Enhancing Edge Intelligence with Layer-wise Adaptive Precision and Randomized PCA

Sambit Kumar Mishra, Velankani Joise Divya G C, Paavani Aashika Maddi, NVSS Mounika Tanniru, Sri Lakshmi Praghna Manthena

SRM University-AP, India

Email: skmishra.nitrkl@gmail.com, (joisedivya_velankani, paavaniaashika_maddi, mounika_tanniru, srilakshmipraghna_m) @srmap.edu.in

***Abstract* - Edge intelligence is the ability of edge devices to carry out intelligent operations, such as object identification, speech recognition, or natural language processing, utilizing machine learning algorithms. The primary goal is to fix edge computing's problems and improve its performance. The main goal of this work is to apply RPCA to increase energy efficiency and reduce memory usage. The algorithm computes the covariance matrix of the centered data, finds the eigenvectors and eigenvalues of the covariance matrix, sorts the eigenvectors and eigenvalues in descending order of the eigenvalues, chooses the first set of eigenvectors, and projects the data onto the chosen eigenvectors. This article employs a technique known as layer-wise adaptive precision (LAP), which decreases the precision of activations in neural network layers that contribute less to output accuracy.**

***Index Terms* - Randomized principal component analysis (RPCA); Dimensionality Reduction; Edge- intelligence; layer-wise adaptive precision (LAP);**

I. INTRODUCTION

In recent years, edge computing has emerged as a groundbreaking solution, addressing the evolving needs of diverse applications such as healthcare monitoring, swarm intelligence, and medical diagnostics. It offers a compelling alternative to conventional cloud-centric architectures by empowering Internet of Things (IoT) devices to process data locally. This paradigm shift has significant implications, that include assuring data decentralization, and minimizing communication overhead and latency, all while enhancing the efficiency of edge devices.

However, the adoption of edge computing is challenging, particularly when dealing with devices characterized by limited computational resources. These devices, often called "edge devices," play a pivotal role in the network's periphery, closer to where data is generated and farther from centralized cloud data centers. To overcome these resource constraints, the emergence of more capable counterparts known as "edge servers" with enhanced computing capabilities effectively harnessing the full potential of edge computing.

One of the driving forces behind the surge in data generated at the edge is the proliferation of connected devices. This data is often complex, noisy, unstructured, and heterogeneous, posing a significant challenge for traditional machine learning methods. Deep learning has gained prominence in this context, offering superior capabilities for processing vast and intricate datasets. However, the computational demands of deep learning models make their deployment on IoT devices with limited memory and processing power daunting. This underscores the urgent need to develop efficient deep-learning models that can operate effectively within the resource constraints of edge devices.

This introduction sets the stage for the introduction of Hybrid-Net, an innovative network topology designed to tackle the resource limitations of edge devices. Hybrid-Net introduces a novel approach by combining binary and high-precision inputs and activations at various network levels, offering a mixed-precision solution that optimizes memory usage and computational resources. This approach is crucial in addressing the challenges faced by edge devices, which often grapple with low power, limited computation resources, and constrained storage capacity. In some instances, these devices are also bound by stringent power consumption and data processing constraints.

Furthermore, the need for dimensionality reduction becomes evident as the volume of data at the edge grows exponentially. Reducing computation complexity and improving model performance are essential objectives in machine learning and data analysis. While various methods exist for dimensionality reduction, Principal Component Analysis (PCA) is a prominent technique. PCA enables data visualization, data compression, and feature extraction, all of which are essential in optimizing models for edge computing.

This research project aims to go beyond the traditional PCA approach by implementing randomized PCA for dimensionality reduction. Additionally, it incorporates automatic threshold selection and regularization of weight decay (with a weight decay value of 0.001). The project also emphasizes the thorough implementation of the hybrid-precision training technique and incorporates PCA for layer selection and hybrid-precision quantization. These efforts collectively aim to enhance the efficiency and effectiveness of edge computing, addressing critical concerns such as latency, energy consumption, security, privacy, and scalability.

In the following sections, innovative approaches proposed in the literature are stated in the Related Work Section. After that, how the authors contribute to the advancement of edge computing and its applications in diverse domains are indicated in Sections III and IV. Sections V and VI present the simulation results and the significant layer role. The result analysis is elaborated in Section VII, followed by Conclusion Section.

II. RELATED WORK

In the realm of edge computing and the optimization of machine learning models for resource-constrained environments, randomized Principal Component Analysis (PCA) emerges as a pivotal technique. Traditional PCA and its variants have long been applied for dimensionality reduction in various domains. However, the ever-growing demands of edge intelligence necessitate novel approaches to processing data while preserving critical information efficiently. Developing energy-efficient mixed-precision neural network systems using principal component analysis (PCA) is important in edge intelligence. While their work provided valuable insights into energy-efficient models, our approach focuses on layer-wise adaptive precision and randomized PCA, offering more granular control of precision for edge intelligence tasks [1].

Randomized PCA, characterized by its computational efficiency and suitability for large datasets, has garnered significant attention. Nevertheless, applying randomized PCA to enhance the performance of edge-based machine learning models, particularly in scenarios where computational resources are limited, presents a unique set of challenges and opportunities. To contextualize this research, the landscape of randomized PCA and related precision control techniques sheds light on their applications in edge computing. The concept of edge intelligence within IoT architectures has been addressed by various researchers [2] and laid the foundation for understanding IoT architectures. Additionally, we delve into previous endeavors to optimize neural network architectures for edge devices, as these insights align with the goals of the proposed Hybrid-Net design. The emphasis is on the importance of efficient data processing at the edge [3], and this research aligns with their goals but focuses on deep learning models with adaptive precision and randomized PCA, enhancing edge intelligence capabilities. The approach [4] complements energy efficiency in edge computing by introducing advanced precision control techniques for neural networks, contributing to energy-efficient edge intelligence.

Through this review, a comprehensive backdrop for our innovative approach, which focuses on enhancing edge intelligence with layer-wise adaptive precision and a randomized PCA-based Hybrid-Net design, addresses the specific challenges and intricacies of edge computing environments.

The binarized neural networks explored wide reduced-precision networks [5], [6], [7]. These works have paved the way for reduced-precision neural networks. This research builds upon these foundations by incorporating layer-wise precision adaptation for edge intelligence applications. A low-effort approach using PCA in network design [8] for convolutional neural network framework. This approach takes inspiration from their use of PCA but applies it in a layer-wise adaptive manner, optimizing neural networks for edge intelligence.

Energy-efficient deployment of edge data centers for mobile clouds within the framework of Sustainable IoT [9], this paper resonates with their emphasis on resource-efficient edge computing, particularly in the context of neural network precision control. The Principal Component Analysis study focuses on data analysis [10], shedding light on thedata-driven approach to understanding and processingcomplex datasets. A time-efficient dynamic threshold-based load balancing technique for Cloud Computing [11], which is relevant to the efficient allocation of computational resources. A comprehensive review of deep learning in the context of edge computing [12], aligns with the goals of this research in enhancing edge intelligence. The significant contributions [13][14][15] to energy-efficient computing include VM placement in cloud data centers, and load balancing in cloud computing. These additional references enrich the background and contribute to the broader understanding of edge intelligence and machine learning optimization.

In summary, this research in enhancing edge intelligence with layer-wise adaptive precision and randomized PCA builds upon existing techniques related to edge computing, energy-efficient neural networks, and dimensionality reduction. We extend these methodologies to provide fine-grained control over precision in neural networks, ultimately enhancing their suitability for edge intelligence applications.

III. METHODOLOGY

In this paper, a method called the Randomized Principal Component Analysis (RPCA) will work on the hybrid-net design. PCA (Principal Component Analysis) and randomized PCA are techniques used for dimensionality reduction, but they differ in how they compute the principal components. Randomized PCA is a faster alternative to PCA, approximating the principal components using randomization techniques. Instead of computing the eigenvectors of the covariance matrix, randomized PCA applies a random linear transformation to the input data to obtain a smaller matrix and then computes the eigenvectors of this smaller matrix. This smaller matrix can be computed much faster than the full covariance matrix, making randomized PCA a faster alternative to PCA for large datasets.

A. RPCA-driven Hybrid-Net Design

In the pursuit of advancing edge intelligence and optimizing neural networks for resource-constrained environments, the Randomized PCA-Based Hybrid-Net Design algorithm emerges as a groundbreaking approach. This algorithm introduces a novel framework for enhancing neural network performance while addressing the challenges posed by edge computing scenarios. At its core, the algorithm leverages Randomized Principal Component Analysis (PCA) as a key tool for dimensionality reduction, enabling the efficient processing of data on edge devices. The algorithm's architecture is structured around layer-wise adaptive precision control, resulting in a Hybrid-Net design that balances computational efficiency and model accuracy.

The algorithm begins by employing the Randomized PCA function, which operates on layer activations. It reduces the dimensionality of the data while retaining critical information, a crucial step for resource-constrained devices. The threshold parameter T plays a crucial role in controlling the amount of variance preserved, ensuring a fine-tuned balance between data reduction and model fidelity. This threshold-driven approach is a cornerstone of the algorithm's adaptability.

The subsequent steps direct the creation of the Hybrid Net, a neural network architecture designed to maximize efficiency while maintaining performance. This process involves training a binary neural network with N layers, setting thresholds (T and Δ) to guide the Randomized PCA, and determining the bit precision (kb) of significant layers. The algorithm dynamically identifies important layers by comparing the dimensionality of the activations and records them in the Sig_layer list.

These significant layers are then assigned higher bit precision,optimizing their representation for improved performance.

The Hybrid Net's initialization and training phases are crucial, with weight initialization and training techniques ensuring that the model learns effectively. The algorithm's ability to adapt bit precision at different layers allows it to strike a balance between computational efficiency and model accuracy, making it a compelling approach for edge intelligence applications.

The Randomized PCA-Based Hybrid-Net Design algorithm represents a significant advancement in the field of edge computing and neural network optimization. It harnesses the power of Randomized PCA and layer-wise precision control to create efficient yet accurate neural network architectures. This research provides a promising avenue for realizing the potential of edge intelligence in a wide range of applications where computational resources are limited, latency is critical, and data decentralization is paramount.

Table 1: Comparison of Randomized PCA and PCA

Feature	**RPCA**	**PCA**
Scalability	Large datasets that PCA cannot fit into memory can be handled using RPCA.	May struggle with large data sets.
Speed	Faster for large datasets.	Slower for large datasets.
Sparsity	Handle sparse datasets well.	Not well suited for sparse datasets.
Real-Time	Suitable for real-time.	Not suitable for real-time.
Eigen- value Ordering	Randomized ordering of eigenvalue.	Deterministic ordering of eigenvalues
Application	Suitable for large-scale applications and real-time problems.	General-purpose established method.

IV. PROPOSED WORK

The proposed RPCA for hybrid net design in place of PCA to improve efficiency. The significant dimensionality of layers sometimes increases. This section proposes an approach for identifying significant layers and then using those layers' increased bit-precision to create mixed-precision networks.

Algorithm 1 : Randomized PCA-Based Hybrid-Net Design

Function Randomized PCA(activations, layer, T)

1. [M, H, W, O] ← size(activations[layer])
2. act_fl ← flatten(activations[layer], M * H * W, O)
3. Perform randomized PCA on act_fl
4. tot_var ← total variance
5. cum_var ← cumulative sum of variance
6. k ← num of components with cum_var < T * tot_var
7. **return** k

Directs the hybrid neural network process

8. Train a N-layer binary network
9. Set Threshold T
10. Set Delta Δ
11. Set Delta kb // Bit Precision of significant layers
12.
13. k_prev ← 0
14. Siglayer ← empty list

15. **for** i ← 0 **to** N - 1 **do**
k_i ← RandomizedPCA(activations, i, T)
if k_i - k_prev > Δ and i > 0 **then**
Add i to Siglayer
end
k_prev ← k_i
end

Initialize Hybrid-Net:

16. **for** i ← 1 to N - 1
doif i ! Siglayer
then
Prec_layer ← kb
else
Prec_layer ← 1
end
end

17. Initialize weights with the same seed
18. Train Hybrid-Net

The modified algorithm utilizes randomized PCA to identify the optimal number of principal components (k) representing the variance in each layer's activations above a threshold (t). It applies a shortened SVD and Randomized PCA to transform the activations into a 2D array and calculate the explained variance. By comparing consecutive k values, significant layers with substantial activation changes are identified. These layers are assigned lower bit precision, while the rest maintain standard precision. The algorithm initializes weights with the same seed for reproducibility and trains the hybrid network using a chosen algorithm. This approach balances computational efficiency and accuracy, resulting in a streamlined and effective network.

V. SIMULATION RESULTS

Principal Component Analysis and Randomized Principal Component Analysis are dimensionality reduction techniques. Randomized PCA is faster than PCA, making it suitable for large datasets. Time Complexity of PCA is $O(n^3)$where, n is the number of features in the dataset and RPCA is $O(^{2k})$ where, k is the number of principal components that need to be calculated and the comparison graph for PCA and RPCA on time complexity as shown in Figure 1. The consumption of energy is higher for PCA whereas memory usage for PCA is lower compared to RPCA as shown in Figures 2 and 3 respectively.

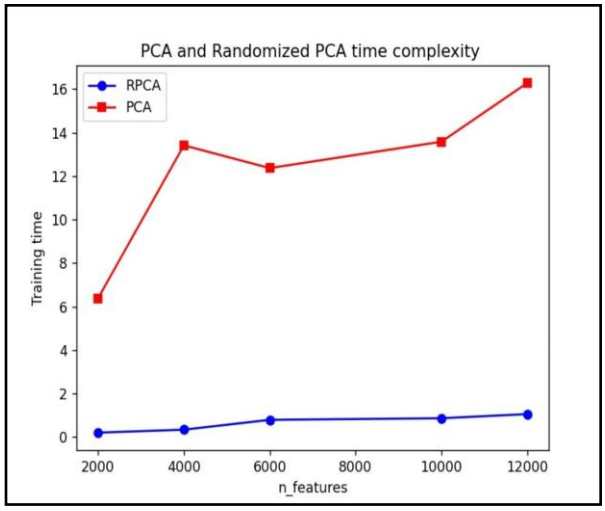

Figure 1: Time Complexity of RPCA and PCA

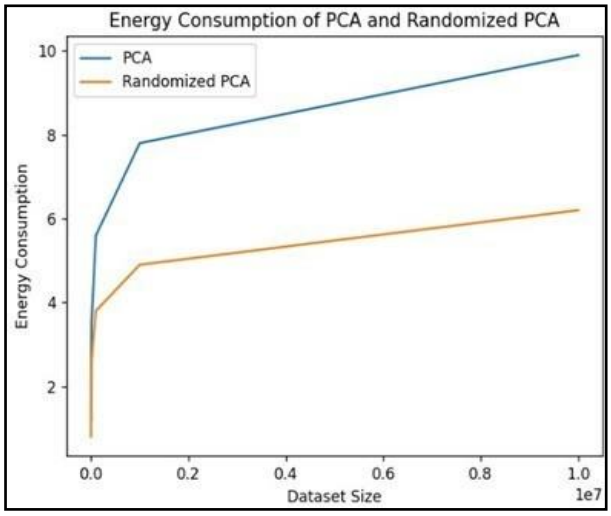

Figure 2: Energy Consumption of RPCA and PCA

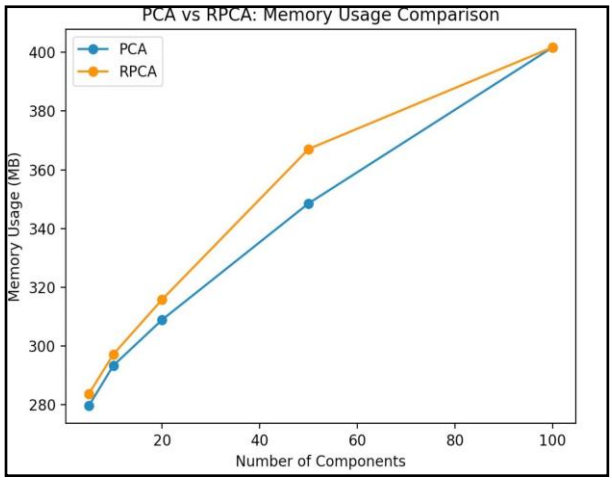

Figure 3: Memory usage comparison of RPCA and PCA

VI. RPCA IDENTIFICATION OF SIGNIFICANT LAYERS

Central to this algorithm is the process of identifying significant layers within the neural network, guided by Randomized Principal Component Analysis (RPCA). This critical step forms the foundation for subsequent layer-wise precision control and the development of the Hybrid-Net framework. In this section, the methodology behind this RPCA-driven identification process.

For each of the N layers in the neural network, we employ the Randomized PCA function to operate on the layer's activations. This function reduces the dimensionality of the activation data, transforming it into a more compact representation that retains essential information. Key to this operation is the threshold parameter (T), which determines the fraction of variance to preserve. Setting an appropriate threshold ensures that PCA captures vital patterns and features while discarding non-essential noise and redundancy.

The outcome of the RPCA analysis is a reduced-dimensional representation of the activations for each layer. The principal components (k) needed to represent a cumulative variance larger than T times the total variance of the original data are assessed to determine the relevance of each layer. Layers exhibiting higher complexity and information content will demand a larger number of principal components to meet this threshold, indicating their importance in retaining critical information. To refine this identification process, a parameter Δ, representing the minimum difference in the number of principal components between consecutive layers for a layer to be considered significant is introduced. We compare k_i, the number of components required for layer i, with k_prev, the number of components for the previous layer. If the difference (k_i - k_prev) exceeds Δ, and the layer index (A) is greater than zero, it is classified as a layer as significant and added it tothe Sig_layer list.

This RPCA-driven identification of significant layers serves as the bedrock of this algorithm, facilitating the selective reinforcement of specific layers with higher bit-precision representation in subsequent steps. It ensures that computational resources are allocated judiciously, maximizing the efficiency of the resulting Hybrid Net while upholding model accuracy. In the following sections, the construction and training of the Hybrid-Net, leveraging this layer-wise precision control to optimize edge intelligence applications. judiciously, maximizing the efficiency of the resulting Hybrid Net while upholding model accuracy. In the following sections, the construction and training of the Hybrid-Net, leveraging this layer-wise precision control to optimize edge intelligence applications.

VII. EXPERIMENTAL STUDY AND RESULT ANALYSIS

For calculating principal components, randomized PCA provides a quicker approximation to the classic PCA approach. It is used to minimize the dimensionality of the data and extract the most informative features from neural network layer activations. This is particularly beneficial for deep neural networks with high-dimensional activations. The number of primary components that should be employed to describe the data is determined automatically via threshold selection. Rather than needing to manually modify this parameter, it enables a more automated and data-driven approach to determine the number of components. It entails

adding a penalty term to the network's loss function, which discourages big weights and favours simpler models. This prevents the network from memorizing the training data and allows it to generalize to new data more effectively. Application of the constructed PCA method to three datasets of varied sizes to evaluate the performance of the proposed scheme. The findings were compared to those produced using the Scikit-learn package. The findings revealed that the implementation produced comparable results to Scikit-learn but with a considerable decrease in calculation time for big datasets. Finally, a hybrid neural network is created by combining the significant and non-significant layers with different precision settings, and the weights are initialized with the same value. The hybrid neural network is then trained to achieve the desired accuracy.

VIII. CONCLUSION AND FUTURE WORKS

The efficiency of the PCA technique in decreasing the dimensionality of huge datasets in this project by implementing it in Python without the need for additional libraries by using Randomized PCA, Automatic threshold selection, and Weight decay is demonstrated. The suggested technique produced comparable results to the Scikit- learn package while drastically lowering computation time. In summary, while both PCA and Randomized PCA are valuable dimensionality reduction approaches, Randomized PCA is a quicker and more scalable variation of PCA that may accomplish equivalent results with significantly less time complexity.

To avoid more memory usage for RPCA, it may be necessary to employ distributed computing techniques or use other dimensionality reduction methods specifically designed for large-scale data analysis, such as Incremental PCA or Kernel PCA with incremental updates.

REFERENCES

(1) I. Chakraborty, D. Roy, I. Garg, A. Ankit, & K. Roy (2019). Constructing energy-efficient mixed-precision neural networks through principal component analysis for edge intelligence. Nature Machine Intelligence, 2, 43-55.

(2) J. Gubbi, R. Buyya, S. Marusic, & M. Palaniswami Internet of things (IoT): A vision, architectural elements, and future directions. Futur. Gener. Comput. Syst. 29, 1645–1660, DOI: 10.1016/ j.future.2013.01.010 (2013).

(3) S. Yao, S. Hu, Y. Zhao, A. Zhang, & T. Abdelzaher, DeepSense. In Proceedings of the 26th International Conference on World Wide Web - WWW '17, DOI: 10.1145/3038912.3052577 (ACM Press, 2017).

(4) Z. Zhou, Shojafar Mohammad, Abawajy Jemal, H. Yin, H. Lu (2022): ECMS: An Edge Intelligent Energy Efficient Model in Mobile Edge Computing. Deakin University. Journal contribution.

(5) M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv & Y. Bengio, Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830 (2016).

(6) A. Mishra, E. Nurvitadhi, J. J. Cook, & Marr, D. Wrpn: wide reduced-precision networks. arXiv preprint arXiv:1709.01134 (2017).

(7) I. Garg, P. Panda, & K. A. Roy, low effort approach to structured cnn design using pca. arXiv preprint arXiv:1812.06224 (2018).

(8) S. K. Mishra, D. Puthal, B. Sahoo, S. Sharma, Z. Xue and A. Y. Zomaya, "Energy-Efficient Deployment of Edge Datacenters for Mobile Clouds in Sustainable IoT," in IEEE Access, vol. 6, pp. 56587-56597, 2018, doi: 10.1109/ ACCESS.2018.2872722.

(9) D. Xu, T. Li, Y. Li, X. Su, Tarkoma, S., Jiang, T., Crowcroft, J. and Hui, P., 2020. Edge intelligence: Architectures, challenges, and applications. arXiv preprint arXiv:2003.12172.

(10) S. Sehgal, H. Singh, M. Agarwal, V. Bhasker and Shantanu, "Data analysis using principal component analysis," 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), Greater Noida, India, 2014, pp. 45-48, doi: 10.1109/ MedCom.2014.7005973.

(11) S. K. Mishra, M. A. Khan, B. Sahoo, D. Puthal, M. S. Obaidat and K. Hsiao, "Time efficient dynamic threshold- based load balancing technique for Cloud Computing," 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, China, 2017, pp. 161-165, doi: 10.1109/CITS.2017.8035327.

(12) J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," in *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655-1674, Aug. 2019, doi: 10.1109/ JPROC.2019.2921977.

(13) S. K. Mishra, R. Deswal, S. Sahoo and B. Sahoo, "Improving energy consumption in the cloud," 2015 Annual IEEE India Conference (INDICON), New Delhi, India, 2015, pp. 1-6, doi: 10.1109/ INDICON.2015.7443710.

(14) Mishra, S. K., Puthal, D., Sahoo, B., Jayaraman, P. P., Jun, S., Zomaya, A. Y., & Ranjan, R. (2018). Energy-efficient VM placement in the cloud data center. *Sustainable computing: informatics and systems*, *20*, 48-55.

(15) Mishra, S. K., Sahoo, B., & Parida, P. P. (2020). Load balancing in cloud computing: a big picture. *Journal of King Saud University-Computer and Information Sciences*, *32*(2), 149-158.