Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/2025.TJ23594

Communication-Efficient Distributed Machine Learning: Techniques and Innovations

Juntao Wang

Faculty of Mathematics, University of Waterloo, Waterloo, Canada
j238wang@uwaterloo.ca

Abstract: As Artificial Intelligence (AI) technologies continue to advance, the size and complexity of machine learning models are rapidly increasing. Distributed Machine Learning (DML) has been proposed to improve the limitations of centralized training in terms of computational power and memory. However, communication overhead remains a significant obstacle in DML, restricting training efficiency. This paper proposes a hybrid approach combining Adaptive Gradient Compression (AGC) and Locally Updated Stochastic Gradient Descent (LU-SGD) to maintain model performance while reducing communication overhead. Specifically, the communication load is first reduced by compressing the gradient during each transmission round via AGC. Second, this study uses LU-SGD to minimize the number of communication phases by executing several local updates before synchronizing the gradients. Extensive experiments are conducted on Modified National Institute of Standards and Technology (MNIST), Canadian Institute for Advanced Research (CIFAR)-10, and ImageNet datasets with LeNet, Residual Neural Network (ResNet), and Visual Geometry Group (VGG). Experimental results show the hybrid approach reduces communication data while maintaining efficiency and model accuracy. This approach effectively optimizes communication and demonstrates its potential to improve distributed machine learning frameworks' expansion capability and efficiency.

Keywords: Distributed Machine Learning, Communication Overhead, Adaptive Gradient Compression, Stochastic Gradient Descent.

1. Introduction

As Artificial Intelligence (AI) technologies continue to advance, the size and complexity of machine learning models are rapidly increasing. Due to the limitations of centralized training in terms of arithmetic power and memory, Distributed Machine Learning (DML) has been proposed to improve computational efficiency by distributing the workload across multiple machines [1]. However, frequent communication between nodes has become the bottleneck, degrading the learning efficiency [2]. Consequently, it is imperative to investigate methods to minimize communication overhead. This paper is intended to discuss and compare a range of techniques systematically.

Several related techniques have been proposed and applied to enhance the communication efficiency of DML. Standard methods include gradient compression, Local Update Stochastic Gradient Descent (LU-SGD), federated learning, etc. Gradient compression techniques effectively reduce the communication overhead by reducing the amount of gradient data transmitted between nodes. For example, Chen et al. introduced a standard deviation-based gradient compression method,

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.TJ23594

which lowers communication overhead by setting a threshold based on the gradients' standard deviation during training [3]. Sun et al. proposed a sparse gradient compression method for distributed SGD that leverages momentum to lower communication costs without compromising accuracy [4]. LU-SGD allows nodes to perform multiple training rounds before synchronizing the communication. For example, cooperative SGD proposed by Joshi et al. designed new algorithms such as decentralized periodic averaging, which is faster than the baseline without compromise in accuracy, and averaging SGD with overlapped communication and computation [5]. The Federated Averaging (FedAvg) algorithm proposed by McMahan et al. achieves efficient model training and communication reduction [6]. Additionally, Sattler et al. proposed Sparse Binary Compression, enhancing compression efficiency by combining communication delay and gradient sparsification with a novel binarization strategy and an improved encoding method for weight updates [7]. These studies show that communication efficiency enhancement can significantly improve the overall performance of distributed training.

The primary purpose of this study is to investigate various techniques and innovations aimed at enhancing communication efficiency in distributed machine learning, while also analyzing the performance and applicability of each approach. Specifically, this paper emphasizes a hybrid method that combines Adaptive Gradient Compression (AGC) with LU-SGD. Experimental results demonstrate that combining AGC and LU-SGD significantly reduces communication overhead while preserving model accuracy, particularly in environments with many nodes and high communication costs.

2. Methodology

2.1. Dataset description and preprocessing

This study used three datasets: Modified National Institute of Standards and Technology (MNIST), Canadian Institute for Advanced Research (CIFAR)-10, and ImageNet. First, MNIST, a subset of the MNIST dataset, contains 60,000 examples of handwritten digits for a basic image classification task [8]. Second, CIFAR-10 contains 60,000 32*32 real-world images of different classes for medium classification tasks [9]. Finally, the ImageNet dataset contains more than 1 million high-resolution images covering over 1000 categories, which offers challenging image classification problems [10]. The preprocessing used in the experiments includes image normalization and graphical adjustments (rotation and/or cropping) to improve the model's generalization performance.

2.2. Proposed approach

The primary objective of this research is to enhance node communication efficiency in distributed machine learning, reduce communication overhead, and maintain model accuracy. The overall methodology is outlined in Figure 1, which encompasses data input, data preprocessing (including data augmentation and normalization), model training, and the communication optimization strategy. Initially, datasets such as MNIST, CIFAR-10, and ImageNet are input into LeNet, Residual Network (ResNet), and Visual Geometry Group (VGG) models, respectively, after undergoing normalization and enhancement. During the training phase, a hybrid approach combining AGC with LU-SGD is employed to optimize communication. This approach combines the strengths of both techniques to improve communication efficiency further. To assess the effectiveness of the proposed method, experimental comparisons are conducted, focusing on the impact of different configurations on both communication efficiency and model accuracy.

DOI: 10.54254/2755-2721/2025.TJ23594

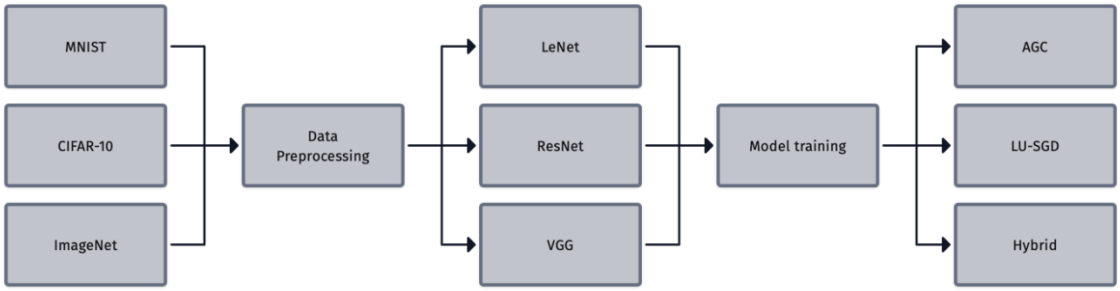

Figure 1: The pipeline of this research (picture credit: original)

2.2.1. LeNet

The initial model employed is LeNet, a classical convolutional neural network designed by Yann LeCun and colleagues. LeNet is widely used to recognize handwritten digits and is one of the crucial foundations of modern deep learning. The model has a hierarchical structure consisting of alternating fully-connected layers, convolutional layers, and pooling layers, which have the characteristics of local awareness, weight sharing, and dimension reduction, and can effectively extract local spatial features in images. LeNet has a simpler overall structure, fewer parameters, and a low consumption of computational resources, making it especially suitable for processing low-resolution photos. In this study, LeNet is used to train the MNIST dataset. Since the model is lightweight, the effectiveness of different techniques can be quickly verified.

2.2.2. ResNet

The second model is ResNet. ResNet, proposed by Microsoft Research, was developed specifically for image recognition. It introduces residual connection, stabilizing deep neural networks' training and convergence. When the depth of the network increases, the neural network is prone to gradient vanishing or performance degradation. Still, ResNet makes it easier for the network to learn the residuals through the “constant mapping”, to realize a deeper network structure design. ResNet-18 is a lightweight version of this series, consisting of several basic residual blocks, each containing two convolutional layers with jump connections, which can maintain high expressive power while considering computational efficiency. The thesis uses ResNet-18 to train on the CIFAR-10 dataset, balancing model accuracy and efficiency.

2.2.3. VGG

The last model is VGG, which the Visual Geometry Group of the University of Oxford proposed. The VGG network is primarily a deep convolutional neural network that increases layer depth without adding extra complexity to the model [11]. VGG increases the depth of the network by repetitively stacking 3×3 small convolutional kernel layers to enhance the feature expression capability of the model, while maintaining the receptive field. VGG19, one of the series's deeper structures, has good generalization performance and stability. Its standardized structure and simple design make it easy to deploy and optimize on various hardware platforms, so it is widely used in academic research and industrial practice. In this study, VGG19 is applied to the CIFAR-10 and ImageNet datasets to handle tasks with high image complexity.

DOI: 10.54254/2755-2721/2025.TJ23594

2.2.4. Loss function

The loss function is essential in the neural network training, evaluating the discrepancy between model predictions and actual values. In this study, cross-entropy loss, alternatively termed negative log likelihood loss, is employed and defined as follows:

$$H(p,q) = -\sum_i p(i)\, log\, q\,(i) \tag{1}$$

where p(i) being the actual distribution and q(i) standing for the model's estimated distribution. The log likelihood loss function improves the prediction probability of the correct category by penalizing the misclassified prediction results, thus improving the overall classification performance of the model. Due to its good mathematical properties, cross-entropy is widely used in neural network structures for various supervised learning tasks.

2.3. Implementation detail

A PC with an NVIDIA RTX 3080 GPU was used to conduct the experiments, and the deep learning framework used is PyTorch. Data preprocessing steps include random cropping, level flipping, and color adjustment. All models were trained with a batch size of 64 and an initial learning rate of 0.001. Training was set for 100 rounds, with early stopping applied to prevent overfitting. A multi-node environment is simulated in a distributed setup, and the communication frequency and compression rate are controlled to evaluate the effect of different optimization strategies.

3. Results and discussion

3.1. Result analysis

As shown in Figure 2, the loss trends of different methods during training are demonstrated. From Figure 2, Baseline, AGC, LU-SGD, and hybrid methods decrease rapidly within the first 10 epochs, reflecting good initial convergence. Baseline and AGC initially have slightly lower loss values, while LU-SGD and Hybrid start with slightly higher values. Still, all four methods tend to converge after 20 rounds and reach stability around 40 rounds. Overall, the final loss values of all four methods drop to close to 0, indicating that they all have strong fitting ability on the training set.

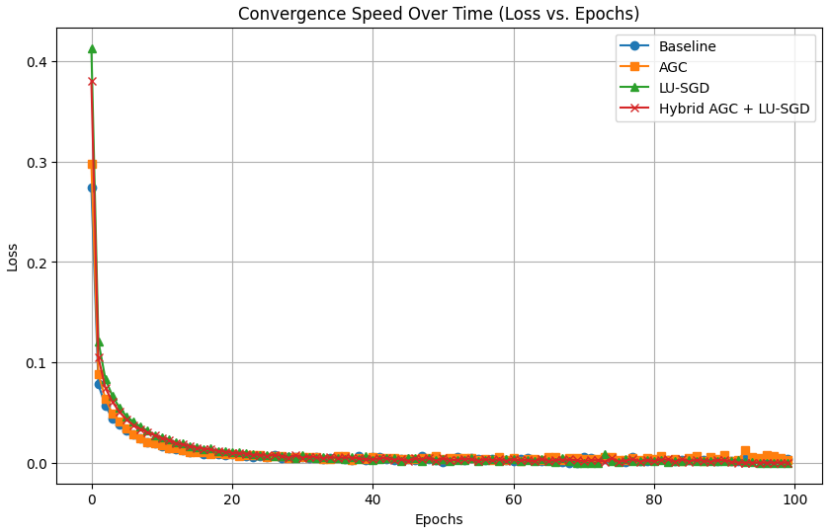

Figure 2: Convergence speed over time (loss vs. epochs) (picture credit: original)

DOI: 10.54254/2755-2721/2025.TJ23594

Figure 3 illustrates the variation of the accuracy of the different methods on the test set with the number of training rounds. It can be observed that the accuracies of all the methods increase rapidly and exceed 98% around the 10th round and show a slight fluctuation between 98.5% and 99.5% in the subsequent training. AGC and Hybrid methods perform slightly better in some rounds, which may be attributed to their compression or synchronization strategies that inhibit overfitting and enhance the generalization ability. The difference in the final accuracy of the four methods is minimal, and all of them are maintained at about 99%, indicating that under the same model structure and training data conditions, Baseline, AGC, LU-SGD, and Hybrid methods perform similarly in terms of model accuracy.

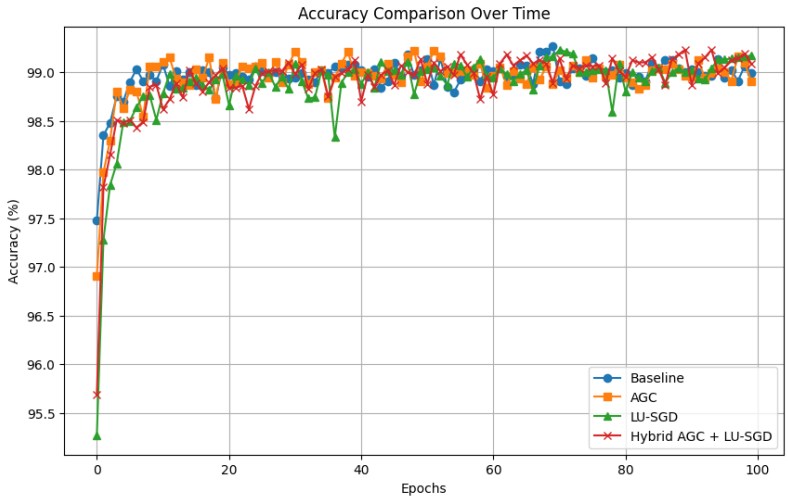

Figure 3: Accuracy comparison over time (picture credit: original)

As shown in Figure 4, the comparison results of the four methods in terms of communication efficiency are demonstrated. Communication efficiency is defined as the:

$$Communication\ Efficiency = \frac{Final\ Accuracy}{Number\ of\ Synchronication\ Step} \quad (2)$$

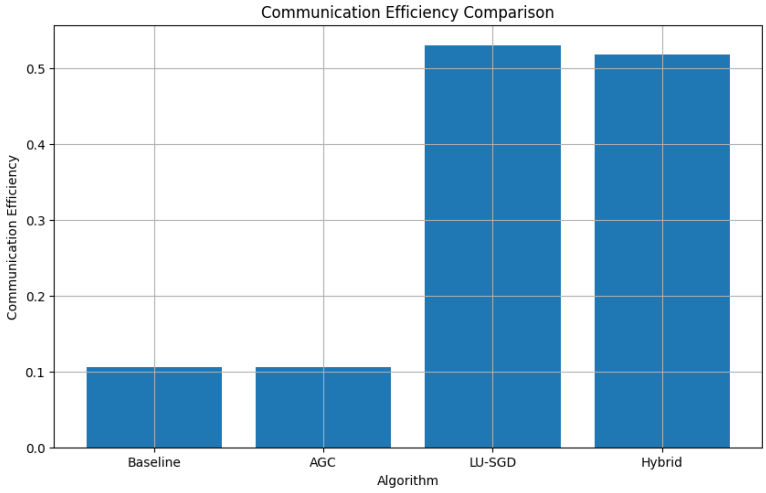

Figure 4: Comparison of communication efficiency (picture credit: original)

It is used to measure the performance gain per unit communication cost. It can be seen from Figure 4 that the communication efficiencies of baseline and AGC are close to each other, both maintained at about 0.11, indicating that the frequent gradient synchronization operations in each round of

DOI: 10.54254/2755-2721/2025.TJ23594

training bring a high communication overhead. In contrast, the LU-SGD and Hybrid methods show a significant advantage in terms of communication efficiency, reaching about 0.53 and 0.52, respectively. This improvement is mainly attributed to the local update strategy of LU-SGD, which reduces frequent parameter synchronization and drastically reduces the communication frequency while guaranteeing the model performance.

As shown in Figure 5, the amount of gradient data that needs to be transmitted in one communication is demonstrated for different methods during each gradient synchronization process. From Figure 5, the baseline and LU-SGD gradient size is about 0.17MB, which means they transmit almost the complete gradient information in the communication process. In contrast, the gradient size of AGC and hybrid methods is significantly reduced to only about 0.021MB, about 1/8 of the original gradient size. This result reflects the significant role of Adaptive Gradient Compression (AGC) in reducing the communication load. Although LU-SGD is optimized for communication frequency, it still needs to transmit the complete gradient each time it is synchronized, resulting in unimproved data volume for a single communication. In contrast, after incorporating AGC's compression strategy, the Hybrid method retains the synchronization frequency advantage of LU-SGD and significantly reduces the gradient size per communication.

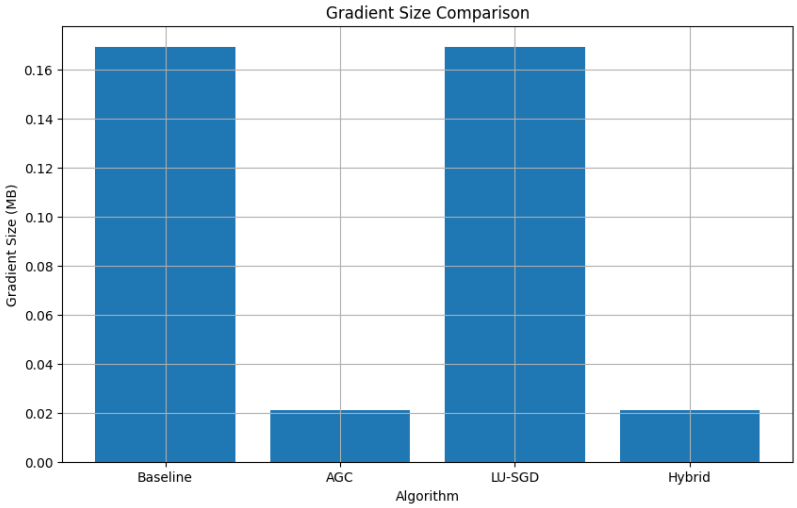

Figure 5: Gradient size comparison (picture credit: original)

3.2. Discussion

Through the comparison experiments, the hybrid method is the most balanced in comprehensive performance, combining the advantages of AGC and LU-SGD. Regarding communication efficiency, the hybrid method significantly reduces the synchronization frequency and gradient size, considerably reducing the overall communication overhead. Regarding model accuracy, the model's accuracy is on par with the baseline. It even has a slight advantage in some rounds, indicating that its compression and local updating have not compromised the model's convergence and generalization ability.

Specifically, AGC effectively reduces the amount of gradient data transmitted in each communication by an adaptive compression strategy. In contrast, LU-SGD reduces the synchronization cost by reducing the frequency of communication. The Hybrid method combines the two, reducing the volume of data in each synchronization and reducing the synchronization frequency, thus achieving the dual communication optimization goal of “compression + less synchronization”. The hybrid method maintains training speed and performance like other methods while effectively reducing the gradient transmission overhead and improving communication efficiency. Therefore,

DOI: 10.54254/2755-2721/2025.TJ23594

the proposed hybrid method is an effective strategy to trade off model performance and communication cost, and is especially suitable for distributed scenarios with limited bandwidth or collaborative computing across devices.

4. Conclusion

This study focuses on " Communication-Efficient Distributed Machine Learning", which aims to solve the problem of high communication overhead in distributed training. By analyzing the existing communication bottlenecks, the study concentrates on how to effectively improve the communication efficiency in distributed training to guarantee the model's performance, to promote the application of distributed deep learning in real-world scenarios. This paper proposes a hybrid method incorporating adaptive gradient compression and local update SGD to optimize communication efficiency. The method achieves the effect of “double compression” by reducing the communication frequency and compressing the gradient size in each round, which, together with the standard model structure and training process, significantly reduces the communication overhead while maintaining the model accuracy. Through experimental validation on multiple models and datasets, the results show that the proposed hybrid method significantly outperforms the traditional method in terms of communication efficiency and gradient transmission size, and at the same time, it is consistent with the baseline in terms of model convergence speed and final accuracy. The method demonstrates good generalization ability and has significant practical value. Future research will further consider applying the technique to distributed systems with heterogeneous devices or cross-geographic nodes, and introducing dynamic communication scheduling mechanisms to adapt optimization strategies under different network states. In addition, the communication optimization effect in larger datasets and multi-task learning scenarios will be explored to extend the adaptability and scalability of the method. Moreover, the technique's robustness against adversarial conditions will be investigated to ensure secure, fault-tolerant deployments in real-world environments.

References

[1] *Nassef, O., Sun, W., Purmehdi, H., et al. (2022). A survey: Distributed Machine Learning for 5G and beyond. Computer Networks, 207, 108820.*

[2] *Xie, C., Koyejo, O., Gupta, I. (2022). Zenops: A distributed learning system integrating communication efficiency and security. Algorithms, 15(7), 233.*

[3] *Luo, P., Yu, F. R., Chen, J., et al. (2021). A novel adaptive gradient compression scheme: Reducing the communication overhead for distributed deep learning in the Internet of Things. IEEE Internet of Things Journal, 8(14), 11476-11486.*

[4] *Tang, Y., Ramanathan, V., Zhang, J., et al. (2021). Communication-efficient distributed SGD with compressed sensing. IEEE Control Systems Letters, 6, 2054-2059.*

[5] *Wang, J., Joshi, G. (2021). Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms. Journal of Machine Learning Research, 22(213), 1-50.*

[6] *Sun, Y., Ochiai, H., Esaki, H. (2021). Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness. IEEE Transactions on Artificial Intelligence, 3(6), 963-972.*

[7] *Fei, J., Ho, C. Y., Sahu, A. N., et al. (2021). Efficient sparse collective communication and its application to accelerate distributed deep learning. Proceedings of the ACM SIGCOMM Conference, 676-691.*

[8] *LeCun, Y. (2013). The MNIST Database of handwritten digits. Retried from http://yann.lecun.com/exdb/mnist/*

[9] *Chen, L., Liang, X., Shen, C., et al. (2021). Synthetic CT generation from CBCT images via unsupervised deep learning[J]. Physics in Medicine & Biology, 66(11), 115019.*

[10] *Li, D., Ling, H., Kim, S. W., et al. (2022). Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 21330-21340.*

[11] *Thakur, N., Bhattacharjee, E., Jain, R., et al. (2024). Deep learning-based parking occupancy detection framework using ResNet and VGG-16. Multimedia Tools and Applications, 83(1), 1941-1964.*