the center of data infrastructure. End-users unknowingly interacts with fancy web or app but at the backend distributed database is powering up.

III. Machine learning algorithms

Dividing the learning process among numerous workstations is a logical approach of scaling up learning algorithms, distributed learning appears to be a promising avenue of research. ML algorithms learn to produce data-driven judgments or predictions.

The three qualities that we use to classify contemporary machine learning techniques are as follows: feedback, purpose and method.

There are numerous forms of feedback available.

A. Supervised Learning

Training data for supervised learning consists of input objects (typically vectors) and the desired output values. The goal of supervised learning algorithms is to discover a function that maps the input data to the desired output. The outcome may then be predicted by applying this function to new input data. One of the objectives is to reduce the projected findings' bias and variance error. The bias mistake is created by the learning algorithm's simplifying assumptions in order to make learning the target function easier. On issues that do not fully fulfil the assumptions, however, approaches with significant bias have worse predictive performance.

B. Unsupervised Learning

Unsupervised learning mainly trains the system using unlabeled data. The aim of this learning method is to create a function that describes the data structure and analyze the raw input data. Because the input data is unlabeled, there is no visible output accuracy statistic. The most typical use of unsupervised learning is to clustering data together. Clustering is mostly based on similarities and hidden patterns between the data. Dimensionality reduction is another area where unsupervised learning algorithms perform efficiently. Dimensionality reduction is the process of isolating the most important features of data. In this situation, the feedback is generated using a similarity metric.

C. Semi-Supervised Learning

Semi-supervised learning combines a little quantity of labelled data with a big amount of unlabeled data to get results. Clustering can be used to extrapolate known labels onto data points that haven't been labelled. This is based on the premise that data points with similar labels have the same label.

D. Reinforcement Learning

Reinforcement learning is a method of training an agent about its surrounding environment using trial-and-error method. Normally, a goal is set for the machine in this learning method. If the machine does something that helps it to achieve the goal, it gets a positive reward. Otherwise, it gets a negative reward. If it gets negative reward, it tries to avoid the task. Cost function is used to examines the systems and feedback is based on this function. The main obstacle here is the credit assignment problem, or deciding whether behaviors truly lead to higher long-term benefits. A local incentive system is important for the scalability of the learning algorithm as global incentive system is not efficient for scaling here.

.

IV. Frameworks for ML in distributed systems

There are different frameworks for machine learning in distributed systems. Apache Spark is emerging rapidly as a general-purpose distributed system as the systems are dealing with massive amount of data nowadays. Libraries like MLib in Apache Spark is popular among the researchers and developers. There are some other frameworks such as Hadoop, TensorFlow (Google), NCCL (Nvidia) and CNTK (Microsoft), which are also used extensively. In this paper, we will discuss in detail about two frameworks for two different paradigms of distributed machine learning.

A. Hadoop Ecosystem for Big-Data Analysis

The Hadoop ecosystem was developed by Apache. It is an open-source project and is now used for solving problems related to big data and its computation. Hadoop adopted the Google File System (GFS) architecture. It has different modules for collecting, storing, and processing data [5]. The Hadoop ecosystem mainly has five modules, which are Hadoop distributed file systems (HDFS), MapReduce, YARN, Hadoop common and Hadoop Ozone.

B. Modules of Hadoop

HDFS is the Hadoop ecosystem's storage layer. It's a distributed file system that's optimized for low-cost hardware. The HDFS design is essentially a master-slave architecture. A NameNode and multiple DataNodes make up an HDFS cluster. NameNode serves as a master server in this scenario, keeping information about file locations and metadata. Read and write requests from clients or users are typically handled by the DataNodes [6].

MapReduce is a framework within Hadoop ecosystem specially for processing data. It has two parts: “Map” and “Reduce”. Map usually transforms a set of data to another by breaking those data into key-value pairs. Reduce takes those pairs as an input and combines them to generate a smaller set of data.

YARN or Yet Another Resource Negotiator is a distributed operating system. The basic idea of YARN is to split up the resource management layer and job scheduling [7]. Both of these layers were tightly coupled prior to Hadoop 2.0; however, they were split when YARN took over the resource management layer. If an application wishes to run under YARN, its client must ask the resource manager to start an

application manager process. After that, this process seeks a node manager. The application process then runs in a container created by the node manager [5].
Hadoop common is usually the common utilities, which can be used by other Hadoop modules.
Hadoop ozone is mainly a distributed key-value store built on Hadoop Distributed Data Store. Ozone is a highly scalable storage system that overcomes the limitations of standard HDFS [8].

C. Data Processing in Hadoop

The Hadoop ecosystem can be divided into storage, processing, and management layers. HDFS is Hadoop's storage layer, YARN is the resource management layer, and MapReduce is the processing layer. Other frameworks apart from these can also be integrated into the ecosystem.
The analysis is mainly done in the processing layer with the help of the MapReduce engine. The Hadoop implements MapReduce using two types of components: a Job-Tracker and a large number of Task-Trackers. MapReduce has three phases in the whole analysis process. They are map, shuffle or merge and reduce. Task-Trackers process data using map and reduce functions by the command of the Job-Tracker. Scheduling of those functions is done by the Job-Tracker. It acts as a monitor of the whole analysis process.
Apache Spark is a MapReduce-based framework for data analysis that addresses the constraints of standard MapReduce. MLib is a machine learning library in Apache Spark which can implement variety of learning algorithms for various problems. These are the two most common processing engines for big-data processing. Other data processing engines include Flink, Storm, and H2O, in addition to MapReduce and Spark [9].

D. Tensorflow for Deep Neural Networks

TensorFlow is a large scale and open-source machine learning system developed by google. It can be used for traditional machine learning, but it is mainly developed with a focus on training and inference on deep neural networks. In TensorFlow, a concept called dataflow graphs represents shared state, computation, and operations that mutate the state. It maps dataflow graph nodes across multiple machines in a cluster as well as inside a single machine's computing components [10]. Google used DistBelief before, which was a distributed system for training neural networks and predecessor of TensorFlow.

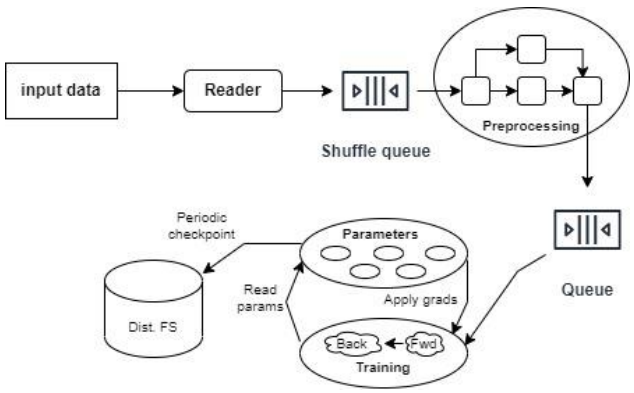

Figure 1: dataflow graph (training pipeline)

A single dataflow graph in TensorFlow represents all of a machine learning algorithm's computation and state (Figure 1), as well as the different mathematical operations, parameters and their update rules, and input preprocessing. The dataflow graph explicitly specifies communication between sub-computations, making distributed execution easier. As a result, the same TensorFlow software can be utilized on the GPU, TPU, and cellphone for training, serving, and mobile interface [10].

V. Conclusion

We are producing an enormous amount of digital data through our actions. We are still learning to use machine learning in processing those data to make our life better. More efficient handling of that enormous amount of data will be the key factor for humanity to move to the next human evolution and uplift our status in society. Even though in terms of hardware we have come a long way, now we can make use of our supercomputer to solve our problems. The problem that we are trying to solve requires an even more efficient machine with the data that we collectively produce, often it is not possible for a single supercomputer to process all at once. Also, another challenge in neural networks is when an application is deployed with hardware accelerators, by nature deep neural networks are not always resilient to hardware errors. That's where the necessity of Distributed Machine Learning comes into the picture. To efficiently use distributed machine learning we need to get over the challenge that comes with distributed machine learning. The most challenging aspect of distributed systems is to efficiently parallelize the training process and coherent model creation. We have to figure out an efficient and effective way to improve concurrent processing. At the same time which will allow us the combine the result into a single coherent model.

References

[1] Jinhyun So, Ba¸sak Güler and A. Salman Avestimehr. 2021. *CodedPrivateML: A Fast and Privacy-Preserving Framework for Distributed Machine Learning*. arXiv:1902.00641. Version 2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Ning Wang, Jie Li, Shen-Shyang Ho and Chenxi Qiu. 2021. *Distributed machine learning for energy trading in electrical distribution system of the future.* https://doi.org/10.1016/j.tej.2020.106883.

[3] Ebtesam Alomari, Iyad Katib, Aiiad Albeshri, Tan Yigitcanlar and Rashid Mehmood. 2021. *Iktishaf+: A Big Data Tool with Automatic Labeling for Road Traffic Social Sensing and Event Detection Using Distributed Machine Learning.* https://doi.org/10.3390/s21092993.

[4] Ebtesam Alomari, Iyad Katib, Aiiad Albeshri, and Rashid Mehmood. 2021. *COVID-19: Detection Government Pandemic Measures and Public Concerns from Twitter Arabic Data Using Distributed Machine Learning.* https://doi.org/10.3390/ijerph1801**0282.**

[5] Landset, S., Khoshgoftaar, T. M., Richter, A. N., & Hasanin, T. (2015). A survey of open source tools for machine learning with big data in the Hadoop ecosystem. Journal of Big Data, 2(1), 1-36.

[6] Borthakur, D. (2008). HDFS architecture guide. Hadoop apache project, 53(1-13), 2.

[7] Apache Hadoop 3.3.1 – Apache Hadoop YARN. Hadoop.Apache.Org. https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html

[8] *Apache Hadoop Ozone*. Apache. https://hadoop.apache.org/ozone/

[9] Wang, Y., Jiao, Y., Xu, C., Li, X., Wang, T., Que, X., ... & Yu, W. (2012). Assessing the performance impact of high-speed interconnects on mapreduce. In Specifying Big Data Benchmarks (pp. 148-163). Springer, Berlin, Heidelberg.

[10] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (pp. 265-283).