Dean&Francis

ISSN 2959-6157

Design and Optimization of Hardware Accelerators for Convolutional Neural Networks

Jialun Liang[1, *]

[1] Department of Telecommunications, University of New South Wales, Sydney, Australia

*Corresponding author: jialun.liang@student.unsw.edu.au

Abstract:

With the expansion of Convolutional Neural Networks (CNNs) applications in various domains such as autonomous driving and real-time data processing, the demand for efficient computational resources has increased dramatically. Traditional computing platforms such as CPUs struggle to manage the complex and data-intensive tasks required by modern CNNs. This paper delves into the development and system optimization of dedicated hardware accelerators - GPUs, FPGAs, and ASICs to meet these demands. Through innovative architectural design and optimization techniques, these gas pedals improve computational speed and energy efficiency. Our study demonstrates that through these optimizations, the processing efficiency of hardware accelerators is significantly improved while energy consumption is effectively controlled, setting a new standard for the design of future hardware accelerators for convolutional neural networks. In addition, I discuss practical applications and future challenges, providing a comprehensive overview of current technologies and their potential development. This research highlights the critical role of advanced hardware accelerators in enabling the next generation of AI applications, ensuring technological advancement and sustainability in high-demand computing environments.

Keywords: Convolutional neural networks; hardware accelerators; computational efficiency; energy consumption; system optimization.

1. Introduction

Convolutional neural networks (CNNs), a powerful deep learning architecture, have been widely used in a variety of fields such as image and video analysis, natural language processing, and autonomous driving [1][2]. As technology continues to advance and application areas expand, CNN models become increasingly complex, and the amount of data and computation required to process these models in-

creases dramatically [3][4]. This increase brings about the need for higher computational power and more efficient data processing. Traditional computing platforms, such as general-purpose CPUs, are equipped with some processing power, but they often appear to be overwhelmed when dealing with complex CNN tasks. In addition, energy-efficiency ratio becomes an important issue, especially in data centers and cloud computing environments where large-scale models need to run continuously. Therefore, designing dedicated hardware accelerators such as GPUs, FPGAs, and ASICs that not only provide the necessary computational resources, but also enhance the processing speed and energy efficiency by optimizing specific computational operations, is crucial to meet the demands of modern deep learning tasks. Hardware accelerators are able to provide parallel processing capabilities, which makes them particularly suitable for performing large-scale matrix and vector operations in CNNs. In addition, these devices can be optimized at the hardware level to further improve computational efficiency. For example, operational costs and environmental impact can be reduced through designs that optimize data flow and reduce energy consumption. The use of hardware accelerators in devices from smartphones to servers helps enterprises and researchers tackle real-time data processing and complex model training challenges.

This paper will first introduce the basic concepts of CNN hardware accelerators and their working principles, and then discuss in detail the design and optimization strategies of different types of hardware accelerators, as well as their advantages in terms of performance and energy efficiency. Finally, the challenges faced by current technologies and future trends will be discussed, aiming to provide researchers and engineers with a comprehensive perspective on the design and optimization of CNN hardware accelerators.

2. Basic Concepts and Working Principles of Hardware Accelerators

2.1 Classification and Basic Concepts of Hardware Accelerators

2.1.1 Graphics processing unit(GPU)

Originally designed to process graphics and image data, GPUs are now widely used in areas such as scientific computing and deep learning due to their highly parallel processing architecture[5]. Capable of processing large amounts of data simultaneously, GPUs are ideally suited to perform complex matrix and vector operations, which are essential for training and reasoning about neural networks.

GPUs have been used in a wide range of scientific computing applications such as molecular dynamics simulations, astrophysics, and climate modeling, and can dramatically increase the speed of complex calculations in these fields. Meanwhile, when training deep neural networks, GPUs can provide the ability to process large amounts of data in parallel, significantly reducing training time. In video games and virtual reality, GPUs can render high-quality graphics in real time.

2.1.2 Field programmable gate array(FPGA)

FPGAs are devices that can be reconfigured at the hardware level, allowing users to customize logic gates to specific needs. Compared to GPUs, FPGAs offer greater flexibility and energy efficiency and are especially suited for applications that require specific hardware optimizations. FPGAs excel at tasks that are latency-sensitive or require high throughput, such as real-time data processing and complex control systems[6],[7].

FPGAs are often used for signal processing in wireless communication systems, such as the implementation of LTE and 5G communication protocols. In addition, FPGAs are used for high-speed image data processing and real-time analysis in medical imaging and security monitoring systems. In financial computing, FPGAs enable extremely fast data processing and decision making during high-frequency trading.

2.1.3 Application-specific integrated circuit(ASIC)

ASICs are chips that are specifically designed for a particular application and are therefore extremely efficient and performant in performing specific tasks. Although development costs and time are higher than GPUs and FPGAs, ASICs typically operate far more efficiently and perform far better in terms of power consumption than other types of hardware accelerators. They are commonly used in high-performance computing and large-scale data centers [8].

ASICs are the most commonly used hardware for cryptocurrency mining such as Bitcoin, due to their superior energy efficiency ratio and processing power. In addition, in smartphones and tablets, ASICs are used to handle specialized tasks such as image processing and audio processing to optimize battery usage efficiency. ASICs are used in autonomous driving technology to process complex sensor data and support real-time decision making.

2.2 Working Principle

Hardware accelerators greatly increase the speed of data processing through parallel processing. For example, GPUs dramatically speed up data operations by processing

ISSN 2959-6157

a large number of operations in parallel with thousands of small processing cores, which is particularly suitable for performing repetitive and independent tasks such as convolutional operations. FPGAs and ASICs, on the other hand, can be optimized at the hardware level for specific application requirements, such as customizing data paths and pipeline structures to optimize execution efficiency and response time. In addition to parallel processing, these hardware accelerators improve efficiency by optimizing data management. They employ efficient data caching and transfer strategies to reduce data access latency and bandwidth bottlenecks, enabling efficient data throughput to be maintained when processing large-scale neural networks.

By combining these hardware-specific parallel processing capabilities and data management optimizations, GPUs, FPGAs, and ASICs can significantly increase the processing speed and efficiency of CNN models, especially when performing complex image and video analysis tasks.

3. Key Technologies for Hardware Accelerator Design

3.1 Architectural Design

Architecture design is key when designing hardware accelerators for processing convolutional neural networks (CNNs). Different types of hardware accelerators (e.g., GPUs, FPGAs, ASICs) have significantly different architectural designs.

GPUs use a massively parallel processing architecture that is capable of performing a large number of floating-point operations simultaneously, which is ideal for data-parallel operations such as convolutional layer processing in CNNs. FPGA architectures can be customized to support pipelined processing and parallel execution, which makes FPGAs very flexible in processing CNNs with complex control logic. ASICs are designed specifically for specific computational tasks and can implement highly optimized pipelines and specific memory management strategies, which greatly improves the exclusivity and efficiency of operations.

3.2 Performance Optimization Strategies

3.2.1 Algorithmic hardware co-design

Co-design of algorithms and hardware involves adapting and optimizing the algorithms according to the specific characteristics of the hardware, so as to achieve the purpose of improving the execution efficiency of the hardware. For example, more efficient data processing and computation can be achieved by simplifying the computational complexity of the algorithms or realigning the algorithm structure to better fit the hardware architecture. Such optimizations not only allow the hardware to operate more efficiently, but also help to reduce energy consumption and extend the life of the hardware.

3.2.2 Parallelism optimization

Another important direction for performance optimization is parallelism optimization. Hardware resources can be better utilized by improving the hardware's parallel processing capabilities, such as adding more parallel execution units or improving parallel algorithms. Such optimizations allow hardware accelerators to handle more tasks or data streams simultaneously, significantly improving overall processing speed and efficiency. Parallelism optimization is a key component of hardware accelerator design and is particularly important for large and complex computational tasks.

With these focused efforts, hardware accelerator performance optimization can further improve the overall performance and responsiveness of the hardware system while maintaining operational efficiency.

3.3 Energy Efficiency Optimization Measures

3.3.1 Dynamic voltage and frequency scaling(DVFS)

Dynamic Voltage Frequency Scaling (DVFS) is an energy-efficiency optimization technique widely used in hardware design. It allows the system to dynamically adjust the processor voltage and frequency based on the current load. By reducing the processor's voltage and frequency when the load is low, power consumption can be significantly reduced, while restoring normal levels when high performance is required to ensure timely completion of computational tasks. This approach not only conserves energy, but also reduces system heat and extends equipment life[9][10].

3.3.2 Energy efficient computing architecture

A critical step in the design of low-power component architectures is the use of low-power processing units and memory components. These low-power components are specifically designed to minimize overall system power consumption without sacrificing performance. Selecting processors that can run at lower voltages and frequencies while maintaining good performance is one of the key ways to achieve a low-power system design. At the same time, the use of advanced memory technologies, such as LPDDR (Low Power Double Data Rate Memory), can further reduce the energy consumption of the system while it is running.

In addition, optimizing data transfer paths is an effective means of reducing energy consumption. Energy consumption can be significantly reduced by carefully de-

signing paths for data flow and reducing redundant data operations. For example, implementing more efficient caching strategies can reduce unnecessary data exchanges between the processor and memory, thereby reducing the demand for energy. Meanwhile, adopting data compression techniques can reduce the amount of data that needs to be transferred and stored, further optimizing memory usage efficiency and reducing energy consumption. These strategies not only improve data processing efficiency, but also significantly reduce power consumption, which is in line with the current trend of green and environmentally friendly technology development.

By implementing these strategies, hardware designers are able to provide high-performance, environmentally friendly solutions for modern computing devices that meet increasingly stringent energy efficiency standards and market demands. The adoption of this "green computing architecture" is helping to move the industry toward a more sustainable future.

3.3.3 Hardware acceleration specialized module

The design concept of dedicated low-energy modules is to integrate low-power hardware modules dedicated to specific tasks at the hardware level. By specifically optimizing the computing processes for these tasks, these modules are able to significantly reduce energy consumption while maintaining high performance. Compared to general-purpose processors, such dedicated modules can use energy more efficiently and consume less energy on unnecessary computations because they are optimized directly for specific tasks.

Another energy-efficient design approach is "on-demand activation". In this design model, certain parts of the hardware are activated only when they are needed and remain off when they are not. This strategy effectively reduces energy consumption when the hardware is not in operation. For example, in a processor, processing units that are not involved in the current task can be turned off so that only the parts that are involved in computing consume power. This not only reduces the overall energy consumption but also extends the lifetime of the device.

With these integrated energy-efficiency optimization modules and on-demand activation design methodologies, ASICs and FPGAs can achieve higher energy efficiency when performing complex data processing tasks, which is especially important for energy-sensitive application scenarios. Such optimization not only helps protect the environment, but also reduces operating costs and creates greater value for businesses and users.

3.3.4 Support at the software level

Optimizing hardware energy efficiency also requires support from within software, including the development of specialized tools and algorithms to ensure efficient operation of the overall system:

Developing specialized compilers that can optimize code execution paths to reduce CPU cycles and lower energy consumption. Such compilers help reduce processor power consumption by intelligently rearranging and optimizing instructions and reducing unnecessary calculations, thus enhancing overall system energy efficiency.

Intelligent scheduling algorithms implement system software capable of intelligently scheduling tasks based on the priority of the task and the current state of the processor. Such intelligent scheduling algorithms ensure that high-priority tasks receive the necessary processing power by efficiently allocating processor resources, while minimizing energy consumption without compromising performance.

These energy-efficiency optimization measures work together to reduce overall system energy consumption and improve the energy-efficiency-to-performance ratio through a combination of innovative hardware design and intelligent software support. Adopting these strategies not only saves energy, but also helps to minimize the impact on the environment, in line with current global goals for sustainable development. Such an integrated optimization strategy ensures a balance between technical efficiency and ecological responsibility, driving further development of green technologies and energy-saving measures.

4. Practical Application Examples of Hardware Accelerators

4.1 Practical Applications of Hardware Accelerators in Self-driving cars

With the rapid development of autonomous driving technology, the need for efficient computing resources that can process large amounts of sensor data in real time is growing. Self-driving cars must be able to rapidly process data from a wide range of sensors such as video cameras, radar and lidar. These data processing tasks involve complex image and signal processing algorithms that place extreme demands on computational speed and processing power. Conventional central processing units (CPUs) are no longer able to meet these requirements, making hardware accelerators an indispensable component of autonomous driving technology.

4.2 Selection and Application of Hardware Accelerators

Graphics Processing Units (GPUs) and Field Programma-

ISSN 2959-6157

ble Gate Arrays (FPGAs) are the most common hardware accelerators in the field of autonomous driving. For example, Tesla Motors uses its own Full Self-Driving (FSD) chip, which is designed based on Application Specific Integrated Circuit (ASIC) technology. Such hardware accelerators were developed specifically to process sensor data for self-driving cars, and they offer higher processing speeds and lower energy consumption than traditional CPUs.

Tesla's FSD chip has the ability to process 2,300 frames of video data per second, compared to 210 frames for standard hardware under the same conditions. This significant performance increase allows the car to ensure driving safety with faster response times.

4.3 Advantages of Hardware Accelerators in the Case

Hardware accelerators can dramatically reduce data processing time by processing large amounts of data in parallel, enabling autonomous driving systems to respond instantly to changes in the environment and improve response times. Hardware accelerators designed for specific tasks are more energy efficient than traditional central processing units (CPUs) in performing those tasks. This feature is particularly important for electric vehicles, as energy efficiency directly affects the vehicle's range, thereby reducing energy consumption. Efficient data processing is crucial for autonomous driving systems. It enables these systems to quickly and accurately recognize complex road conditions. As a result, this significantly reduces the risk of traffic accidents and enhances system safety.

5. Conclusion

There are some key challenges in the design and optimization of hardware accelerators, such as the increasing complexity of the technology that requires higher design and maintenance costs, as well as the increasing market demand for high performance and efficiency, which makes it more difficult to develop general-purpose solutions to fit various application scenarios. In addition, the widespread adoption of hardware accelerators is also limited by high initial investment costs and a shortage of specialized technical staff.

Looking to the future, hardware accelerators technology is expected to evolve in a more intelligent and adaptive direction. With advances in artificial intelligence and machine learning technologies, future hardware accelerators will integrate more intelligent decision-making capabilities to achieve higher operational efficiency and self-optimization capabilities. At the same time, as technology matures and production costs decrease, this advanced hardware will be more widely used in various industries, driving innovation and development across the industry.

References

[1] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, Optimizing FPGA-based accelerator design for deep convolutional neural networks, in Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays, 2015.

[2] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, XNOR-Net: ImageNet classification using binary convolutional neural networks,in Proc. Eur. Conf. Comput. Vision (ECCV), 2016.

[3] S. Han, H. Mao, and W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding, in Proc. Int. Conf. Learn. Representations, 2016.

[4] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning, ACM Sigplan Notices, 2014, 49(4): 269-284.

[5] Wong, H., Papadopoulou, M.-M., Sadooghi-Alvandi, M., & Moshovos, A. (2010). Demystifying GPU microarchitecture through microbenchmarking, in Proc. IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS), 2010.

[6] C. Farabet, C. Poulet, J. Han, and Y. LeCun, «CNV: FPGA-based convolutional networks for machine vision applications, in Proc. 21st ACM/SIGDA Int. Symp. Field Programmable Gate Arrays, 2011.

[7] M. Cho and Y. Kim, FPGA-based convolutional neural network accelerator with resource-optimized approximate multiply-accumulate unit, Electronics, 2021,10(22) :2859.

[8] Z. Liu et al., A high-efficiency ASIC accelerator for convolutional neural networks, IEEE Trans. VLSI Syst., 2021,29(12) :2810-2822

[9] Sheth, A., Doerr, C., Grunwald, D., Han, R., Sicker, D. Understanding and mitigating the impact of RF interference on 802.11 networks,» in Proc. ACM Int. Conf. Measurement and Modeling of Comput. Syst. (SIGMETRICS), 2008.

[10] Fan, X., Weber, W.-D., Barroso, L. A. Power provisioning for a warehouse-sized computer,in Proc. Annu. Int. Symp. Comput. Archit. (ISCA), 2007.