International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 06 June 2024, pp: 01-06 www.ijaem.net ISSN: 2395-5252

Deep Learning: A Tool for Distributed Communication Systems

Onyeyili Tochukwu .Innocent,Nwosu Ifeoma, AtokoloJoel, David Idakwo Friday
1, 2, 3, 4 Nnamdi Azikiwe University Awka

Date of Submission: 28-05-2024 Date of Acceptance: 06-06-2024

ABSTRACT
Distributed communication systems (DCS) involve the independent communication and operation of interconnected devices in order to enable faster computation and memory than standalone systems. These systems are crucial in industries like telecommunications and banking, where reliable data transmission is important. However, the lack of a central control unit in DCS makes them susceptible to security threats.This paper explored how Deep Learning (DL) can address these challenges and improve communication network performance. The techniques for training DL models and the programming frameworks needed for implementation are detailed with respect to their operation on DCS. Furthermore, the potential and limitations of DL in enhancing DCS performance are examined.

I. INTRODUCTION

Deep Learning (DL), a rapidly advancing area of Artificial Intelligence, has shown versatile application in various researches and has led to significant advancement in the field of computer vision, natural language processing, autonomous vehicles and speech recognition. The increased interest and advancement can be attributed to the availability of data and computing resource needed for the effective operation of various deep architectures (Alzubaidi et al, 2021). Since deep learning has proven a powerful tool in other fields, its potential to enhance the efficiency and reliability of distributed communication systems is now being explored.Distributed communication systems (DCS)involve the interconnection of groups of devices which are able to communicate and operate independently via Inter-process communication (Vatter et al, 2023). These devices need not necessarily be in the same location, but they are designed to work together for effective data transmission. DCS form a basis of the operation of various intelligent devices in today's world because it enables faster computation and memory than that offered by only one device (Tang et al, 2020). They are utilized in industries where efficient and reliable transmission is vital such as telecommunications, banking and air transportation. The lack of a central hub that oversees the interaction between devices means that this interconnectedness could be exploited maliciously, risking the network operation.

DL is important in communication systems due to its ability to improve the performance of network by the utilization of data-driven algorithms.This survey explores the various applications of DL as it pertains to communication networks. A review of recent works was carried out and the common DL algorithms used are detailed in Section II. Section III explores the techniques utilized for training DL models on distributed data. Section IV consists of an overview of the programming frameworks used in implementing DL in communication systems. Section V mentions the challenges in various DL implementations for distributed communication systems while a conclusion is reached in section VI.

II. LITERATURE REVIEW

The research in deep learning for distributed communication systems has been compared from various review articles. Liu B. et al (2021) presented the privacy concerns and solutions related to the use of machine learning tools in communication systems. They covered machine learning both as an adversary tool as well as a protection tool and protection target. The work by Aguzzi et al (2022) proposed a guideline for the implementation of machine learning in aggregate computing development. Likewise, Mayer and Jacobsen (2019) presented a survey on the various issues pertaining to the design of large distributed systems for DL. Kang et al (2020) proposed an improved parallel SVM model for distributed systems which significantly improves on the communication overhead problem while maintaining the model's classification accuracy.

For the IoT communication space, Williams et al (2022) and Murshed et al (2021) presented review articles which highlighted how effectively ML can be deployed in IoT while retain users' privacy.

A description of the various distributed machine learning and deep learning techniques was presented by Dehghani and Yazdanparast (2023). Their classification covered all aspects of distributed supervised and unsupervised learning algorithms as well as distributed reinforcement learning. The work by Kako (2021) reviewed distributed deep learning systems with a focus on classification based onoptimization and scheduling patterns. Langer et al (2020) investigated the implementation of distributed training of deep learning models. Niknam et al (2020) evaluated the various motivations and challenges of employing federated learning in wireless communications. Nama et al (2021) presented an overview of machine learning techniques for traffic scheduling in intelligent transportation systems. Their work highlighted the various strategies and challenges in their research area. A survey of various communication optimization in distributed deep learning was presented in the article by Shi et al (2020).

These researches utilized the computation of neural networks which are interconnected network of artificial nodes programmed to perform certain mathematical analysis on an input data in order to present or deduce conclusions at its output. The entry layer of the network is called the input layer, the data is then passed to subsequent layers known as hidden layers and finally a result is computed at the output layer. Deep neural networks with two or more hidden layers are commonly termed deep learning algorithmsbecause they model the human brain characteristic by learning from large amounts of data. These algorithms are able to determine which features are most important to distinguish one object from another in various classification tasks. The neural network algorithm is trained to perform its analysis based on computation methods like backpropagation, genetic algorithms (Verbraeken et al, 2020), equilibrium propagation (Scellier and Bengio, 2017) and the HSIC bottleneck (Ma et al., 2020). Backpropagation, the most commonly used method, involves the calculation of the loss gradient based on corresponding weights of the various layers. This gradient is then propagated backwards through the layers of the network, from the output to the input layers, for effective weight updates in order to minimize the error (Verbraeken et al, 2020).

Several DL algorithms have been proposed and evaluated based on their performance on benchmark object detection benchmark datasets. The commonly utilized DL architectures ofteninclude the following

Convolutional Neural Network (CNN): A feedforward neural network model that makes use of convolutional and pooling layers for efficient feature extraction of input data. The convolutional layers apply filters to the data while the pooling layers spatially reduces the filtered data output in order to reduce the number of layers and computation required. The final output layer is use to classify the input data into specified categories. The work by Wang T. (2021), CNN model was implemented in order to improve the communication efficiency of an edge-based federated learning network. The CNN model accelerated the convergence speed by 30.8%. Based on 6G technology, Mukherjee et al(2020) proposed a CNN for effective network optimization by ensuring proper resource allocation.

Autoencoder (AE): A type of neural network that is utilized for learning a compressed representation of a dataset. It is made of two layers trained to effectively encode an decode the data into an appropriate representation. This characteristic of AEs makes it useful anomaly detection and dimensionality reduction tasks as seen in the research by Lin et al (2020) which investigates the channel estimation for marine communications which present complex and rapidly-changing environment.

Generative Adversarial Network (GAN**):** These present a generative DL algorithm that is commonly utilized for data generation due to its capability to generate that is almost indistinguishable from real data. It consists of two neural networks – a generator that generates the "fake" data and a discriminator that differentiates with real data – to ensure the authenticity of its generated output. Zixu et al (2020) utilized a GAN model that generated realistic traffic data for network anomaly detection system. The performance of their model was evaluated using an AE-based algorithm on the UNSW Bot-IoT dataset.

Recurrent Neural Network (RNN): A DL architecture that is capable of processing of sequential data such as text and languages. Its architecture feature network synapses which give the network a memory structure that permits the knowledge of previous inputs in the network's internal state (Verbraeken et al, 2020). Long Short-term Memory (LSTM) network are a specialized form of RNNs that is used in the research byParra et al (2020)for a cloud-based detection of Botnet attacks in a network of IoT devices.

Deep Belief Network (DBN): DBNs are made up of multiple layers of restricted Boltzmann Machines (RBM) that work as an unsupervised DL model. This means that the model is able the learn the underlying distribution of the input data without requiring any prior labeling. In communication systems, DBNs are often used for signal demodulation (Lee-Leon et al, 2021) and channel estimation tasks.

III. DEEP LEARNING TECHNIQUES IN DCS

Conventional communication system design relied on the design of the channel model and was based on the establishment and optimization of the mathematical model of the system (Yu et al, 2022), but these models were often a far cry from reality. Recently, modern communication systems are modelled to be complex and have a lot of channels difficult to estimate, thus requiring the use of data-driven solutions that can be implemented and adjusted automatically based on the network parameters. The execution of DL on distributed systems, without collecting all the data on one server, is motivated by the privacy concerns as it pertains to centralized training approaches (Tang et al, 2020). For efficient data processing and faster computation, distributed deep learning is implemented by a process of parallelization. The parallelization methods can be majorly categorized into two: model and data parallelization.

Data parallelism works with the goal of increasing the throughput rate of the model by reproducing the model across the various nodes on the network. During its computation, each node copies the model and then computes its local gradients independently. At the end, their respective result is computed as an aggregate and the original model is updated (Jang, 2022). This method is extremely useful for the fast processing of large datasets located at various network points. Data parallelism research was spearheaded with the result of the DistBelief model by Chen and Lin (2014) which proposed a parallel distributed algorithm capable of spreading model partitions acrossdedicated server nodes. Another research by Nguyen et al (2021) reduced network contention by fixing the routing path length between computing nodes. Their work utilized a Distributed Loop network topology.Model parallelism enables DL training by splitting the model into partitions across the different nodes in the network. The individual nodes are responsible for the computation of the parameters in its model partition (Dehghani and Yazdanparast, 2023). The implementation of this parallelism strategy is difficult due to the complexity of its partitioning algorithms which is not scalable in practice (Jang, 2022). An application of this approach can be seen in the study by Moreno-Alvarez et al (2021) which presented a heterogenous model parallel DNN which ensures load balancing between uneven devices on a HPC platform. Another work by Li et al (2023) maximized the overlapping communication and computation operation to accelerate DL model training on multiple devices. Additional analysis on execution of parallelization is detailed in the work by Ben-Nun and Hoefler (2019).

IV. TOOLS FOR DISTRIBUTED DL IMPLEMENTATION

The implementation of a desired parallelization model for a distributed system can be achieved via any of the frameworks detailed in Table 3.1. These frameworks are open-source and designed in a way that a user can easily implement it into a desired structure.

Table 3.1:Commonly used programming frameworks and available cloud platforms for implementation of deep learning on distributed systems

Programming frameworks	**Cloud platforms**
TensorFlow	Amazon Sagemaker
PyTorch	Databricks
Horovod	Microsoft Azure
Microsoft Cognitive ToolKit	Google Cloud ML Services

Microsoft Cognitive Toolkit is an open-source API framework that can be utilized for creating Machine learning programs in C++ language. It was developed by Microsoft for their AI solutions like Cortana. TensorFlow, a commonly utilized API framework in DL research, is used in natural language processing tasks, speech recognition and computer vision. It makes use of both Python and C++ programming languages. Horovod is a specially designed framework for distributed deep learning based on TensorFlow and PyTorch platforms. It makes the process of parallelization across multiple GPUs much easier. Other similarly utilized frameworks for distributed deep learning include DeepLearning4j, MXNet by Apache, FfDL by IBM, and DeepSpeed by Microsoft and these can be implemented in a different languages like Python, JAVA, C++, R, Perl etc.

The execution of the DL program is typically performed on GPU or TPU devices capable of handling the model complexity. However, in their absence, several cloud-based platforms such as Databricks, Sagemaker, Azure and Google ML services have been established by leading companies like Google, Amazon and Microsoft. These services provide cloud solutions for evaluation of machine learning programs.

V. CHALLENGES IN DL IMPLEMENTATION FOR DISTRIBUTED SYSTEMS

Deep learning algorithms modelled on distributed systems are designed to handle large quantities of data because of the various access points of operation. The execution of these DL models, however, pose a number of challenges because of the complexity of the network. Ensuring the convergence of the model during distributed training is a major challenge because distributed systems may not always have access to all data at once and different nodes may have somewhat different models depending on their subset of data (Kaur, 2023). It poses a problem in terms of the computation overhead, computation ability and memory availability across the various devices.

Also, the processing ofa huge volume of data in distributed deep learning systems presents a communication problem due to the several instances of parallel training that is ongoing at each moment (Mayer et al, 2020). As a result, there exists a form of delayed communication among the network nodes.Ahmed et al (2021) experimented on how to avoid congestion in their network as much as possible. Another work by Kahira (2021) explored the utilization of a data and spatial parallelism of VGG16 in order to eliminate the memory issues of their model.

VI. CONCLUSION

This review paper examined the potential of DL in ensuring the efficiency and reliability of Distributed Communication Systems (DCS). Beyond its successes in other fields, DL offers promising solutions for tackling the unique challenges of networks of interconnected devices. Key takeaways include: the data-driven approach of DL algorithms that strengthens network performance optimizations and its use in reducing the security risks inherent in decentralized networks, the distributed data training techniques that enable efficient model training on the huge amount of data generated by DCS and the programming frameworks tailored for DL implementation in communication systems which facilitate the rapid development and deployment of intelligent network solutions. Although challenges remain, such as computational complexity and security concerns, the potential of DL for DCS is undeniable. As research and development continue, DL can be expected to play a pivotal role in shaping the future of interconnected, intelligent networks.

Future research directions such as exploring the integration of DL with other emerging technologies like edge computing (Liu S.et al, 2023) and blockchain (Wang S. et al, 2023) for enhanced security and scalability could be looked into. As with any rapidly evolving field, continuous research on improved models that leverage the application of DL on DCS would further enable seamless connection and communication in modern communication systems.

REFERENCES

[1]. Aguzzi, G., Casadei, R., & Viroli, M. (2022, July). Machine learning for aggregate computing: a research roadmap. In 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW) (pp. 119-124). IEEE.

[2]. Ahmed, A. S., Abood, M. S., & Hamdi, M. M. (2021, June). Advancement of deep learning in big data and distributed systems. In 2021 3rd international congress on human-computer interaction, optimization and robotic applications (HORA) (pp. 1-7). IEEE.

[3]. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, 1-74.

[4]. Ben-Nun, T., & Hoefler, T. (2019). Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. ACM Computing Surveys (CSUR), 52(4), 1-43.

[5]. Chen, X. W.& Lin, X. (2014). Big data deep learning: challenges and perspectives. IEEE access, 2, 514-525.
[6]. Dehghani, M & Yazdanparast, Z. (2023). From Distributed Machine Learning to Distributed Deep Learning: a comprehensive survey. Journal of Big Data. 10 (158). doi:10.1186/s40537-023-00829-x
[7]. Jang, I. (2022). Parallelism in Distributed Deep learning. https://insujang.github.io/2022-06-11/parallelism-in-distributed-deep-learning/
[8]. Kahira, A., N. (2021). Convergence of Deep Learning and High Performance Computing: Challenges and Solutions. Department of Computer Architectures (DAC), Universitat Politecnica De Catalunya, Barcelona. Published PHD Dissertation.
[9]. Kako, N. A. (2021). DDLS: Distributed Deep Learning Systems: A Review. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 7395-7407.
[10]. Kang, Z., Xiao, N., Chen, Z., Ou, Y., & Li, X. (2020, October). An Improved Parallel SVM Algorithm on Distributed System. In 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 204-209). IEEE.
[11]. Kaur, J. (2023, May). Distributed Machine Learning frameworks and its benefits. https://www.xenonstack.com/blog/distributed-ml-framework
[12]. Langer, M., He, Z., Rahayu, W., & Xue, Y. (2020). Distributed training of deep learning models: A taxonomic perspective. IEEE Transactions on Parallel and Distributed Systems, 31(12), 2802-2818.
[13]. Lee-Leon, A., Yuen, C., & Herremans, D. (2021). Underwater acoustic communication receiver using deep belief network. IEEE Transactions on Communications, 69(6), 3698-3708.
[14]. Li, S., Lai, Z., Hao, Y., Liu, W., Ge, K., Deng, X., ... & Lu, K. (2023). Automated Tensor Model Parallelism with Overlapped Communication for Efficient Foundation Model Training. arXiv preprint arXiv:2305.16121.
[15]. Lin, B., Wang, X., Yuan, W., & Wu, N. (2020). A novel OFDM autoencoder featuring CNN-based channel estimation for internet of vessels. IEEE Internet of Things Journal, 7(8), 7601-7611.
[16]. Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., & Lin, Z. (2021). When machine learning meets privacy: A survey and outlook. ACM Computing Surveys (CSUR), 54(2), 1-36.
[17]. Liu, S., Wang, X., Hui, L., & Wu, W. (2023). Blockchain-Based Decentralized Federated Learning Method in Edge Computing Environment. Applied Sciences, 13(3), 1677.
[18]. Ma, W. D. K., Lewis, J. P., & Kleijn, W. B. (2020, April). The HSIC bottleneck: Deep learning without back-propagation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 04, pp. 5085-5092).
[19]. Mayer, R., & Jacobsen, H. A. (2020). Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools. ACM Computing Surveys (CSUR), 53(1), 1-37.
[20]. Moreno-Alvarez, S., Haut, J. M., Paoletti, M. E., & Rico-Gallego, J. A. (2021). Heterogeneous model parallelism for deep neural networks. Neurocomputing, 441, 1-12.
[21]. Mukherjee, A., Goswami, P., Khan, M. A., Manman, L., Yang, L., & Pillai, P. (2020). Energy-efficient resource allocation strategy in massive IoT for industrial 6G applications. IEEE Internet of Things Journal, 8(7), 5194-5201.
[22]. Murshed, M. S., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., & Hussain, F. (2021). Machine learning at the network edge: A survey. ACM Computing Surveys (CSUR), 54(8), 1-37.
[23]. Nama, M., Nath, A., Bechra, N., Bhatia, J., Tanwar, S., Chaturvedi, M., & Sadoun, B. (2021). Machine learning-based traffic scheduling techniques for intelligent transportation system: Opportunities and challenges. International Journal of Communication Systems, 34(9), e4814.
[24]. Nguyen, T. T., & Wahib, M. (2021, May). An allreduce algorithm and network co-design for large-scale training of distributed deep learning. In 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid) (pp. 396-405). IEEE.
[25]. Niknam, S., Dhillon, H. S., & Reed, J. H. (2020). Federated learning for wireless communications: Motivation,

opportunities, and challenges. IEEE Communications Magazine, 58(6), 46-51.

[26]. Parra, G. D. L. T., Rad, P., Choo, K. K. R., & Beebe, N. (2020). Detecting Internet of Things attacks using distributed deep learning. Journal of Network and Computer Applications, 163, 102662.

[27]. Scellier, B., & Bengio, Y. (2017). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. Frontiers in computational neuroscience, 11, 24.

[28]. Shi, S., Tang, Z., Chu, X., Liu, C., Wang, W., & Li, B. (2020). A quantitative survey of communication optimizations in distributed deep learning. IEEE Network, 35(3), 230-237.

[29]. Tang, Z., Shi, S., Wang, W., Li, B., & Chu, X. (2020). Communication-efficient distributed deep learning: A comprehensive survey. arXiv preprint arXiv:2003.06307.

[30]. Vatter, J., Mayer, R., & Jacobsen, H.-A. (2023). The evolution of distributed systems for graph neural networks and their origin in graph processing and deep learning: a survey. arXiv preprint arXiv:2305,13854v1.

[31]. Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., & Rellermeyer, J. S. (2020). A survey on distributed machine learning. Acm computing surveys (csur), 53(2), 1-33.

[32]. Wang, S., Sheng, H., Zhang, Y., Yang, D., Shen, J., & Chen, R. (2023). Blockchain-empowered distributed multi-camera multi-target tracking in edge computing. IEEE Transactions on Industrial Informatics.

[33]. Wang, T., Liu, Y., Zheng, X., Dai, H. N., Jia, W., & Xie, M. (2021). Edge-based communication optimization for distributed federated learning. IEEE Transactions on Network Science and Engineering, 9(4), 2015-2024.

[34]. Williams, P., Dutta, I. K., Daoud, H., & Bayoumi, M. (2022). A survey on security in internet of things with a focus on the impact of emerging technologies. Internet of Things, 19, 100564.

[35]. Yu, W., Sohrabi, F., & Jiang, T. (2022). Role of Deep Leraning in Wireless Communications. ArXiv Preprint. arXiv:2210.02596v1 [cs.IT]

[36]. Zixu, T., Liyanage, K. S. K., & Gurusamy, M. (2020, December). Generative adversarial network and auto encoder-based anomaly detection in distributed IoT networks. In GLOBECOM 2020-2020 IEEE Global Communications Conference (pp. 1-7). IEEE.