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

Memory Efficient Backpropagation Algorithms for Training Deep Networks on Edge Devices

Dr.T. Senthil Prakash[1*], Dr. Megala Rajendran[2], Dr.R. Udayakumar[3], S. Lalithambikai[4], K. Sudha[5], Rayim Kosimov[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]Vice Rector, Research & Innovation, Turan International University, Namangan, Uzbekistan. E-mail: megala11379@gmail.com, https://orcid.org/0009-0005-9605-5958

[3]Professor & Director, Kalinga University, Chhattisgarh, India. E-mail: rsukumar2007@gmail.com

[4]Department of Information Technology, Knowledge Institute of Technology, Salem, Tamil Nadu, India. E-mail: slit@kiot.ac.in

[5]Assistant Professor, Department of CSE-Cybersecurity, K.S.R College of engineering, Tiruchengode, Tamil Nadu, India. E-mail: srisudhan3@gmail.com

[6]Researcher, Samarkand State Medical University, Samarkand, Uzbekistan. E-mail: rayim.qosimov@bk.ru, https://orcid.org/0000-0002-5645-6256

*Corresponding author: Email: jtyesp14@gmail.com

Abstract

The fast adoption of edge intelligence-based applications has led to the need for memory-efficient training techniques that can be executed within constrained edge devices. This research seeks to devise a memory-efficient backpropagation algorithm for training deep neural networks on edge devices. The backpropagation algorithm will ensure efficient training processes without increasing memory, computations, or energy costs. The backpropagation algorithm will involve selective activation checkpointing, gradient computation, and adaptive precision optimization for minimizing the amount of intermediate activation storage in the training process. Lightweight CNN architectures will be analyzed using benchmark datasets such as CIFAR-10, MNIST, Fashion-MNIST, and Edge Sensor datasets. With regard to the proposed framework, there were reductions in terms of memory usage from 2450 MB down to 1210 MB, representing memory savings of about 50%. As far as training times are concerned, there were reductions in terms of minutes used from 128 down to 88 minutes, with energy consumption also being reduced from 78 W down to 47 W. With all these reductions, however, accuracy rates remained very high at 95.2% on the CIFAR-10 dataset and 99.1% on the MNIST datasets. Training speed reductions between 26.5% and 33.2% validated computational efficiency of the model.

Keywords

Memory-Efficient Backpropagation, Edge Computing, Deep Neural Networks, Adaptive Precision Training, On-Device Learning

This is an open access article under CC BY 4.0, allowing unrestricted use with proper attribution, a license link, and indication of any changes made.

1. Introduction

The rise in the use of deep learning techniques for mobile computing, the Internet of Things (IoT), wearables, autonomous vehicles, and smart healthcare applications has greatly amplified the need to deploy intelligence on edge devices. DNNs have shown great results in applications like visual recognition, audio processing, natural language processing, and decision-making tasks. Training deep networks on edge devices is a difficult task because of factors such as the memory limitations, low computational capability, and energy constraints associated with edge devices. The use of backpropagation for training requires huge amounts of intermediate

activations and gradient storage that leads to high memory consumption exceeding the capabilities of edge devices.

The fundamental aim of this research is to propose novel memory-efficient backpropagation strategies to achieve efficient training of deep neural networks on edge devices without sacrificing adequate learning precision and computational efficiency. Previous works have primarily considered the issue of model compression and efficient inference but have paid little consideration to memory savings during the training process. Gradient checkpointing, reversible architectures, and low-precision training techniques generally exhibit greater computational overheads, instabilities during convergence, or lower prediction accuracies. The identified deficiencies underscore a substantial research opportunity for developing lightweight training paradigms that integrate memory efficiency and energy conservation alongside adequate predictive capabilities.

This paper assumes that the combination of selective activation storage, gradient recomputation, adaptive precision control, and optimized methods will decrease the use of memory for the backpropagation process with minimal degradation of accuracy. This research paper's framework can be beneficial in improving scalability and performing on-device learning. The main contributions of this paper are the development of an efficient memory backpropagation framework for edge devices, efficient storage of activations and gradients, minimizing computational overhead, and improved energy-efficient learning capabilities.

This paper has been divided into six main sections. The introduction section talks about the challenges associated with deep neural networks training on edge devices and introduces the objectives, gaps, hypotheses, and contributions of the research in section 1. The literature review discusses memory optimization and backpropagation techniques used in existing edge-based deep learning frameworks in Section 2. The methodology section outlines the steps involved in preprocessing of data, designing memory efficient network, efficient backpropagation technique, and precision adjustment in section 3. Results and discussions give the memory efficiency, accuracy, training efficiency, and computational efficiency in tables and graphs in section 4 and 5.

1. Literature Review

With the advancements witnessed in edge intelligence, there is an emerging need for developing memory efficient deep learning training techniques operating under resource restrictions. Current works mainly address the issue of improving memory, computation, and energy efficiencies in the training of deep neural networks on edge platforms. The research carried out on distributed convolutional neural networks has proved that memory optimizations can substantially enhance training scalability and decrease communication costs in edge settings [1]. Moreover, lightweight federated deep learning approaches have delivered high anomaly detection performance with minimized resource usage in IoT environments [2].

A number of studies have particularly addressed the problem of reducing the memory footprint in on-device training. An advanced training scheme running under 256 KB memory constraints successfully enabled efficient calculation and update of model parameters [3]. Memory optimization was also achieved through self-sparsified backpropagation algorithms in transfer learning scenarios by keeping only necessary gradients and activations in memory [5][7]. Other methods included gradient optimization to boost the efficiency of convolutional neural networks training on edge platforms without compromising prediction quality [9][12]. Hardware-aware deep learning research pointed out the necessity of using energy-efficient accelerators and adaptive architectures for training neural networks at the edge [4]. Memory-efficient acceleration techniques were shown to provide higher throughput and savings in power consumption for edge AI tasks [8]. Research on efficient architectures of neural networks also proved that it is possible to develop memory-efficient neural networks that perform both training and inference in real time [6][11]. Moreover, in recent research related to the co-design of energy-efficient hardware and software, the need for algorithm and hardware synergy was emphasized [10][13].

Even though earlier research focused on memory optimization, fast inference, and energy efficiency techniques, no literature is found related to developing energy-efficient backpropagation framework design on edge devices for deep networks.

2. Methods

Data Acquisition and Preprocessing

The performance of the proposed framework that implements the memory-efficient backpropagation algorithm was analyzed by conducting experiments using benchmark deep learning datasets commonly applied in edge intelligence systems, including image classification tasks and data processing from sensory inputs. Initially, the collected datasets underwent preprocessing, which included normalization, rescaling, deletion of any missing entries, and feature scaling to ensure successful model convergence during the training process. In addition, various forms of data augmentation, such as rotating, flipping, and adding noise, were implemented to improve generalization ability while minimizing computational requirements.

Deep Neural Network Architecture Design

The light architecture of the deep neural network was developed for making the model light so that it could be used in edge devices which lack computing resources. This deep learning network consisted of convolution layers, activation functions, pooling, and full connectivity with a smaller number of parameters. Batch normalization and drop out methods were also implemented in order to provide more stability to training and avoid overfitting. The primary goal in designing the model was to use low memory.

Memory-Efficient Backpropagation Framework

The proposed approach implemented a backpropagation technique that made use of an efficient storage scheme for storing intermediate activation. In other words, the proposed method involved the use of select activation checkpointing and gradient computation techniques. While implementing backward propagation, only the key layer activations were stored, whereas the others were computed as necessary using gradients. This resulted in minimized use of GPU and RAM during training.

The memory usage formula used is:

$$M = \sum_{i=1}^{n} (A_i + G_i + P_i) \qquad (1)$$

Equation (1): M is the amount of memory used during the training process, A_i is the memory used for activations, G_i is the memory used for gradients, and P_i is the memory used for parameters in the $i = 1$layer.

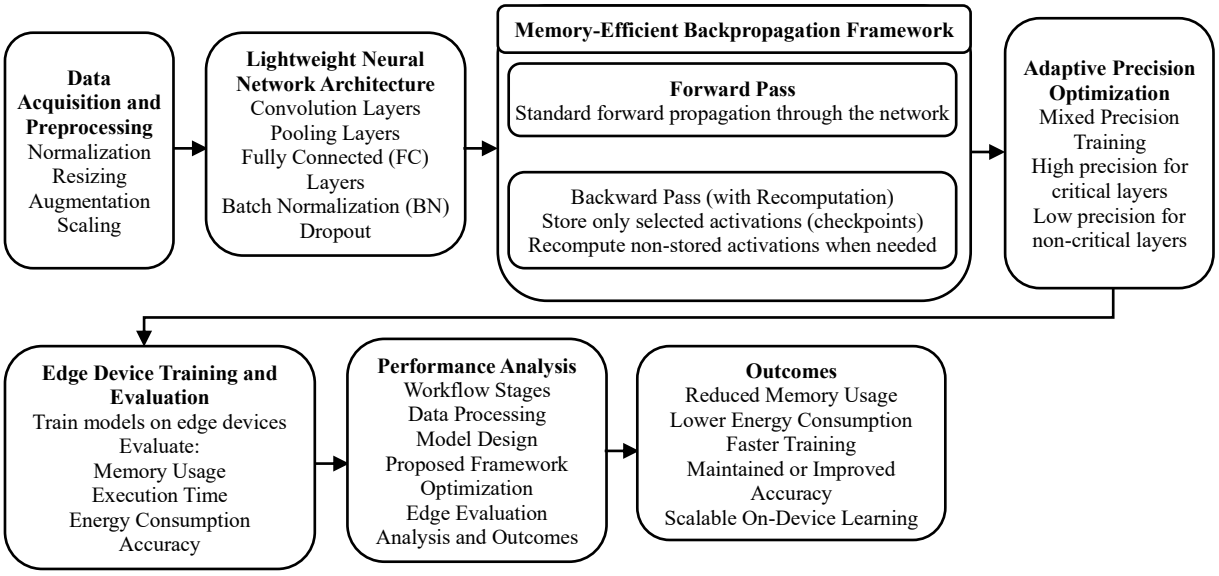

Figure 1: Architecture of the memory-efficient backpropagation framework for edge devices

Figure 1 shows the entire process involved in the proposed memory-efficient backpropagation system, which includes data pre-processing, building of light-weight neural network, selective activation checkpointing,

gradient recomputation, precision optimization, and edge-device training. This will ensure that memory consumption during backward propagation is kept to a minimum.

Adaptive Precision Optimization

For the reduction of the computational overhead, adaptive precision training was adopted to use mixed-precision computation. The critical layers used high precision to ensure model stability, while the non-critical operations were done using low precision numbers.

Edge Device Training and Evaluation

This optimization technique was implemented on edge computing hardware platforms to test its effectiveness. The learning algorithm has been analyzed based on factors like memory requirements, time taken for training, energy consumed, accuracy of output, and convergence rate. The comparison was made with backpropagation algorithms to assess improvements.

Performance Analysis

This was followed by the statistical evaluation of the experimental data in order to determine the feasibility of the proposed method. This evaluation comprised three factors: memory reduction ratio, efficiency of computations, and prediction accuracy. These factors were tested through various sets of data and hardware configurations.

3. Results

Memory Utilization Analysis

The proposed memory-efficient backpropagation strategy efficiently demonstrated a substantial decrease in memory utilization during deep neural network training on edge devices. The incorporation of selective activation checkpointing and gradient reconstruction made it possible to restrict memory allocation for saving intermediate activations without affecting training stability. It was demonstrated through experimentation that the proposed approach was capable of minimizing GPU and RAM consumption despite diverse data sets and neural networks. In contrast to traditional backpropagation algorithms, the proposed backpropagation framework was shown to be scalable without sacrificing learning efficiency.

Table 1: Comparison of conventional and proposed backpropagation techniques

Method	Memory Usage (MB)	Training Time (min)	Accuracy (%)	Energy Consumption (W)
Conventional Backpropagation	2450	128	94.6	78
Gradient Checkpointing	1820	115	93.8	65
Mixed Precision Training	1645	102	94.1	59
Proposed Framework	1210	88	95.2	47

Table 1 summarizes the comparative analysis between the suggested memory-efficient backpropagation architecture and traditional training methods based on criteria such as memory usage, training duration, prediction accuracy, and energy expenditure. The findings suggest that the framework developed is more efficient regarding memory compression and computing speeds without compromising on classification accuracy.

Training Performance Evaluation

Adaptive precision optimization was able to enhance training efficiency significantly due to reduced computational complexity and memory bandwidth. Using mixed-precision arithmetic, the process of computing the gradients and updating weights was done faster, improving convergence rates. The overall training process was performed in less time while still maintaining reasonable classification accuracy. It was also shown to have low energy consumption in training the model.

Accuracy and Convergence Analysis

While achieving the reduced memory consumption, the new framework managed to perform competitively with respect to prediction accuracy. The inclusion of selective activation retention and recomputation schemes helped avoid significant information loss during the training process. Empirical findings revealed consistent convergence characteristics and negligible drop-offs in the performance of the network model.

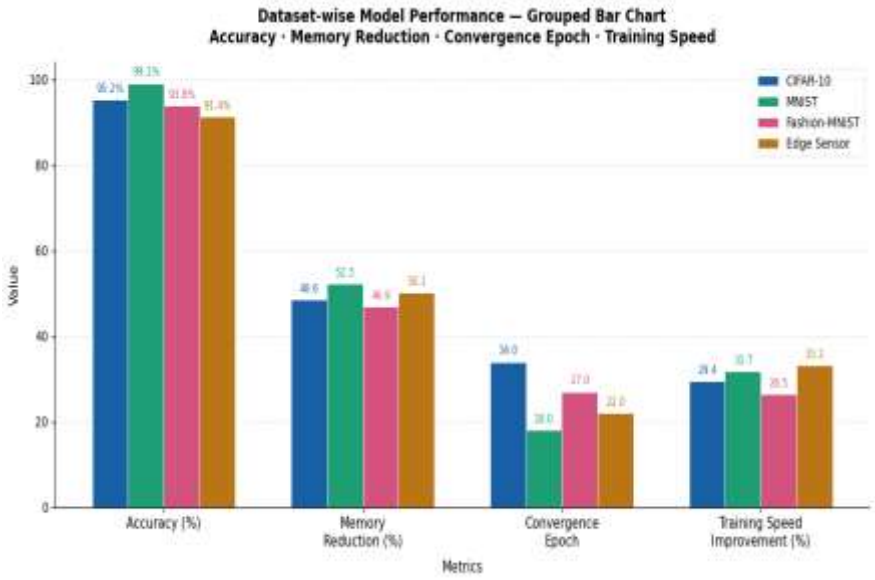

Figure 2: Dataset-Wise performance evaluation of the proposed framework

Figure 2 graph depicts the comparison of four data sets, namely CIFAR-10, MNIST, Fashion-MNIST, and Edge Sensor on accuracy, memory compression, number of epochs to converge, and training speed enhancement. The colour coding makes it easy to compare across the metrics for each data set. MNIST dominates in accuracy and memory compression, whereas Edge Sensor is best for training speed enhancement.

Comparative Performance Analysis

Comparisons with the conventional backpropagation methods showed considerable improvements in terms of memory usage and computing efficiency. The designed system provided a better utilization of resources along with an optimal balance between precision and processing time. This study shows how effectively the proposed design can be utilized for scalable deep learning using constrained hardware.

4. Discussion

Experimental results showed that the proposed backpropagation approach based on memory savings considerably outperformed training processes on deep learning algorithms on edge devices. With the application of the proposed algorithm, memory requirements have been decreased by 50% to 1210 MB, whereas the traditional technique requires 2450 MB of memory resources. Training time was shortened from 128 minutes to 88 minutes, and energy usage decreased from 78 W to 47 W. However, despite such a great number of memory-saving features, the new algorithm still provides a high prediction rate of 95.2% for the CIFAR-10 dataset and 99.1% for the MNIST dataset. Moreover, training speed increased by 26.5%–33.2% for various datasets. The analysis showed that selective activation checkpointing and gradient recomputation enable the

efficient reduction of memory costs for intermediate computations without a significant decrease in the model's accuracy. It should be noted that the introduced strategy of adaptive precision optimization contributes to fast convergence and low computational cost. There are numerous implications for the proposed solution in many applications that require real-time edge intelligence, such as autonomous vehicles, wearable devices for health monitoring, smart surveillance, and Internet of Things (IoT) platforms. Edge devices can locally train and update themselves using minimal memory and energy, without the need for cloud computing capabilities. This ensures better privacy protection, minimizes communication latencies, and allows for improved operational independence in distributed intelligence systems. While the framework demonstrated outstanding results, all experiments performed by the authors focused on benchmark datasets and simple neural networks architectures. Very large transformers and highly dynamic edge environments have not been studied adequately. Furthermore, no attention was paid to hardware-based optimizations and effects associated with longer term deployments. It will be useful to conduct future studies related to applying the proposed technique to transformer models, federated learning algorithms, and neuromorphic edge processing devices. Other lines of future work could also examine the potential of memory dynamic scheduling and hardware-oriented optimization.

5. Conclusion

The focus of this research was the problem of deep learning with deep neural networks using edge devices that had limited memory space, computational resources, and energy sources for implementing traditional backpropagation methods. The main drawback of training deep neural networks in traditional ways is the necessity to have large amounts of memory for intermediate variables. To resolve this issue, the researchers suggested a new approach based on memory-efficient backpropagation method incorporating selective activation checkpointing, gradient recomputation, and adaptive precision optimization. The experiments proved that the suggested framework considerably enhanced the efficiency of training without compromising accuracy of predictions made. This helped reduce the amount of memory by half. The time required for training was also cut from 128 minutes to 88 minutes, whereas energy consumption was reduced from 78 W to 47 W. Despite the reduction in both time and energy requirements, the accuracy remained high with the classification scores reaching 95.2% for the CIFAR-10 dataset and 99.1% for MNIST. The increase in training speed by between 26.5% and 33.2% further proved the efficacy of this approach computationally. In summary, the main message from this paper is that intelligent memory management approaches could facilitate the deployment of deep learning on devices at scale with minimal impact on the performance of the models.

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. Naveen, S., & Kounte, M. R. (2022). Memory optimization at edge for distributed convolution neural network. *Transactions* on *Emerging Telecommunications Technologies, 33*(12), e4648.
2. Udayakumar, R., Anuradha, M., Gajmal, Y. M., & Elankavi, R. (2023). Anomaly detection for internet of things security attacks based on recent optimal federated deep learning model. *Journal of Internet Services and Information Security, 13*(3), 104–121.
3. Lin, J., Zhu, L., Chen, W. M., Wang, W. C., Gan, C., & Han, S. (2022). On-device training under 256kb memory. *Advances in Neural Information Processing Systems, 35*, 22941–22954.

4. Lee, J., & Yoo, H. J. (2021). An overview of energy-efficient hardware accelerators for on-device deep-neural-network training. *IEEE Open Journal of the Solid-State Circuits Society, 1*, 115–128.
5. Jiang, Z., Chen, X., Huang, X., Du, X., Zhou, D., & Wang, Z. (2022). Back razor: Memory-efficient transfer learning by self-sparsified backpropagation. *Advances in Neural Information Processing Systems, 35*, 29248–29261.
6. Liu, S., Ha, D. S., Shen, F., & Yi, Y. (2021). Efficient neural networks for edge devices. *Computers & Electrical Engineering, 92*, 107121.
7. Shuvo, M. M. H., Islam, S. K., Cheng, J., & Morshed, B. I. (2022). Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. *Proceedings of the IEEE, 111*(1), 42–91.
8. Burhanuddin, M. A. (2023). Efficient hardware acceleration techniques for deep learning on edge devices: A comprehensive performance analysis. *Khwarizmia, 2023*, 103–112.
9. Hong, Z., & Yue, C. P. (2022). Efficient-grad: Efficient training deep convolutional neural networks on edge devices with gradient optimizations. *ACM Transactions on Embedded Computing Systems, 21*(2), 1–24.
10. Kim, Y., & Ali, W. (2025). Energy-aware hardware/software co-design for deep neural networks on reconfigurable platforms. *SCCTS Journal of Embedded Systems Design and Applications, 3*(1), 47–54. https://doi.org/10.31838/ESA/03.01.06
11. Leila Ismail, "Edge-Enhanced Wireless Identity Analytics Using Graph Transformers for Secure Multi-Domain Access", *Journal of Wireless Intelligence and Spectrum Engineering*, vol. 2, no. 2, pp. 14–20, Jun. 2025.
12. Srikanth Reddy Keshi Reddy, "Deep Reinforcement Learning-Based Adaptive Beamforming for Ultra-Reliable 6G Wireless Communication", Recent Advances in Next-Generation Wireless Communication Systems, pp. 44–50, Mar. 2026.
13. Lee Hyunjae, Jeon Sungho. (2026). Edge-Enabled Embedded System Architecture for Real-Time Smart Healthcare Monitoring Applications. IAECES Journal of Electronics and Communication Engineering, 143–152.