WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Optimizing Neural Network Energy Efficiency Through Low-Rank Factorisation And PDE-Driven Dense Layers

Jiby Mariya Jose

Independent Researcher
jiby.jose1404@gmail.com

ABSTRACT :

As deep learning models continue to grow in complexity, the computational and energy demands associated with their training and deployment are becoming increasingly significant, particularly for convolutional neural networks (CNNs) deployed on CPU-bound and resource- limited devices. Fully connected (FC) layers, while vital, are energy-intensive, accounting for 85.7% of a network's parameters but contributing only 1% of the computations. This research proposes a novel approach to optimising these layers for greater energy efficiency by integrating low-rank factorisation with differential partial differential equations (PDEs). The introduction of the LowRankDense layer, which combines low-rank matrix factorisation with a differential PDE solver, aims to reduce both the parameter count and energy consumption of FC layers. Experiments conducted on the MNIST, Fashion MNIST, and CIFAR-10 datasets demonstrate the effectiveness of this approach, yielding promising results in terms of reduced energy usage and maintaining comparable performance, thereby enhancing the practicality and sustainability of CNNs for widespread use in environments with limited computational resources.

Introduction :

Convolutional neural networks (CNNs) have emerged as foundational tools across a broad spectrum of applications, from drug discovery to human-machine interactions [1]–[3]. These architectures are known for their ability to capture complex patterns in data, yet their success comes with significant demands on computational resources [4], [5]. As CNNs have become more sophisticated, the energy required for both training [6]–[8] and inference [9] has surged, raising concerns about the sustainability of such approaches, especially in energy-constrained environments.

Among the various components of a CNN, the training phase stands out as particularly resourcein- tensive [10]. During this phase, CNNs undergo iterative processes to optimise their performance, transforming and storing data across multiple layers [11]. This not only necessitates considerable computational power but also results in high energy consumption, a challenge that becomes more pronounced as networks grow in complexity. Addressing energy efficiency during training is crucial, as it not only determines the immediate resource usage but also influences the long-term operational efficiency of the model.

A critical yet energy-intensive component of CNNs is the fully connected (FC) layer [12]. Despite constituting a small fraction of the computational workload—contributing only 1 percent of the operations—FC layers are responsible for a disproportionate 85.7 percent of the network's parameters [13], [14]. This imbalance highlights the need for targeted optimisations in the design of FC layers to reduce their energy footprint without compromising model performance[15]. In this work, we focus

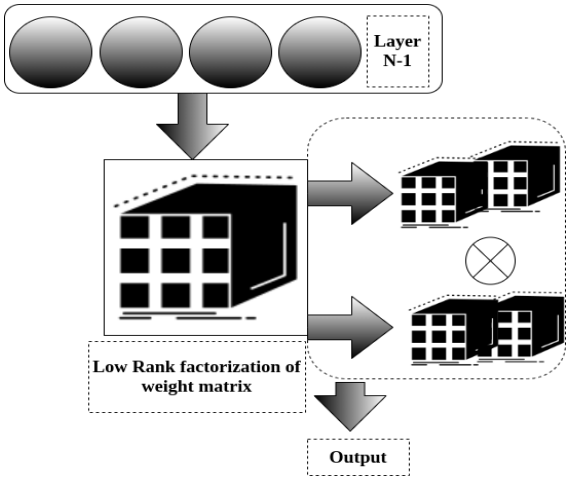

Figure 1: Low Rank Factorisation

on designing lightweight, dense layers to tackle the challenges posed by the energy-intensive nature of fully connected layers in CNNs.

Background Study

In this section, we discuss some of the recent work focused on dense layers. The work [16] has replaced matrix multiplication with matrix factorisation to reduce the parameters of the dense layer. However, this study also addresses convolutional layers, and thus the impact specifically on dense layers is not thoroughly explored. Another study [17] replaces exact adders and multipliers in the MAC blocks with approximate ones to reduce the parameters of dense layers. While this approach aims to lower energy consumption, it does not fully address the tradeoff between energy efficiency and performance efficiency in dense layers. Similarly, [18] introduced parameter sharing within dense layers to reduce computation overhead, but the impact on the energy-performance tradeoff remains unclear. Additionally, [19] focused on compressing the weight matrix through permutation, yet this method does not provide a comprehensive analysis of energy and performance efficiency. Overall, existing research lacks a focused investigation into designing dense layers with a clear emphasis on reducing the tradeoff between energy efficiency and performance efficiency.

Methodology :

In this work, we propose a modified version of the dense layer, termed the **LightDense**, which integrates low-rank factorisation, a PDE-based update mechanism, and subsequent Normalisation. The design and implementation of this layer are as follows:

Low-Rank Factorisation

The weight matrix of the dense layer is decomposed into two smaller matrices, significantly reducing the overall number of parameters. Specifically, the factorisation is achieved using two weight matrices, W_1 and W_2, where W_1 has a smaller rank compared to the original weight matrix. This approach aims to maintain the expressiveness of the model while lowering the computational burden and memory usage.Figure 1 shows the low rank computation performed.

Let W be the original weight matrix with dimensions $m \times n$. The number of parameters in this dense layer is:

$$\textbf{Parameters}_{\textbf{original}} = m \times n \qquad (1)$$

After low-rank Factorisation, W is approximated by:

$$W \approx W_1 W_2 \qquad (2)$$

where W_1 is $m \times r$ and W_2 is $r \times n$. The number of parameters now is:

$$\textbf{Parameters}_{\textbf{factorized}} = (m \times r) + (r \times n) \qquad (3)$$

Since r is much smaller than m and n, we have:

$$(m \times r) + (r \times n) < m \times n \qquad (4)$$

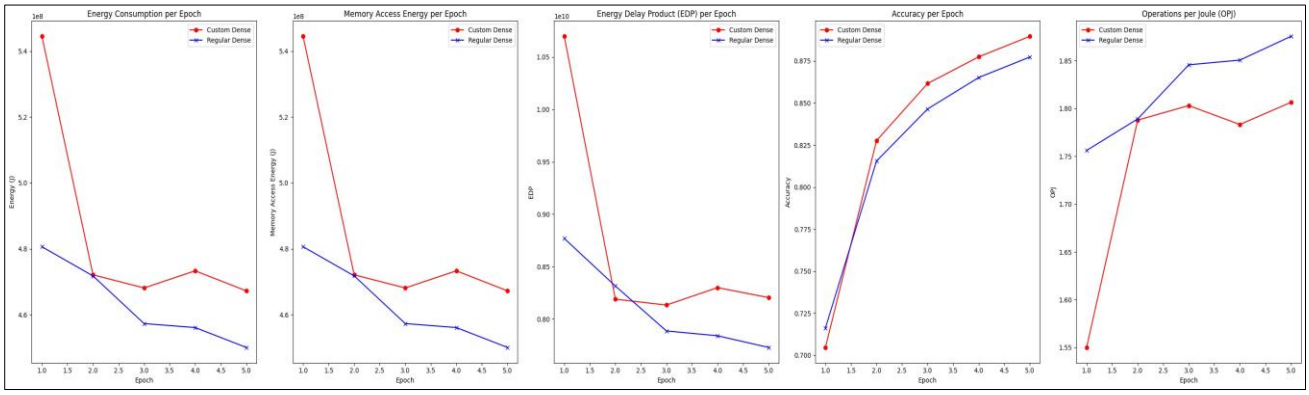

Figure 2: Evaluation with Fashion MNIST dataset

Differential PDE Solver

To enhance the adaptability of the layer to varying input data, we introduce a simple PDE solver that simulates update mechanism that allows the model to respond more effectively to changes in the input, potentially leading to improved performance and reduced energy consumption during training. Given a matrix z and a PDE solver that models dynamics, the updated matrix $z_{\text{pde\_solved}}$ is given by:

$$z_{\text{pde\_solved}} = z + \Delta t \cdot \frac{\partial z}{\partial t} + \mathbf{v} \cdot \nabla z \qquad (5)$$

where:

- z is the original matrix.
- Δt is the time step used in the PDE solver.

- $\frac{\partial z}{\partial t}$ represents the temporal rate of change of z.
- $\mathbf{v}$ is the velocity field vector representing advection dynamics.
- ∇z denotes the spatial gradient of z.

Normalisation

After applying the PDE solver, a Normalisation step is included to stabilize the activations and improve convergence. This Normalisation ensures that the updated activations are appropriately scaled, addressing issues related to vanishing or exploding gradients and thereby contributing to more robust training dynamics.

Given the matrix $z_{\text{pde\_solved}}$, the L2 Normalisation is applied as follows:
The normalised value $z_{\text{normalised}}$ is given by:

$$z_{\text{normalised}} = \frac{z_{\text{pde\_solved}}}{||z_{\text{pde\_solved}}||_2} \quad \textbf{(6)}$$

where the L2 norm of $z_{\text{pde\_solved}}$ is:

$$||z_{\text{pde\_solved}}||_2 = \sqrt{\sum_i \sum_j (z_{\text{pde\_solved}}[i,j])^2} \quad (7)$$

For Normalisation along a specific axis, it can be written as:

$$z_{\text{normalised}}[i,j] = \frac{z_{\text{pde\_solved}}[i,j]}{\sqrt{\sum_k (z_{\text{pde\_solved}}[i,k])^2}}$$

This combination of low-rank Factorisation, dynamic PDE-based updates, and Normalisation in the LowRankDense layer provides a novel approach to improving the efficiency and effectiveness of dense layers within CNNs, particularly in the context of reducing energy consumption during training.

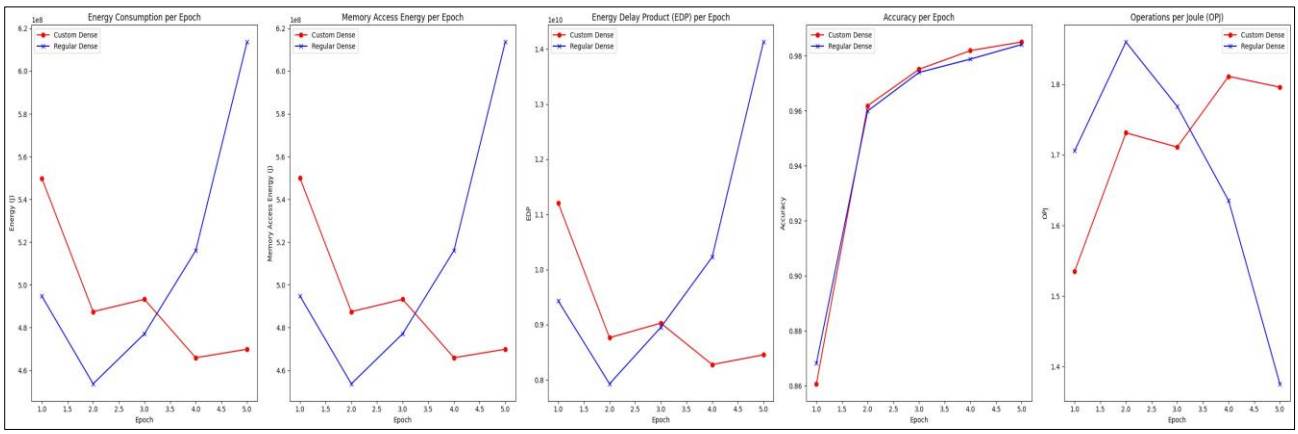

Figure 3: Evaluation with MNIST dataset

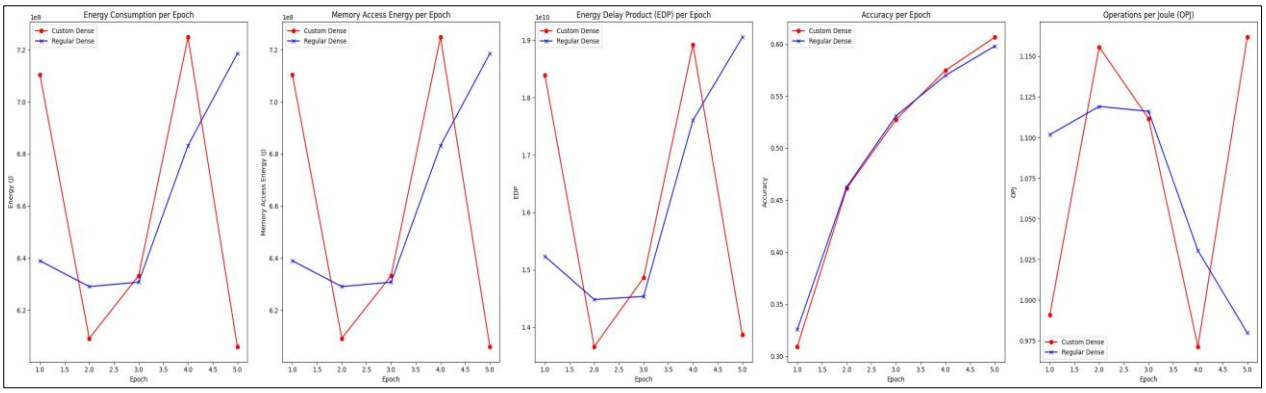

Figure 4: Evaluation with CIFAR-10 dataset

Experiments :

The experimental setup involves training the model using standard datasets including MNIST, CIFAR- 10, CIFAR-100, and FASHION-MNIST. We evaluate the performance of our proposed dense layer by comparing it with traditional dense layers in terms of both energy consumption and accuracy. To quantify the impact of our proposed layer on computational efficiency, we use the pyRAPL tool for energy measurement. All experiments are conducted on an AMD Ryzen CPU, ensuring a consistent and controlled environment for performance evaluation.Table 1 and 2 illustrate the model architecture utilized for testing. Our experimental results illustrate that the custom Dense layer

Layer (type)	Output Shape	Param #
InputLayer	(None, 32, 32, 3)	0
Conv2D	(None, 30, 30, 32)	896
MaxPool2D	(None, 15, 15, 32)	0
Conv2D	(None, 13, 13, 64)	18,496
MaxPool2D	(None, 6, 6, 64)	0
Conv2D	(None, 4, 4, 64)	36,928
GAP2D	(None, 64)	0
LightDense	(None, 128)	1,664
LightDense	(None, 100)	1,924

Table 1: CNN with Light Dense

Layer (type)	Output Shape	Param #
InputLayer	(None, 32, 32, 3)	0
Conv2D	(None, 30, 30, 32)	896
MaxPooling2D	(None, 15, 15, 32)	0
Conv2D	(None, 13, 13, 64)	18,496
MaxPooling2D	(None, 6, 6, 64)	0
Conv2D	(None, 4, 4, 64)	36,928
GAP2D	(None, 64)	0
Dense	(None, 128)	8,320
Dense	(None, 100)	12,900

Table 2: CNN without Light Dense

consistently outperforms the standard Dense layer across the following datasets FashionMNIST, MNIST, and CIFAR-10, in terms of energy efficiency, accuracy, and performance.Figures 3,Fig.2 and Fig.4 demonstrates the results.It demonstrates lower overall energy consumption and Energy-Delay Product (EDP) values while achieving higher or comparable accuracy, particularly evident in the MNIST dataset where it outperforms in accuracy and reduces operational energy per joule (OPJ). The custom layer also shows improvements in memory access energy, particularly in the CIFAR-10 dataset, indicating better optimization for energy usage without compromising on model accuracy or validation performance.

Conclusion :

In conclusion, this research highlights the critical need to address the energy inefficiencies of fully connected (FC) layers in convolutional neural networks (CNNs), particularly for deployments on CPU-bound and resource-constrained devices. By introducing the LowRankDense layer named as Lightdense, which effectively combines low-rank matrix Factorisation with differential Partial Differential Equations (PDEs), we have demonstrated a promising strategy for reducing both param- eter count and energy consumption without sacrificing performance. The experimental results on the MNIST dataset validate the efficacy of this approach, showing significant reductions in energy usage while maintaining comparable accuracy. This work not only advances the state-of-the-art in energy-efficient deep learning but also contributes to the sustainability of CNNs in diverse application scenarios, particularly in environments where computational resources are limited. Future work will explore the generalisability of this approach to more complex datasets and broader network architectures, further evaluating its potential for widespread adoption.

Acknowledgement

We would like to express our sincere gratitude to Dr. Dan Zhao, from MIT, for his invaluable guidance and mentorship throughout this project. His insights and expertise have greatly contributed to the development and success of this work. We also extend our thanks to the NeurIPS CCAI team for assigning such a supporting mentor and for their continuous support throughout the process.

REFERENCES :

1. T. Sun, B. Feng, J. Huo, *et al.*, "Artificial intelligence meets flexible sensors: Emerging smart flexible sensing systems driven by machine learning and artificial synapses," *Nano-Micro Letters*, vol. 16, no. 1, p. 14, 2024.
2. X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, 2024.
3. R. Khanam, M. Hussain, R. Hill, and P. Allen, "A comprehensive review of convolutional neural networks for defect detection in industrial applications," *IEEE Access*, 2024.
4. W. Lim, K. Y. S. Chek, L. B. Theng, and C. T. C. Lin, "Future of generative adversarial networks (gan) for anomaly detection in network security: A review," *Computers & Security*, p. 103 733, 2024.
5. M. A. K. Raiaan, S. Sakib, N. M. Fahad, *et al.*, "A systematic review of hyperparameter opti- mization techniques in convolutional neural networks," *Decision Analytics Journal*, p. 100 470, 2024.
6. J. Tmamna, E. B. Ayed, R. Fourati, *et al.*, "Pruning deep neural networks for green energy- efficient models: A survey," *Cognitive Computation*, pp. 1–22, 2024.

7. F. Chen, S. Li, J. Han, F. Ren, and Z. Yang, "Review of lightweight deep convolutional neural networks," *Archives of Computational Methods in Engineering*, vol. 31, no. 4, pp. 1915–1937, 2024.
8. S. Bhalgaonkar, M. Munot, *et al.*, "Model compression of deep neural network architectures for visual pattern recognition: Current status and future directions," *Computers and Electrical Engineering*, vol. 116, p. 109 180, 2024.
9. M. Sponner, B. Waschneck, and A. Kumar, "Adapting neural networks at runtime: Current trends in at-runtime optimizations for deep learning," *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–40, 2024.
10. H.-I. Liu, M. Galindo, H. Xie, *et al.*, "Lightweight deep learning for resource-constrained environments: A survey," *ACM Computing Surveys*, 2024.
11. Y. Ren and X. Cheng, "Review of convolutional neural network optimization and training in image processing," in *Tenth International Symposium on Precision Engineering Measurements and Instrumentation*, SPIE, vol. 11053, 2019, pp. 788–797.
12. Y. Chen, B. Zheng, Z. Zhang, Q. Wang, C. Shen, and Q. Zhang, "Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–37, 2020.
13. T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze, "A method to estimate the energy consumption of deep neural networks," in 2017 51st asilomar conference on signals, systems, and computers, IEEE, 2017, pp. 1916–1920.
14. N. Matsumura, Y. Ito, K. Nakano, A. Kasagi, and T. Tabaru, "A novel structured sparse fully connected layer in convolutional neural networks," Concurrency and Computation: Practice and Experience, vol. 35, no. 11, e6213, 2023.
15. J. Šıma, J. Cabessa, and P. Vidnerová, "On energy complexity of fully-connected layers," Neural Networks, p. 106 419, 2024.
16. D. Jha, A. Yazidi, M. A. Riegler, D. Johansen, H. D. Johansen, and P. Halvorsen, "Lightlayers: Parameter efficient dense and convolutional layers for image classification," in Parallel and Distributed Computing, Applications and Technologies: 21st International Conference, PDCAT 2020, Shenzhen, China, December 28–30, 2020, Proceedings 21, Springer, 2021, pp. 285–296.
17. M. Esmali Nojehdeh and M. Altun, "Energy-efficient hardware implementation of fully connected artificial neural networks using approximate arithmetic blocks," *Circuits, Systems, and Signal Processing*, vol. 42, no. 9, pp. 5428–5452, 2023.
18. M. Biswas, R. Sikdar, R. Sarkar, and M. Kundu, "A cnn model with pseudo dense layers: Some case studies on medical image classification," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 13, no. 1, p. 41, 2024.
19. D. Nagaraju and N. Chandrachoodan, "Compressing fully connected layers of deep neural networks using permuted features," *IET Computers & Digital Techniques*, vol. 17, no. 3-4, pp. 149–161, 2023.