Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Machine Learning Meets Edge Computing: Trends and Innovations in Distributed AI

Dr. P S Sumathi[1], Dr. S. Belina V J Sara[2], T. Kanimozhi[3]

[1]Assistant Professor, Department of Computer Science, Government Arts and Science College, India
[2]Assistant Professor, Department of Computer Applications, Faculty of Science and Humanities, SRM Institute of Science and Technology, Kattankulathur, Chengalpet, India
[3]Assistant Professor, Department of Computer Science and Applications, Faculty of Science and Humanities, SRM Institute of Science and Technology, Ramapuram, 600089,Chennai, India.

In recent years, the convergence of machine learning (ML) and edge computing has revolutionized how data is processed and analyzed. As the demand for real-time insights grows across industries, the integration of these technologies has become critical for enabling distributed AI systems that are efficient, scalable, and responsive. Traditionally, ML models have been trained and deployed in centralized cloud environments. While this approach offers significant computational power and scalability, it also comes with challenges such as latency, bandwidth consumption, and data privacy concerns. Edge computing addresses these issues by bringing computation closer to the data source, reducing the need to transmit large volumes of information to centralized servers. Reduced Latency: Real-time applications like autonomous vehicles and industrial automation require rapid decision-making, which edge computing facilitates by minimizing data transmission delays. Enhanced Privacy: Sensitive data can be processed locally, reducing the risk of exposure during transmission. Bandwidth Efficiency: By processing data at the edge, organizations can significantly lower their bandwidth usage and costs.

Keywords: Machine learning, distributed machine learning, Internet of Things, edge computing, cloud computing, edge-cloud continuum.

1. Introduction

The convergence of machine learning (ML) and edge computing is transforming how data is processed and analyzed across various sectors. With the increasing need for real-time insights, this integration is paving the way for distributed AI systems that are not only efficient and scalable but also responsive to dynamic environments. By combining the predictive power of ML with the proximity advantages of edge computing, organizations can achieve unprecedented levels of performance and innovation.

Traditionally, ML models have been trained and deployed in centralized cloud environments. While this approach offers significant computational power and scalability, it also comes with

challenges such as latency, bandwidth consumption, and data privacy concerns. Edge computing addresses these issues by bringing computation closer to the data source, reducing the need to transmit large volumes of information to centralized servers.

Therefore, in Edge Intelligence, it is essential to promote collaboration between devices to compensate for their lower computing capacity. Some synonyms of this concept found in the literature are: distributed learning, edge/fog learning, distributed intelligence, edge/fog intelligence and mobile intelligence.

The leverage of edge intelligence reduces some drawbacks of running ML tasks entirely in the cloud, such as:

High latency: offloading intelligence tasks to the edge enables achievement of faster inference, decreasing the inherent delay in data transmission through the network backbone;

Security and privacy issues: it is possible to train and infer on sensitive data fully at the edge, preventing their risky propagation throughout the network, where they are susceptible to attacks. Moreover, edge intelligence can derive non-sensitive information that could then be submitted to the cloud without further processing;

The need for continuous internet connection: in locations where connectivity is poor or intermittent, the ML/DL could still be carried out;

Bandwidth degradation: edge computing can perform part of processing tasks on raw data and transmit the produced data to the cloud (filtered/aggregated/pre-processed), thus saving network bandwidth. Transmitting large amounts of data to the cloud burdens the network and impacts the overall Quality of Service (QoS);

Power waste: unnecessary raw data being transmitted through the internet demands power, decreasing energy efficiency on a large scale.

The steps for data processing in ML vary according to the specific technique in use, but generally occur in a well-defined life cycle, which can be represented by a workflow. Model building is at the heart of any ML technique, but the complete life cycle of a learning process involves a series of steps, from data acquisition and preparation to model deployment into a production environment. When adopting the Edge intelligence paradigm, it is necessary to carefully analyze which steps in the ML life cycle can be successfully executed at the edge of the network. Typical steps that have been investigated for execution at the edge are data collection, pre-processing, training and inference.

Considering the aforementioned steps in ML and the specific features of edge nodes, we can identify many challenges to be addressed in the edge intelligence paradigm, such as (i) running ML/DL on devices with limited resources, (ii) ensuring energy efficiency without compromising the inference accuracy; (iii) communication efficiency; (iv) ensuring data privacy and security in all steps; (v) handling failure in edge devices; and (vi) dealing with heterogeneity and low quality of data. In this paper, we present the results of a systematic literature review on current state-of-the-art techniques and strategies developed for distributed machine learning in edge computing. We applied a methodological process to compile a series of papers and discuss how they propose to deal with one or more of the aforementioned challenges.

I Innovations in Edge-Based Machine Learning

Edge-based machine learning is experiencing a surge in innovation, enabling powerful computational capabilities directly at the data source. Key developments include:

- Model Optimization Techniques:

Quantization: Reduces model precision to lower computational costs while maintaining accuracy.

Pruning: Removes unnecessary connections in neural networks, making models lighter and faster.

Knowledge Distillation: Transfers knowledge from large models to smaller ones for edge deployment.

These techniques make it possible to deploy efficient ML models like MobileNet and TinyML on resource-constrained edge devices.

- Federated Learning:

A decentralized training approach where models are trained across multiple devices using local data. This methodology ensures data privacy, minimizes the need for data transfer, and leverages the computational power of distributed devices.

- Edge AI Hardware:

Specialized hardware platforms such as NVIDIA Jetson, Google Coral, and Intel Movidius have been designed to support complex ML workloads at the edge. These devices provide enhanced processing capabilities and energy efficiency tailored for edge AI applications.

- Edge-Oriented Frameworks:

Frameworks like TensorFlow Lite, PyTorch Mobile, and ONNX Runtime have simplified the deployment of ML models on edge devices. These tools support streamlined workflows, from model optimization to deployment, enabling rapid prototyping and scaling.

- Real-Time Inference:

Advances in edge computing enable real-time inference, critical for applications such as predictive maintenance, autonomous navigation, and augmented reality. This capability minimizes the need for constant cloud connectivity, enhancing reliability and responsiveness.

II Examples and use cases for edge computing

In theory, data is gathered, filtered, processed, and analyzed "in-place" at or close to the network edge using edge computing techniques. It's a potent way to use data that can't be initially transferred to a single location, typically due to the sheer volume of data making such transfers prohibitively expensive, technologically unfeasible, or potentially violating regulatory requirements like data sovereignty. Numerous real-world examples and use cases have been generated by this definition:

1. Production. In order to identify production mistakes and enhance the quality of product manufacturing, an industrial manufacturer implemented edge computing to monitor

manufacturing. This allowed for real-time analytics and machine learning at the edge. Environmental sensors were added throughout the manufacturing facility with the help of edge computing, which gave information about how each product component is put together.

2. Farming. Think of a company that cultivates crops indoors without the use of pesticides, soil, or sunlight. Grow times are shortened by almost 60% thanks to the method. The company can monitor water consumption, nutrient density, and harvest efficiency by using sensors. To determine the effects of environmental elements, continuously enhance crop-growing algorithms, and guarantee that crops are harvested in optimal condition, data is gathered and evaluated.

3. Optimization of networks. By monitoring user performance across the internet and using analytics to identify the most dependable, low-latency network path for each user's data, edge computing can aid in network performance optimization. For the best time-sensitive traffic throughput, edge computing is essentially utilized to "steer" traffic across the network.

4. Safety at work. Businesses can monitor workplace conditions or make sure that workers adhere to established safety protocols by using edge computing to aggregate and analyze data from on-site cameras, employee safety devices, and other sensors. This is particularly useful in remote or exceptionally hazardous workplaces like oil rigs or construction sites.

5. Better medical care. The amount of patient data gathered from gadgets, sensors, and other medical equipment has significantly increased in the healthcare sector. In order for physicians to take prompt action to assist patients in avoiding health crises in real time, edge computing is necessary to apply automation and machine learning to access the data, disregard "normal" data, and identify problem data.

III Distributed Machine Learning

Generally, machine learning tasks can be classified into supervised, unsupervised, and reinforcement learning. Briefly, training data is labelled in supervised learning, in contrast to unsupervised learning which does not require any label. Differently, reinforcement learning is concerned with learning from feedback coming from external interactions. Machine learning algorithms are designed to run on powerful machines, which are often equipped with acceleration hardware such as GPUs and FPGA. However, nowadays due to the growing size of training data and machine learning models, learning on single machines cannot be done either efficiently or effectively due to limited hardware. Distributed computing can therefore help alleviate these problems. In distributed machine learning multiple workers cooperate and communicate with each other for training a model in parallel. In particular, it can be done with two different approaches: distributing the data or distributing the model. In the first approach data is partitioned on the worker nodes of the distributed system, which all execute the same algorithm on different partitions.

The models obtained by training the algorithm on the various partitions must then be aggregated. In the second approach, instead, the same data is processed by the worker nodes by executing different partitions of the model and the final model is therefore generated by the aggregation of all parts. This approach can be applied to all those machine learning algorithms in which parameters can be partitioned (e.g., neural networks).

IV Trends and Innovations in Distributed AI

Distributed AI is a paradigm where AI systems are decentralized, leveraging multiple devices, edge nodes, cloud servers, and other resources. This approach is increasingly vital for scaling AI applications, improving efficiency, and addressing data privacy concerns. Here are the latest trends and innovations in the field:

1. Federated Learning

Description: A collaborative machine learning approach where models are trained locally on edge devices or client nodes, with only model updates shared to a central server.

Innovations:

Cross-Device Federated Learning: Efficient training across billions of devices, with minimal impact on battery life and bandwidth.

Privacy-Enhancing Technologies: Integration of differential privacy and homomorphic encryption to protect sensitive data during the training process.

Applications: Healthcare (collaborative disease detection), mobile devices (personalized recommendations), and IoT (smart home optimization).

2. Edge AI

Description: Deploying AI models on edge devices for localized processing and decision-making.

V Introducing Distributed Artificial Intelligence Trends

Distributed Artificial Intelligence (AI) is a subfield of AI devoted to researching and developing distributed solutions. However, it has received much attention in recent times because of its decentralized and distributed nature.

To make Artificial Intelligence more reachable and scalable, it needs to be distributed and decentralized. Recently, distributed AI systems have collaborated with Machine learning and Deep Learning Technologies, giving them a new dimension to explore. Today, we will discuss the latest trends in Distributed Artificial intelligence.

- Particle Swarm Optimization

Inspiration of the technology: It is inspired by the social foraging behaviour of some organisms, such as birds' flocking behaviour and fishes' schooling behaviour. The way these organisms pass information while moving in the right directions in these swarms gives researchers opportunities to develop such systems artificially.

Methodology: The main objective of this algorithm is to use all agents to locate the optima in a multidimensional space. This optimum is initially assigned any random position and velocity in the space, but as time elapses with exploration and exploitation, the optima are found. For a greater understanding, see the image below.

Applications: The applications of Swarm Optimization are listed below.

Dimensionality reduction in machine learning uses swarm-based dimensionality reduction,

which uses the vectorized implementation of the PSO.

PSO has also been used as an optimisation technique to hyperparameter-tune deep learning algorithms.

- Ant Colony System

Inspiration of Technology: It is inspired by the communication of the ants, which is done by using a harmonic chemical known as a pheromone. The agent's probability of choosing the path is a function of the chemical intensity and the distance between the locations. This phenomenon of using pheromones for communication is one of the types of stigmergy. Researchers have developed a similar artificial structure for this phenomenon.

Methodology: This strategy aims to use historical information and construct the solution for the individual agent using a probabilistic step-wise approach. The probability of selecting any component for constructing a solution depends on that component's heuristic contribution to the overall cost function. Once the cost function is calculated, the history related to that path is also updated.

Applications: This Optimization technique is heavily used in a field called "Swarm Robotics," which studies how a large number of simple robotic agents can be developed such that the desired goal can be achieved by the collective behaviour of these agents using Ant colony optimization techniques.

2. Conclusion

The synergy between machine learning and edge computing is unlocking unprecedented opportunities for real-time, distributed AI applications. By addressing current challenges and leveraging ongoing innovations, organizations can harness the full potential of these technologies to drive efficiency, improve user experiences, and create intelligent systems that respond dynamically to their environments. As the field continues to mature, the collaboration between researchers, developers, and industry leaders will be pivotal in shaping the future of distributed AI. We have seen the latest trends in Distributed AI, though these trending technologies are in their infancy stage. At the pace with which work is going in these domains, technology will make distributed Artificial intelligence feasible and state-of-the-art solutions very shortly.

References

1. L. Lin, X. Liao, H. Jin, & P. Li, "Computation offloading toward edge computing." Proceedings of the IEEE, 107(8), 1584-1607,2019 [online].
2. M. Babar, & M. Sohail Khan, "ScalEdge: A framework for scalable edge computing in Internet of things–based smart systems." International Journal of Distributed Sensor Networks, 17(7),15501477211035332, 2021 [online].
3. R. T. Al-Zubi, N. Abedsalam, A. Atieh, & K. A. Darabkh, "LBCH: load balancing cluster head protocol for wireless sensor networks." Informatica, 29(4), 633-650, 2018 [online].
4. S. Maheshwari, D. Raychaudhuri, I. Seskar, F. & Bronzino,"Scalability and performance evaluation of edge cloud systems for latency constrained applications." In 2018 IEEE/ACM

Symposium on Edge Computing (SEC) (pp. 286-299). IEEE, (2018, October).
5. Mirza Golam Kibria Kien Nguyen, Gabriel Porto Villardi, Ou Zhao,Kentaro Ishizu and Fumihide Kojima, "Big Data Analytics - MachineLearning and Artificial Intelligence in Next Generation WirelessNetworks", IEEE, pp. 1-9, 2018.
6. Itamar Arel, Derek C. Rose, Thomas P. Karnowski, "Deep MachineLearning - A New Frontier in Artificial Intelligence Research", IEEEComputational Intelligence Magazine, pp. 13- 18, 2010.
7. Athmaja S, Hanumanthappa M, Vasantha Kavitha, "A survey ofmachine learning Algorithms for big data analytics", IEEE International Conference on Innovations in Information, Embedded and Communication Systems, pp.1-4, 2017.
8. Qiang Liu, Pan Li , Wentao Zhao, Wei Cai, Shui Yu, Victor C. M.Leung, "A Survey on security threats and defensive techniques of machine learning: A data driven view", IEEE Access, pp. 12103-12117, 2018.
9. Divyakant Agrawal, Sudipto Das, Amrel Abbadi, "Big Data and Cloud Computing: Current State and Future Opportunities", ACM - EDBT, pp. 530-533, 2011.
10. AV.Karthick, E.Ramaraj, R.Ganapathy, "An Efficient Multi Queue Job Scheduling for Cloud Computing", IEEE WCCCT, pp. 164 – 166, 2014.
11. AV. Karthick, E. Ramaraj, R. Kannan, "An efficient Tri Queue job Scheduling using dynamic quantum time for cloud environment", International Conference on Green Computing, Communication andConservation of Energy (ICGCE), pp. 871-876, 2013.