Volume 3, Issue 2(Page 1-4)
ISSN: 2709-0248

A Review of Machine Learning for Big Data Analysis

Nadia Mahmood Hussien[1], Samira Abdul-Kader Hussain[1], Khlood Ibraheem Abbas[1], Yasmin Makki Mohialden[1]

Corresponding Email: nadia.cs89@uomustansiriyah.edu.iq

[1]Computer Science Department, Collage of Science, Mustansiriyah University, Baghdad, Iraq

Abstract

Big data is the key to the success of many large technology companies right now. As more and more companies use it to store, analyze, and get value from their huge amounts of data, it gets harder for them to use the data they get in the best way. Most systems have come up with ways to use machine learning. In a real-time web system, data must be processed in a smart way at each node based on data that is spread out. As data privacy becomes a more important social issue, standardized learning has become a popular area of research to make it possible for different organizations to train machine learning models together while keeping privacy in mind. Researchers are becoming more interested in supporting more machine learning models that keep privacy in different ways. There is a need to build systems and infrastructure that make it easier for different standardized learning algorithms to be created. In this research, we look at and talk about the unified and distributed machine learning technology that is used to process large amounts of data. FedML is a Python program that let machine learning be used at any scale. It is a unified, distributed machine learning package.

Keywords: Big Data Analysis, Fedml , Machine Learning, DDB, Python

Received: May 17, 2022
Received in Revised: June 25, 2022
Accepted: July 2, 2022

Introduction

Significant advances in distributed computing and parallel processing technologies have occurred concurrently with the maturation of DBMS technology. Because of this change, there are now a lot of distributed database management systems and parallel database management systems.

These systems are beginning to establish themselves as a means of managing traffic for applications dealing with large amounts of data. A distributed database, also called a DDB, is a set of multiple logically linked databases spread across a network of computers.

A distributed database management system is a piece of software that manages a database that is spread out but doesn't tell users where it is. DBMS implementation can be done in many different ways. The client/server topology is the simplest way to set up a computer network. The database is spread out over a number of servers and uses a number of client-server systems. This gives the database more flexibility. User requests are sent from each client to its own "master" server. When a user makes a request or does a transaction, they don't know that the server is connected. Most modern database management systems are either client-server or hybrid client-server. The distributed database management system doesn't care whether a computer is a client or a server. Each site can do tasks for both clients and servers. Peer-to-peer architectures use complex protocols to manage data across many nodes.

Machine learning (ML) lets programmers make programs that "learn" from the information that is sent to central servers. Transferring data can be expensive and put a user's privacy at

DOI: https://doi.org/10.47667/ijpasr.v3i2.154

risk. FedML, or Unified Machine Learning, solves these problems by training locally on client devices. After these stages, only light, grouped data is sent. FedML systems should have the same response times and accuracy as other ML applications, even though they are managed by devices that may be different, may crash, and aren't as powerful as cloud servers (Chamikara et al., 2021; He et al., 2020).

Related Work

In [2020], They changed federated learning into FedGKT, a group knowledge transfer training algorithm. FedGKT uses a version of the alternating minimization approach to train small CNNs on edge nodes and periodically transfer their knowledge to a large server-side CNN through knowledge distillation. FedGKT combines several benefits into a single framework: less need for computation at the edge, less communication bandwidth for large CNNs, and asynchronous training, all while keeping model accuracy similar to FedAvg. We train CNNs based on ResNet-56 and ResNet-110 using three different sets of data (CIFAR-10, CIFAR-100, and CINIC-10) and their non-IID versions. Based on our results, FedGKT can be as accurate as FedAvg or even a little more accurate. More importantly, FedGKT makes edge training cheaper. FedGKT on edge devices requires 9-17 times less computational power (FLOPs) and 54-105 times fewer parameters in the edge CNN than FedAvg (He et al., 2020).

in [2021], FedML systems were used as a unique example of self-adaptive applications, where clients and servers must work together to get the desired results. In particular, this paper proposed formalizing FedML applications as self-adaptive systems. A prototype shows that the approach is possible, and an early evaluation shows that the proposed solution is useful (Baresi et al., 2021).

In the same year [2021In this paper, I will explain what FLSs are and how they work. They define FLSs and look at the parts of the system to better understand the key parts of the design system and to help guide future research. We also give a full classification of FLSs based on six different factors, like data distribution, machine learning model, specificity mechanism, communication architecture, federation size, and federation induction (Li et al., 2021).

Also in 2021, They will build the FedIoT platform, which will have the FedDetect algorithm for finding weird data on the device and a system design for testing federated learning on IoT devices in a realistic way. Also, the FedDetect learning framework proposed uses a local adaptive optimizer and a cross-round learning rate scheduler to increase performance. In a network of real IoT devices (Raspberry PI), they test both the model and the system performance of the FedIoT platform and the FedDetect algorithm. The results show that federated learning is a good way to find more types of attacks that happen on more than one device. The efficiency of the system shows that both the total training time and the cost of memory are reasonable and promising for IoT devices with limited resources (He et al., 2021). He wants to build FEDn in 2022. Since there are now several projects that can simulate federated learning, the algorithmic parts of the problem are moving along quickly. But there aren't any federated machine learning frameworks that focus on fundamental things like scalability, robustness, security, and performance in a geographically distributed setting. So, they came up with and built the FEDn framework to fill this gap. One of the great things about FEDn is that it lets you train across devices and silos. This makes FEDn a powerful tool for researching a wide range of machine learning applications in the real world (Ekmefjord et al., 2021).

DOI: https://doi.org/10.47667/ijpasr.v3i2.154

Implementation of Federated Learning

Federated Learning, also called FedML, is a method of distributed computing that was designed to make machine learning work well and protect users' privacy in a distributed setting. In FedML, the people who own the data put together machine learning models on their own computers. A coordinating server, like a server in the cloud, is used (e.g., edge devices) to make a global model and share machine learning (ML) knowledge between the different entities. FedML is thought to protect the privacy of raw data because the original data never leaves the devices of the people who hold the data. But privacy inference attacks, such as model-memorizing attacks and membership inference, can be used to figure out what the model is doing. Even when black-box security is in place, these kinds of attacks try to get private information from ML models that have been trained. This kind of technology can also be used in healthcare and open banking, which are often spread out geographically and don't have the right ways to share data without compromising privacy.

Unified learning uses an iterative process known as a unified learning round in order to achieve the same level of performance as centralized machine learning. This method involves many exchanges between the server and the client. Each iterative learning interaction or round begins with the propagation of the current or updated global model state to the contributing nodes (participants), followed by training the local models on those nodes to produce certain potential model updates from the nodes, and finally processing and aggregating the updates from local nodes into an aggregated global update so that the central model accordingly.FedMLPython library.

FedML makes machine learning APIs easy to use. They can be used anywhere and at any scale. In other words, FedML supports both federated learning for data silos and distributed acceleration training using MLOps and open source, including both industry use cases and cutting-edge academic research.

Distributed Training: A small cheetah can help model training go faster. Simulator: (1) simulates FL using a single process. (2) an MPI-based FL simulator (3) The fastest FL simulator is based on the NCCL. Federated learning across devices for smartphones and Internet of Things (IoT) devices, including Android and iOS edge SDKs and embedded Linux. Federated cross-silo learning with a Python-based edge SDK for cross-organization/account training. Model Serving: We are committed to providing superior service to edge AI users. MLOps is the FedML machine learning operation pipeline for AI that works anywhere and at any scale.

Source Code Structure

The following describes how each package works:

FedML's core is its basic API package. This package uses communication backends like MPI, NCCL, MQTT, gRPC, and PyTorch RPC to make distributed computing work. It can also manage the topology. Other low-level APIs for security and privacy are also supported. All algorithms and scenarios are built on top of the "core" package.

FedML will provide some default datasets for users to get started. There are also templates that can be changed to fit your needs. device: FedML model zoo. model: FedML computing resource management.

FedML parrot can support (1) simulation FL using a single process, (2) FL simulator using MPI, and (3) FL simulator using NCCL (fastest), cross-silo: federated cross-silo learning for training across organizations or accounts cross-device: cross-device Federated Learning for

DOI: https://doi.org/10.47667/ijpasr.v3i2.154

IoT Devices and Smartphones: Distance learning: Using lightweight materials can speed up the training of models. Cheetah serve is a model serve that has been changed to work with edge inference. Some examples of centralized trainer code that can be used as a standard. Utils are functions that are used by other modules.

Conclusion

Unified Learning makes it possible to train a machine learning model and a deep learning model at the same time for mobile edge network optimization. It is sometimes called both "combined learning" and "spread out learning." With FL software, it is possible to connect a lot of nodes together to make a collaborative learning model. Then, this model can be used to solve important problems like access rights, different ways to get to different kinds of data, privacy, and security. This method of distributed learning can be used in many areas of business, such as healthcare, communications, forecasting, traffic control, and artificial intelligence. Healthcare, the Internet of Things, transportation, self-driving cars, and the pharmacy.

References

Baresi, L., Quattrocchi, G., & Rasi, N. (2021, May). Federated machine learning as a self-adaptive problem. In *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)* (pp. 41-47). IEEE.

Chamikara, M. A. P., Bertok, P., Khalil, I., Liu, D., & Camtepe, S. (2021). Privacy preserving distributed machine learning with federated learning. *Computer Communications*, *171*, 112-125.

Ekmefjord, M., Ait-Mlouk, A., Alawadi, S., Åkesson, M., Stoyanova, D., Spjuth, O., ... & Hellander, A. (2021). Scalable federated machine learning with FEDn. *arXiv preprint arXiv:2103.00148*

He, C., Annavaram, M., & Avestimehr, S. (2020). Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, *33*, 14068-14080.

He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., ... & Avestimehr, S. (2020). Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*.

He, C., Shah, A. D., Tang, Z., Sivashunmugam, D. F. N., Bhogaraju, K., Shimpi, M., ... & Avestimehr, S. (2021). Fedcv: a federated learning framework for diverse computer vision tasks. *arXiv preprint arXiv:2111.11066*.

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., ... & He, B. (2021). A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*.

DOI: https://doi.org/10.47667/ijpasr.v3i2.154