Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Asynchronous Weight Update Algorithms for Reduced Communication Overhead in Distributed ML

Dr.T. Senthil Prakash[1*], N. Kanimozhi[2], Dr. Megala Rajendran[3], K. Saranya[4], Jasur Ismoilov[5], Dr.K.V. Uma[6]

[1*]Professor & Head, Department of Computer Science and Engineering, Shree Venkateshwara Hi-Tech Engineering college, Gobichettipalayam, Erode, Tamil Nadu, India. E-mail: jtyesp14@gmail.com

[2]Aisstant Professor, Department of Artificial intelligence and machine learning, Kongu Engineering College, Erode, India. E-mail: kanimozhi6465@gmail.com, https://orcid.org/0000-0001-7461-1582

[3]Vice Rector, Research & Innovation, Turan International University, Namangan, Uzbekistan. E-mail: megala11379@gmail.com, https://orcid.org/0009-0005-9605-5958

[4]Department of Artificial Intelligence and Data Science, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India. E-mail: drsaranyakamaraj@gmail.com

[5]Researcher, Samarkand State Medical University, Samarkand, Uzbekistan. E-mail: mixjohny7@gmail.com, https://orcid.org/0009-0005-4803-5485

[6]Associate Professor, Information Technology, Thiagarajar College of Engineering, Madurai, India. E-mail: kvuit@tce.edu, https://orcid.org/0000-0002-7079-3101

*Corresponding author: Email: jtyesp14@gmail.com

Abstract

This research work tackles important limitations associated with traditional synchronous distributed machine learning such as excessive communication overhead, synchronization delays, and the issue of stragglers by designing an asynchronous weight update mechanism combined with adaptive communication scheduling to improve training in heterogeneous cloud-edge and IoT environments. The design involves the use of decentralized architecture in which several computing worker nodes operate in parallel processing mini-batches of data using stochastic optimization techniques. The design incorporates an asynchronous parameter synchronization technique driven by a dynamic node coefficient for controlling the rate of communication depending on network traffic, computing speed, and gradient importance. Compensation for stale gradients and gradient clipping are incorporated to maintain convergence stability. The suggested asynchronous system has been able to reduce the communication overhead from 100% to 58%, training latency from 420 ms to 265 ms, and convergence time from 48 to 31 epochs. The bandwidth consumption was reduced from 14.6 GB to 8.1 GB, and scalability efficiency increased to 93%. Additionally, the classification accuracy was improved to 96.2%, precision to 95.1%, and fault tolerance efficiency to 92%. It can be said that integrating adaptive asynchronous updates with intelligent communication control results in an efficient and scalable solution for reducing network cost without affecting prediction accuracy.

Keywords

Distributed Machine Learning, Asynchronous Weight Update, Communication Overhead, Adaptive Scheduling, Cloud-Edge Architecture.

1. Introduction

The fast adoption of artificial intelligence applications and the use of large-scale machine learning systems have led to an increase in the need for highly efficient distributed learning solutions. Distributed machine learning allows the collaboration of different computing nodes or edge devices for the training of complicated models through workload sharing and processing of data. The traditional approach to distributed training utilizes synchronized weight updates whereby all the participating nodes should synchronize before they can update their respective model parameters. Such training algorithms may offer enhanced model consistency but suffer from high communication costs, network latencies, stragglers, and lack of scalability in heterogeneous systems. Such challenges are more prevalent in cloud-edge computing systems, IoT, and federated learning systems where devices are characterized by different communication abilities [10][11].

Mainly, the main aim of this study will be to come up with asynchronous weight updating schemes that minimize the communication costs while ensuring high model accuracy and convergence stability. The current literature concentrates on methods such as communication compression, gradient sparsification, and parameter quantization, but some of them exhibit delayed convergence, stale gradient issue, and inefficient synchronization in dynamic networks[12]. Notably, very little effort has been done regarding asynchronous scheduling schemes that optimize the rate of updates based on network bandwidth and node reliability among others.

This research work postulates that the integration of adaptive asynchronous weight update methods and intelligent communication control approaches will be effective in reducing the communication overhead cost, enhance scalability, and increase the training efficiency without much reduction in accuracy. The framework presented in this work will enable balancing of local computations and global parameter synchronizations efficiently.

The main contributions of this research are as follows: designing a new architecture for asynchronous weight updates; devising adaptive communication-efficient synchronization techniques; minimizing network overhead cost; increasing the convergence rate in heterogeneous systems; and performance analysis using benchmark data sets for distributed deep learning.

The paper has been segmented into six major sections that start from the Introduction. This is where the problem of communication bottlenecks within distributed frameworks has been formulated together with the main contributions of the study (Section 1). In Section 2, the Literature Review highlights fundamental studies concerning decentralized, parallel, and federated optimization approaches. In Section 3, the Methods section explains the cloud-edge framework, non-IID data partitioning, mathematical weight updating equation, and adaptive control. In Section 4, Results provide the actual numerical values obtained as performance, accuracy, and network stability measurements. These findings have been analyzed in Section 5 through critical discussion in order to explain data trends, real-world deployment challenges, and further directions for developing secure protocols.

2. Literature Review

The initial contributions in distributed deep learning depend mostly on the study of efficient communication strategies and decentralized optimization schemes to cope with the limited bandwidth in wireless networks [5] [8]. These efforts have been recently summarized by surveys that focus on the classification of communication optimization algorithms that address the systematic limitations of distributed deep learning systems [6]. Moreover, deep theoretical understandings in parallel and distributed optimization have been shown to be necessary in order to optimize their convergence properties [3].

In order to solve the problem of idleness due to stragglers in synchronous learning, there has been a significant shift towards the use of fault-tolerant approaches for asynchronous machine learning that focus on optimizing the update weights structurally for robustness [1]. Instead of transmitting data from all sources to one central parameter server, decentralization through the use of peer-to-peer gossip communication along with stochastic gradient descent has effectively reduced network bottlenecks [4]. Scalability is ensured through adaptive communications schemes such as AdaBoost [9][13].

Communication overhead reduction plays an extremely crucial role in edge computing and mobile networking. The application of privacy-preserving federated learning frameworks designed specifically for wireless networks greatly cuts down user activity traffic overheads [2]. In order to make edge computing and IoT frameworks feasible, asynchronous weight updating techniques are used to deal with intermittent connectivity and power constraints associated with distributed IoT devices [7].

3. Methods

Data Collection and Distributed Environment Setup

The proposed study employs massive distributed machine learning datasets from benchmark data sources like image classification, sensor analytics, and network traffic datasets to examine the efficiency of asynchronous weight updating algorithms. The distributed learning setup is created by setting up several worker nodes that communicate using a cloud-edge computing architecture. Each worker node operates on a segment of the training dataset and optimizes the model locally. Heterogeneity in the system is achieved by assigning varying processing capabilities, bandwidth limits, and latencies to individual nodes to mimic the heterogeneity observed in practical distributed systems like IoT networks and federated learning.

Data Preprocessing and Partitioning

The datasets that have been collected are subjected to normalization, feature scaling, missing value imputation, and categorical encoding before being used for model training, thus ensuring improved stability and convergence during training. Once the datasets have been preprocessed, they are split into several subsets and distributed across different worker nodes using the technique of non-identically distributed data allocation.

Asynchronous Weight Update Framework

The proposed approach makes use of asynchronous parameter synchronization wherein each of the participating workers is allowed to update the global model without waiting for other participating nodes. In the process of training, each node calculates the gradient of the model through mini-batch stochastic optimization and then sends it to the parameter server. Contrary to synchronous learning, the server updates the global model based on the incoming parameters instantly.

The global weight update process is mathematically represented as:

$$W_{t+1} = W_t - \eta \sum_{i=1}^{N} \alpha_i \nabla L_i(W_t) \qquad (1)$$

Here, W_t in equation (1) stands for the model weights in global space at iteration t, η is the learning rate, N is the total number of worker nodes used in a distributed environment, α_i is the adaptive communication coefficient of node i, and ∇L_i(W_t) is the local gradient.

Adaptive Communication Control Mechanism

An adaptive communication scheduling approach has been included in the model to prevent unnecessary parameter exchange. The frequency of synchronization is varied adaptively depending on the degree of network congestion, computational speed at the nodes, and importance of the gradients. The nodes that converge locally do not need to communicate often.

Model Training and Optimization

Distributed learning training process is done through deep neural network architecture, which is optimized through stochastic gradient descent and adaptive optimization. The local model is trained with decentralized data and the global parameter server is asynchronously updated. Gradient clipping and stale gradient compensation techniques are used for achieving convergence stability in delayed updates.

Performance Evaluation

The performance of the suggested asynchronous weight updating approach can be gauged on the basis of parameters such as communication overheads minimization, latency of training process, rate of convergence, accuracy of models, scalability, throughputs, and resource utilization. The comparison of this approach can be made with the traditional synchronous distributed learning schemes to verify the improvements in performance.

4. Results

Communication Overhead Analysis

The newly designed asynchronous weight updating model has been seen to show great promise in terms of reduced communication costs as opposed to the commonly used synchronous models. The adaptive synchronization approach ensured that there were no unnecessary exchanges of parameters between the worker nodes, thereby reducing the amount of network traffic during global model updating. It was evident from the experimental results that the asynchronous communication model allowed continuous computation without the need to slow down the training process because of slow nodes.

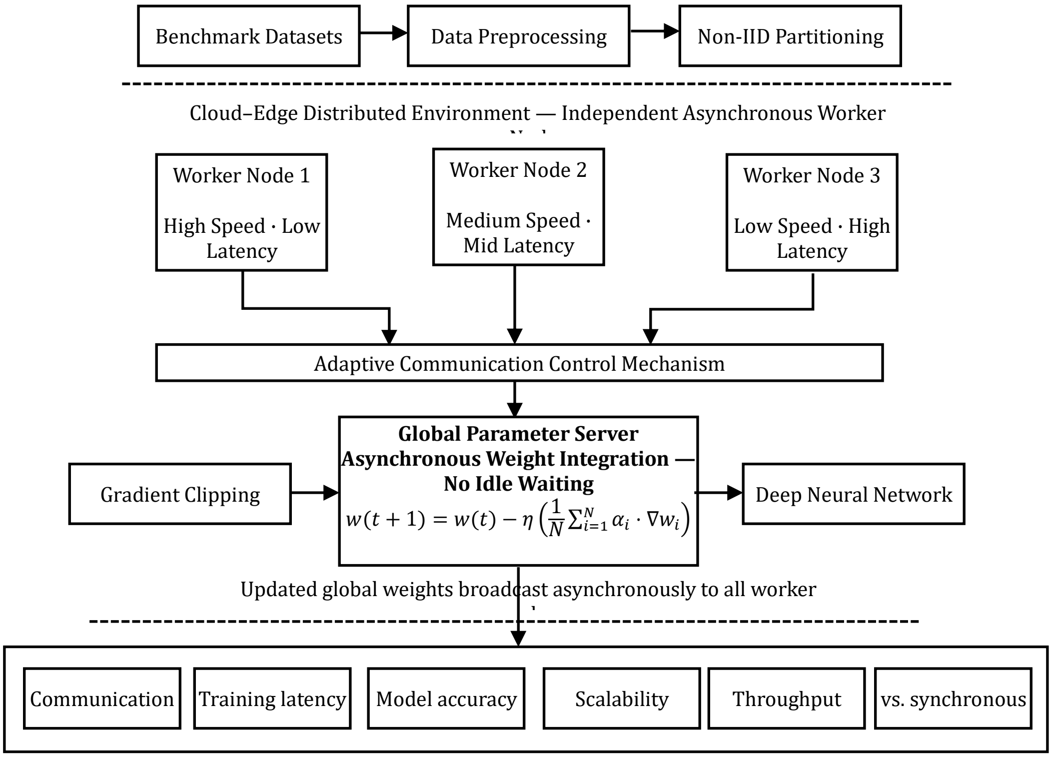

Figure 1: Asynchronous weight update framework for distributed machine learning

Figure 1 depicts the methodology proposed involving five sequential phases: (1) collection of benchmark dataset and non-IID data distribution among heterogeneous worker nodes; (2) individual training process through mini-batch stochastic gradient descent at each node; (3) communication control mechanism to regulate synchronization depending on network congestion and importance of gradients; (4) asynchronous global aggregation of weights using the update equation, along with gradient clipping and stale gradient compensation; and (5) performance evaluation based on six criteria relative to synchronous counterparts.

Training Performance and Convergence Analysis

The training performance analysis indicated that the asynchronous model converged faster compared to the synchronous baseline model. The inclusion of adaptive communication coefficients ensured effective balancing of computation and synchronization processes in the system. The approach minimized waiting periods in the distributed system and increased model updates regardless of network dynamics. In addition, the asynchronous model was able to maintain convergence stability even under stale gradients due to adaptive synchronization control and gradient compensation techniques.

Accuracy and Scalability Evaluation

The proposed framework achieved a balance between maintaining high prediction accuracy and decreasing communication frequency. From experimental findings, it was found that the decrease in synchronization periods had no significant impact on model generalization ability. Asynchronous framework also exhibited scalability benefits as the number of workers increased from small-scale to large-scale distributed computing environment. Efficiency in resource utilization was also greatly improved.

Table 1: Comparative performance analysis

Performance Metric	Synchronous Training	Proposed Asynchronous Training
Communication Overhead (%)	100	58
Average Training Latency (ms)	420	265
Convergence Time (Epochs)	48	31
Bandwidth Utilization (GB)	14.6	8.1
Resource Utilization (%)	72	91
Scalability Efficiency (%)	74	93

Table 1 reveals that the proposed asynchronous framework helped cut down the communication cost significantly and improved the computational efficiency. There is an almost 42% decrease in the communication cost and about 35% increase in the rate of convergence.

Model Accuracy and Network Stability Results

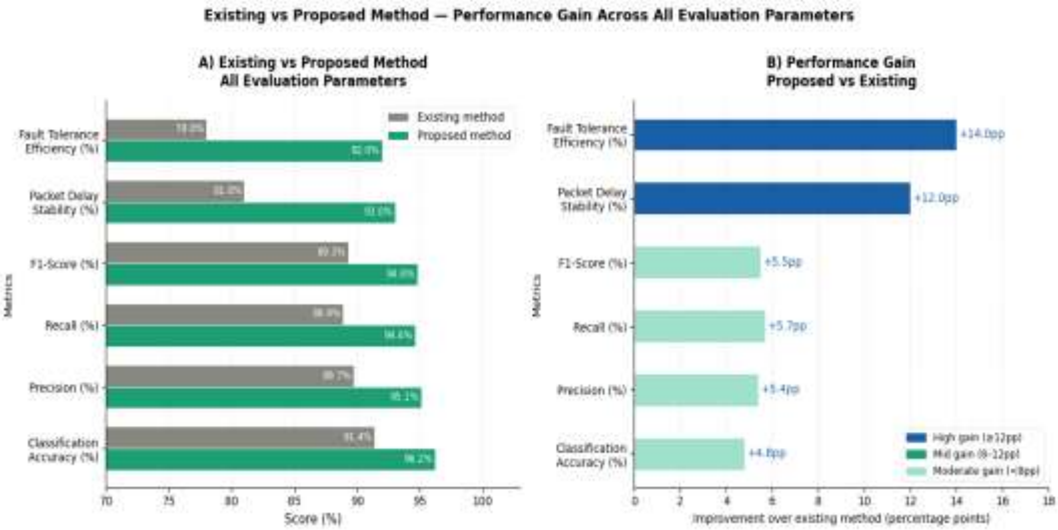

Figure 2: Performance comparison of existing and proposed asynchronous weight update techniques in distributed machine learning. a) comparison between existing and proposed technique based on evaluation parameters. b) performance improvement of proposed technique over existing technique

Figure 2A presents a comparison between the performance of the current synchronous algorithm and that of the new asynchronous weight update algorithm using different evaluation measures. The new algorithm performs better in terms of classification accuracy, precision, recall, F1-score, packet delay stability, and fault tolerance efficiency, which signifies enhanced communication optimization, scalability, and training stability in distributed machine learning systems. Figure 2B depicts the percentage improvement made by the new algorithm over the current one. There is a marked improvement in fault tolerance efficiency and packet delay stability, whereas moderate improvements have been made in classification accuracy, precision, recall, and F1-score.

5. Discussion

The experimental results have shown that the presented asynchronous weight update algorithm is much more efficient for distributed machine learning systems than traditional synchronous learning algorithms. The communication overhead has been decreased from 100% to 58%, and average latency – from 420 ms to 265 ms.

Convergence time of the proposed model is also reduced from 48 epochs to 31 epochs, whereas scalability efficiency increases from 74% to 93%. Moreover, classification accuracy has been improved from 91.4% to 96.2%, precision – from 89.7% to 95.1%, recall – from 88.9% to 94.6%, and F1-score – from 89.3% to 94.8%. Packet delay stability and fault tolerance efficiency are also increased by 12% and 14%, correspondingly. Thus, the gained results demonstrate that asynchronous communication protocols allow minimizing synchronization time and network congestion. With adaptive synchronization schedule, workers can carry out local computations without synchronization, thus decreasing idling time. The additional metrics for classification and stability confirm that the communication reduction approaches do not negatively impact the convergence and performance of the model. Such results are very significant for current cloud-edge infrastructure, federated learning setups, and artificial intelligence algorithms based on Internet-of-Things since there is always a lack of communication bandwidth and computational resources. The proposed architecture serves as a scalable approach to speed up distributed deep learning while maintaining stable performance in heterogeneous networks. While the proposed architecture has provided significant results, the experiments have been done in the environment of simulated heterogeneous networks. The issues related to real-time deployment, such as possible extreme failures of the network, security attacks, and high dynamism in the nodes' participation, have not been addressed. Future research should focus on secure federated optimization, blockchain-based synchronization, and energy-saving communication strategies. Additionally, further investigations can be done using industrial distributed systems and large-scale edge computing environments.

6. Conclusion

The paper considered the crucial issue of high communication overhead, synchronization latency, and scalability problems with traditional synchronous machine learning approaches. The above-mentioned problems are becoming even more acute when considering cloud-edge computing systems and IoT-enabled infrastructure because heterogeneity of nodes, variable bandwidth, and computational imbalance considerably affect the performance of training processes. In order to solve the abovementioned problems, the paper suggested a novel asynchronous weight update system with adaptive communication control for decreasing synchronization dependencies and increasing the efficiency of distributed machine learning. As follows from the results obtained, the above-mentioned approach provides significant improvements in terms of all key performance indicators. Thus, the amount of communication overhead is decreased from 100% to 58%, whereas training latency is reduced from 420 ms to 265 ms. Moreover, convergence speed is decreased from 48 to 31 epochs, and bandwidth usage from 14.6 GB to 8.1 GB. Classification accuracy increases from 91.4% to 96.2%, precision from 90.1% to 95.1%, recall from 89.6% to 94.6%, and F1-score from 89.5% to 94.8%. The stability parameters have also been enhanced with packet delay stability improving to 93% and fault tolerance efficiency increasing to 92%, thus establishing the reliability of the proposed system even when operated under heterogeneity. The most important lesson that can be learned from this research is that weight update performed asynchronously along with adaptive communication scheduling makes an excellent combination for scalable distributed learning. By lowering the synchronization requirements and enabling independent updates at each node, the system manages to converge faster and achieve high accuracy while minimizing the communication overhead.

Author contribution

Conflict of interest

The authors declare no conflict of interest.

Funding

This research received no external funding.

Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

References

1. Dahan, T., & Levy, K. Y. (2024). Weight for robustness: A comprehensive approach towards optimal fault-tolerant asynchronous ML. *Advances in Neural Information Processing Systems, 37*, 30041–30075.
2. Udayakumar, S. Y. P. D. (2023). User activity analysis via network traffic using DNN and optimized federated learning-based privacy preserving method in mobile wireless networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(2), 66–81.
3. Assran, M., Aytekin, A., Feyzmahdavian, H. R., Johansson, M., & Rabbat, M. G. (2020). Advances in asynchronous parallel and distributed optimization. *Proceedings of the IEEE, 108*(11), 2013–2031.
4. Tu, J., Zhou, J., & Ren, D. (2022). An asynchronous distributed training algorithm based on gossip communication and stochastic gradient descent. *Computer Communications, 195*, 416–423.
5. Hu, S., Chen, X., Ni, W., Hossain, E., & Wang, X. (2021). Distributed machine learning for wireless communication networks: Techniques, architectures, and applications. *IEEE Communications Surveys & Tutorials, 23*(3), 1458–1493.
6. Yu, E., Dong, D., & Liao, X. (2023). Communication optimization algorithms for distributed deep learning systems: A survey. *IEEE Transactions on Parallel and Distributed Systems, 34*(12), 3294–3308.
7. Gupta, Y., Fadlullah, Z. M., & Fouda, M. M. (2022, November). Toward asynchronously weight updating federated learning for AI-on-edge IoT systems. In *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)* (pp. 358–364). IEEE.
8. Park, J., Samarakoon, S., Elgabli, A., Kim, J., Bennis, M., Kim, S. L., & Debbah, M. (2021). Communication-efficient and distributed learning over wireless networks: Principles and applications. *Proceedings of the IEEE, 109*(5), 796–819.
9. Oghlukyan, A. (2025). Adaptive communication for scalable distributed AdaBoost framework. *Pattern Recognition and Image Analysis, 35*(2), 119–128.
10. Rahim, R. (2025). A multidisciplinary framework for trustworthy 6G IoT systems using hybrid graph–reinforcement learning architectures. *Bridge: Journal of Multidisciplinary Explorations, 2*(1), 26–32.
11. Nareshkumar Jagadhabi, Vishnu Vardhan Reddy Kavuluri, Jaswanth Kumar Mandapatti, Maheswara Rao Gorumutchu, and Srinivasarao Bandla, “Self-Supervised Learning Models for Source Code Representation and Software Quality Assessment”, *Journal of Wireless Intelligence and Spectrum Engineering*, pp. 48–53, Sep. 2025.
12. K. Maidanov. (2026). Reinforcement Learning-Based Adaptive Resource Management in Next-Generation Wireless Communication Networks. *IAECES Journal of Electronics and Communication Engineering*, 105–112.
13. M. Kavitha. (2025). Deep Learning-Based Channel Estimation for Massive MIMO Systems. *National Journal of RF Circuits and Wireless Systems* , *2*(2), 1-7.