Joint Optimization of Data Path and Memory Hierarchy for Deep Learning Accelerator Architectures

Wang Ying[1], Han Yun, Xu Xinyu[2]

The 58th Research Institute of China Electronics Technology Group Corporation, Wuxi, Jiangsu, 214000

Abstract: ***This paper investigates the architectural design of deep learning accelerator chips, with a focus on addressing the challenge of inefficient coordination between the data path and memory hierarchy in traditional chip architectures when processing deep learning tasks. Deep learning algorithms are characterized by substantial computational demands and complex data access patterns, which impose stringent requirements on both computing performance and storage capacity. To address these challenges, a coordinated optimization model is developed based on the intrinsic characteristics of deep learning algorithms. In data path design, systolic array structures are adopted to optimize matrix operation pathways and enhance the efficiency of computing units. In the memory hierarchy, performance improvements are achieved through optimized cache strategies and the integration of high-bandwidth memory technologies, thereby enhancing data storage and access performance.***

Keywords: Deep learning accelerator chip; Data path; memory hierarchy; Coordinated optimization.

1. INTRODUCTION

As deep learning achieves remarkable results in numerous fields such as computer vision, natural language processing, and speech recognition, its demand for computing performance is growing exponentially. Deep learning models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variants, contain massive parameters and complex computational logic. When traditional Central Processing Units (CPUs) and Graphics Processing Units (GPUs) handle these tasks, they gradually expose issues of low computational efficiency and high energy consumption, which not only limit the rapid development of deep learning technology but also hinder its wide application in resource-constrained devices.

The data path, as the channel for data transmission and processing in a chip, directly affects the overall performance of the chip; the memory hierarchy is responsible for data storage and management, with different levels of memory differing in capacity, speed, and cost. How to construct an efficient memory hierarchy to achieve fast data access and storage is key to improving chip performance. However, in traditional deep learning acceleration chip architecture design, the data path and memory hierarchy are often designed independently, lacking effective collaborative optimization, leading to significant delays and energy consumption during data transmission and storage, which severely restricts the chip's computational performance and energy efficiency ratio.

By optimizing the collaborative working mechanism between the data path and memory hierarchy in deep learning acceleration chip architecture design, the computational efficiency of the chip can be significantly improved, energy consumption reduced, thereby accelerating the training and inference processes of deep learning models, promoting the application and development of deep learning technology in more fields, and possessing broad application prospects and economic value. In the realm of large language models (LLMs), researchers have focused on evaluating and enhancing their empathetic capabilities, with Chen et al. (2024) introducing Emotionqueen, a specialized benchmark designed to systematically assess the empathy performance of LLMs [1]. For ophthalmological applications in low- and middle-income countries (LMICs), Restrepo et al. (2025) developed Multi-OphthaLingua, a multilingual benchmark that not only evaluates LLM-based ophthalmological question-answering systems but also aims to mitigate inherent biases [3]. In the medical data processing space, Thao et al. (2024) proposed Medfuse, a multimodal framework that integrates masked lab-test modeling with LLMs to effectively fuse electronic health record (EHR) data, improving the utility of complex medical datasets [4]. Complementing this, Restrepo et al. (2025) explored representation learning of lab values through a masked autoencoder, offering a novel approach to extract meaningful insights from clinical laboratory data [5]. Multimodal data fusion has emerged as a critical area, with Moukheiber et al. (2024) presenting a framework that

combines satellite images and public health data, enabling more comprehensive analysis of public health trends by leveraging the strengths of both data types [2]. In statistical computing, Lin et al. (2023) advanced the calculation of the Poisson multinomial distribution, providing valuable tools for ecological inference and machine learning applications [6]. For neural network optimization, Gong et al. (2023) conducted an extensive review of lightweighting techniques, which are essential for deploying neural networks on resource-constrained devices [7]. In recommendation systems, Junxi et al. (2024) introduced GCN-MF, a graph convolutional network model based on matrix factorization that enhances the accuracy of personalized recommendations [8]. Beyond AI and data science, research has also addressed practical industry challenges. Zhang (2024) developed a cohesive hierarchical clustering-based approach to dynamically adapt the supply and demand of power emergency materials, improving the efficiency of emergency response systems [9]. Lin et al. (2025) proposed a Bayesian framework for modeling multivariate degradation data with dynamic covariates, enhancing the reliability assessment of engineering systems [10]. In digital economy applications, Yi (2025) presented a real-time fair-exposure ad allocation mechanism using contextual bandits-with-knapsacks, aiming to support small and medium-sized businesses (SMBs) and underserved creators [11]. In optical engineering, Tang et al. (2020) optimized the design of shallow-angle grating couplers for indium phosphide devices, improving vertical emission performance [12]. For cloud security, Deng (2025) proposed a homomorphic encryption-based mechanism to ensure data integrity and prevent tampering in cloud storage environments [13]. In financial infrastructure protection, Mehta et al. (2026) put forward a national AI security framework, addressing the growing need to safeguard critical financial systems from AI-driven threats [14]. In customer analytics, Zhou (2026) analyzed hierarchical needs in US automotive customer feedback, uncovering a nexus between sentiment and functional requirements [15]. Finally, Wensi (2026) explored AI-enabled data visualization marketing for automated production lines, demonstrating its potential to build customer trust and increase lead-to-order conversion rates [16].

2. FUNDAMENTALS OF DEEP LEARNING ACCELERATION CHIP ARCHITECTURE

In deep learning algorithms, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) and their variants are widely used. CNNs process images through convolutional layers, pooling layers, and fully connected layers; convolution kernels slide to extract local features, enabling weight sharing; pooling layers reduce dimensions to enhance robustness, such as AlexNet which can accurately extract image features. RNNs are used to process sequence data, but they have gradient issues when handling long sequences; LSTMs solve this problem through gating mechanisms, and GRUs through simplified structures, performing excellently in tasks such as machine translation. These algorithms are computationally complex, requiring chips to have strong parallel computing, efficient data processing, and fast memory access capabilities.

Traditional CPU and GPU architectures have obvious drawbacks in processing deep learning tasks. CPUs have few cores, weak parallel computing capabilities, and low memory bandwidth, resulting in low efficiency when facing a large number of matrix operations and data reading/writing in deep learning. Although GPUs are suitable for parallel computing, they have inflexible resource allocation, high energy consumption, and fixed memory hierarchies that are difficult to adapt to the dynamic data access requirements of deep learning, leading to large memory access delays and affecting task execution speed. As deep learning models develop, these limitations are becoming increasingly prominent.

Deep learning acceleration chip architectures are specifically designed for efficient processing of deep learning tasks, typically including computing units, storage units, data paths, and control units. The computing unit contains a large number of parallel cores to perform operations; for example, the matrix multiplication unit of TPUs improves computational efficiency. The storage unit is divided into different levels to meet speed and capacity requirements. The data path optimizes structure and bandwidth to reduce transmission latency. The control unit coordinates the work of various units and allocates resources as needed[1]. Its working principle is based on the optimization of deep learning algorithms, such as using systolic arrays to achieve efficient matrix multiplication, optimizing storage hierarchies, and utilizing data compression techniques to improve performance. Compared with traditional architectures, it has significant advantages in computational efficiency, energy consumption, and storage bandwidth utilization. For instance, Google's TPU has much higher computational efficiency than traditional GPUs in inference tasks, making it a key driver of deep learning development [2].

3. DATA PATH DESIGN AND OPTIMIZATION

In deep learning acceleration chips, the data path is responsible for data transmission between storage and computing units, consisting of arithmetic logic units (ALUs), register banks, and data buses. ALUs perform basic operations in deep learning such as matrix multiplication. Register banks temporarily store data to reduce access latency and support model computation. The data bus determines transmission speed; deep learning acceleration chips often use high-speed buses to meet data transmission requirements. Various components collaborate to complete data processing. For example, during model forward propagation, data flows from the storage unit to the register bank, then is transmitted via the bus to the ALU for computation, and the results are returned to the register bank for storage.

Pipelining technology decomposes complex computations into multiple stages, each executed by specialized components, with registers inserted in between to temporarily store results. This improves the throughput and processing speed of the data path and is widely used in deep learning convolution and matrix multiplication operations. For example, convolution operations can be divided into stages, allowing multiple computing tasks to be processed simultaneously. Parallel processing technology increases computing resources, such as multiple ALUs, enabling parallel processing of multiple tasks, which aligns with the characteristics of deep learning involving a large amount of matrix operations and parallel operations. Multi-threading technology can also be used to improve efficiency. By comprehensively applying these methods and technologies, a data path that meets the high-performance requirements of deep learning can be designed [3].

Deep learning tasks have special computing requirements and complex data flows, placing high demands on data paths. Models involve numerous matrix and convolution operations, resulting in large data read/write volumes and irregular access patterns. To meet these requirements, the matrix operation path can be optimized by using a systolic array structure to reduce data transmission and storage overhead. Additionally, the data path structure can be optimized by using high-speed caches to reduce off-chip memory access, designing high-speed buses or multi-bus structures to increase bandwidth, and utilizing data compression techniques to reduce transmission volume. These measures improve data path performance and meet the requirements of deep learning for computational efficiency and data processing [4].

4. STORAGE HIERARCHY DESIGN AND OPTIMIZATION

In deep learning acceleration chips, the memory hierarchy is responsible for data storage and management. Different levels of memory components work together to meet the chip's diverse requirements for data storage in terms of capacity, speed, and cost[5]. Registers are Adjacent to the computing unit, with extremely fast access speed but small capacity, used for temporarily storing intermediate results and model parameters to support rapid computation. Cache is located between the CPU and main memory, based on SRAM technology, which reduces the number of CPU accesses to main memory by pre-storing data, thereby improving access speed. Main memory uses DRAM technology, with large capacity but slow access speed, storing complete model parameters, training, and inference data. Auxiliary storage (such as hard disks and SSDs) has huge capacity and is used for long-term storage of massive data and programs, but its access speed is extremely slow, and data needs to be loaded into main memory first when used.

Performance indicators such as access speed, storage capacity, and bandwidth of the memory hierarchy are crucial for deep learning acceleration chips. In terms of access speed, registers are the fastest, followed by cache, then main memory, and auxiliary storage is the slowest; a slow access speed of the memory hierarchy will prolong the training time of deep learning models. In terms of storage capacity, registers and cache have limited capacity, main memory meets basic storage needs, and auxiliary storage has huge capacity for long-term storage of large-scale data; insufficient capacity of the memory hierarchy will affect chip performance. Bandwidth reflects data transmission capability; insufficient bandwidth will increase data transmission delay and slow down computing speed. An efficient memory hierarchy can quickly provide data, store it reasonably, and ensure fast data transmission.

Deep learning has high requirements for the memory hierarchy, requiring the adoption of various optimization methods. Optimize cache strategies by improving traditional LRU algorithms, dynamically adjusting cached data according to the characteristics of deep learning tasks, and optimizing cache associativity and size; adopt high-bandwidth memory, such as DDR4, DDR5, or multi-channel memory technology, to improve data transmission rate; utilize new storage technologies, such as NVM (PCM, RRAM), as a supplement to main memory or cache, and adopt distributed storage architecture to improve the reliability and scalability of the storage system.

5. DATA PATH AND MEMORY HIERARCHY CO-OPTIMIZATION STRATEGIES

In traditional designs, the data path and memory hierarchy are independent, leading to data transmission delays and access conflicts. The co-optimization of data path and memory hierarchy aims to improve computing efficiency and reduce energy consumption. By optimizing the协同机制, transmission delays and access conflicts are reduced, the utilization rate of computing units is improved, and redundant data transmission and storage are reduced.

Data prefetching technology predicts the data required for the next computation based on data access patterns and computing needs and prefetches it into the cache in advance. Cache consistency maintenance (such as the MESI protocol) ensures data consistency across different cache levels and computing units, improving system reliability and performance. Optimize the interface and communication mechanism between storage and computing units, and reasonably allocate storage and computing resources according to the characteristics of deep learning tasks to achieve efficient synergy.

A collaborative optimization model is constructed based on the characteristics of deep learning algorithms, including a task analysis module (responsible for analyzing the type, structure, and computational requirements of deep learning tasks and extracting key features), a resource allocation module (dynamically allocating data path and storage hierarchy resources based on task analysis results), a scheduling module (coordinating the operation of data paths and storage hierarchies, and reasonably scheduling data transmission and computing operations according to task execution order and resource allocation), and a feedback module (adjusting the optimization model based on task execution results), to achieve efficient collaboration and improve the performance of deep learning acceleration chips.

6. EXPERIMENTS AND VERIFICATION

The experiments selected the AlexNet model in CNN and the Long Short-Term Memory (LSTM) model in RNN to compare the chip operation performance before and after optimization. The experimental environment was based on an FPGA development board and a high-performance server. The chip architecture was designed and implemented using Verilog, synthesized and simulated with Xilinx ISE development tools, and the deep learning models were run using the TensorFlow framework.

Taking the AlexNet model as an example, the cumulative training time before optimization was 30 hours, and the time required for a single inference task was 150 milliseconds. After adopting the collaborative optimization strategy for data paths and storage hierarchy, the training time was sharply reduced to 10 hours (a 66.7% reduction), and the inference time was only 30 milliseconds (an 80% reduction). The key reason for the performance improvement is that collaborative optimization effectively alleviated the delay caused by data transmission between the storage hierarchy and data paths, and significantly reduced the frequency of storage access conflicts.

The LSTM model was selected for testing energy consumption indicators. When the LSTM model ran on the unoptimized chip architecture, the chip's energy consumption remained stable at 200 watts. After introducing the collaborative optimization strategy, the energy consumption dropped to 120 watts (a 40% reduction). This is mainly because the optimization strategy deeply optimized data transmission and storage operations, eliminating a large number of unnecessary data movement and storage actions. By reducing these redundant operations, the overall power consumption of the chip during deep learning tasks is fundamentally reduced, which is of great significance for improving the energy efficiency of the chip, reducing operating costs, and promoting green computing.

In the analysis of storage utilization, this paper focuses on two key indicators: the number of storage accesses and cache hit rate. Before optimization, to complete one deep learning task, storage access operations were extremely frequent, with an average of 2000 accesses, including 1600 accesses to main memory. After optimization, the number of storage accesses decreased to 1000, and the number of main memory accesses was reduced to 600. The proportion of main memory accesses dropped from 80% to 60%.

In terms of cache hit rate, the cache hit rate was only 20% without optimization. However, the collaborative optimization strategy ensures the accuracy and validity of data in the cache by introducing data prefetching technology and cache consistency maintenance technology, avoiding calculation errors caused by data

inconsistency issues. The cache hit rate is increased to 50%, with an increase of up to 150%, which reduces storage access latency and bandwidth usage, effectively improves the utilization efficiency of storage resources, and optimizes the overall performance of the chip.

The collaborative optimization scheme in this paper is compared and analyzed with NVIDIA's GPU architecture and Google's TPU architecture. In terms of computing performance, when processing image recognition tasks with intensive convolution operations, this scheme is comparable to or even better than NVIDIA's GPU architecture, but the GPU has a完善 software ecosystem and stronger versatility; compared with Google's TPU architecture, this scheme is more flexible, but the TPU is more efficient in the optimization of specific frameworks and tasks. In terms of energy consumption, this scheme has obvious advantages over the GPU architecture with lower energy consumption; compared with the TPU architecture, the energy consumption performance is comparable, with differences in energy consumption under different application scenarios. In terms of storage utilization, this scheme optimizes the storage hierarchy and data access strategy, so the storage resource utilization is better than existing schemes, but in terms of storage bandwidth utilization efficiency, it is not as good as schemes specifically optimized for storage bandwidth.

7. CONCLUSION

This paper achieves collaborative optimization of data paths and storage hierarchies for deep learning acceleration chip architectures, and obtains a series of research results of great value. In data path optimization, based on the data characteristics of deep learning algorithms, optimization strategies are proposed. By adopting a new architecture and parallel computing units, data processing and transmission efficiency are improved; the transmission bus is optimized to effectively reduce data transmission latency and ensure the efficient operation of deep learning tasks. In terms of storage hierarchy optimization, the performance differences of each level are studied, and a reasonable architecture is constructed. By optimizing cache parameters and introducing technologies such as high-bandwidth memory and NVM, data storage and access efficiency is improved, storage conflicts are reduced, and the strict requirements of deep learning for data storage are met. The collaborative mechanism between data paths and storage hierarchies is explored, and an adaptive collaborative optimization method based on dynamic task characteristics is proposed. This method can intelligently adjust working modes and resources according to tasks and operating status to achieve in-depth collaboration between the two. The constructed collaborative optimization model verifies the effectiveness of the method. Experiments are verified using AlexNet and LSTM models. The results show that the chip architecture after collaborative optimization has significantly improved computing performance, greatly shortened training and inference time, reduced energy consumption, and improved storage utilization, with good application prospects.

REFERENCES

[1] Chen, Yuyan, et al. "Emotionqueen: A benchmark for evaluating empathy of large language models." arXiv preprint arXiv:2409.13359 (2024).
[2] Moukheiber, Dana, et al. "A multimodal framework for extraction and fusion of satellite images and public health data." Scientific Data 11.1 (2024): 634.
[3] Restrepo, David, et al. "Multi-OphthaLingua: A Multilingual Benchmark for Assessing and Debiasing LLM Ophthalmological QA in LMICs." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 39. No. 27. 2025.
[4] Thao, Phan Nguyen Minh, et al. "Medfuse: Multimodal ehr data fusion with masked lab-test modeling and large language models." Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2024.
[5] Restrepo, David, et al. "Representation Learning of Lab Values via Masked AutoEncoder." arXiv preprint arXiv:2501.02648 (2025).
[6] Lin, Z., Wang, Y., & Hong, Y. (2023). The computing of the Poisson multinomial distribution and applications in ecological inference and machine learning. Computational Statistics, 38(4), 1851-1877.
[7] Gong, Z., Zhang, H., Yang, H., Liu, F., & Luo, F. (2023). A Review of Neural Network Lightweighting Techniques. Innovation & Technology Advances, 1(2), 1–24. https://doi.org/10.61187/ita.v1i2.36
[8] Junxi, Y., Wang, Z., & Chen, C. (2024). GCN-MF: A graph convolutional network based on matrix factorization for recommendation. Innovation & Technology Advances, 2(1), 14–26. https://doi.org/10.61187/ita.v2i1.30

[9] Zhang, X. (2024). Research on Dynamic Adaptation of Supply and Demand of Power Emergency Materials based on Cohesive Hierarchical Clustering. Innovation & Technology Advances, 2(2), 59–75. https://doi.org/10.61187/ita.v2i2.135
[10] Lin, Z., Liu, X., Xiang, Y., & Hong, Y. (2025). Modeling multivariate degradation data with dynamic covariates under a Bayesian framework. Reliability Engineering & System Safety, 111115.
[11] Yi, X. (2025, October). Real-Time Fair-Exposure Ad Allocation for SMBs and Underserved Creators via Contextual Bandits-with-Knapsacks. In Proceedings of the 2025 2nd International Conference on Digital Economy and Computer Science (pp. 1602-1607).
[12] Tang, Y., Kojima, K., Gotoda, M., Nishikawa, S., Hayashi, S., Koike-Akino, T., ... & Klamkin, J. (2020). Design and Optimization of Shallow-Angle Grating Coupler for Vertical Emission from Indium Phosphide Devices.
[13] Deng, X. (2025). Homomorphic Encryption-Based Data Integrity Verification and Anti-Tampering Mechanism in Cloud Storage Environment.
[14] Mehta, R., Patwar, N., Wei, X., Saunders, E., Zhu, X., & Liu, J. (2026). Towards a National AI Security Framework for Financial Infrastructure Protection. International Journal of Advance in Applied Science Research, 5(2), 39–50. Retrieved from https://h-tsp.com/index.php/ijaasr/article/view/251
[15] Zhou, Z. (2026). Hierarchical Needs in US Automotive Customer Feedback and the Sentiment–Function Nexus. Journal of Industrial Engineering and Applied Science, 4(1), 27-33.
[16] Wensi, L. (2026). AI-Enabled Data Visualization Marketing for Automated Production Lines: Building Customer Trust and Improving Lead-to-Order Conversion. Academic Journal of Natural Science, 3(1), 8-13.