Dean&Francis
ISSN 2959-409X

Research on the Application of Hardware Accelerators in Artificial Intelligence Systems

Shuqi Zhu[1, *]

[1]Department of Electrical and Electronic Engineering, University of Birmingham, Wenzhou, China
*Corresponding author: SXZ467@student.bham.ac.uk

Abstract:

Traditional general-purpose processors are constrained by operational efficiency when handling computationally intensive tasks such as deep learning, convolutional neural networks, and recurrent neural networks. Hardware accelerators, as specialized computing architectures, significantly outperform general-purpose processors in parallel computing, data throughput, and operational efficiency. Therefore, hardware accelerators have become one of the key factors driving the advancement of AI. This article systematically studies the basic principles of hardware accelerators, their main types (GPU, FPGA, ASIC), as well as the design and implementation methods of hardware accelerators. By comparing the performance metrics, power consumption characteristics and application adaptability of different accelerator architectures, the advantages of hardware accelerators over other accelerators are demonstrated. The paper also investigates the advantages and challenges of hardware accelerators in accelerating deep learning inference, training, and edge computing. According to the research results, accelerators tailored for specific AI tasks can significantly reduce latency and improve energy efficiency, and have broad application prospects in scenarios such as 5G, autonomous driving, and intelligent manufacturing. This research provides a reference for the designers of artificial intelligence systems to select and optimize hardware acceleration solutions, and also offers a direction for the future innovation of accelerator architectures.

Keywords: Hardware Accelerators, Artificial Intelligence Systems, Applied Research

ISSN 2959-409X

1. Introduction

Artificial intelligence technology is now widely applied in fields such as healthcare, autonomous driving, financial analysis, and smart manufacturing. These applications typically involve large-scale data processing and highly complex model computations [1]. Traditional CPUs are gradually being phased out as they struggle to meet the high parallel computing demands required by deep neural networks when executing such tasks. Hardware accelerators such as GPUs, FPGAs, and ASICs, with their high parallelism, low latency, and customizability, will replace CPUs as the core computational units in AI systems [2]. Today, the parameter sizes of numerous deep learning models (such as Transformers and the GPT series) are growing exponentially demanding processors with high computational power and low energy consumption. Hardware accelerators not only support training on servers but also enable efficient inference on edge devices, while maintaining relatively low power consumption, making them highly suitable for such learning models [3]. Early hardware accelerators designed for deep learning training primarily focused on optimizing GPU performance. For instance, NVIDIA's CUDA platform enabled GPUs to excel in matrix operations and convolution computations [4]. FPGAs leverage their reconfigurable nature to achieve a balance between power consumption and latency in specific model inference tasks [5]. Meanwhile, ASICs, exemplified by Google's TPU, have undergone deep optimization for matrix multiplication and activation function computations [6]. Existing research indicates that different accelerator types offer distinct advantages in computational density, energy efficiency, and programmability. However, determining the optimal choice for various application scenarios remains a current research hotspot [7].

2. Theoretical Foundations

2.1 Definition and Classification of Hardware Accelerators

Hardware accelerator is a dedicated hardware device specifically optimized for certain computing tasks, aiming to achieve stronger computing performance than general-purpose processors through customized architecture. It is mainly divided into three categories: GPU, FPGA, and ASIC.The GPU was originally designed for graphics rendering. With its parallel computing capabilities of thousands of small cores, it is widely used in computationally intensive tasks such as image processing, matrix operations, and deep learning. As reconfigurable hardware, FPGA enables users to customize hardware functions. It offers advantages of high performance, high energy efficiency and low latency in specific tasks, and is particularly suitable for real-time response scenarios such as edge computing. ASIC refers to an integrated circuit specifically tailored for a certain computing task, which can provide the highest performance and energy efficiency in computationally intensive operations. For instance, Google's tensor processing unit, which is optimized for deep learning, significantly reduces computational and memory access bottlenecks through hardware-level optimization.

2.2 Basic Principles of Hardware Accelerators

Hardware accelerators optimize specific computational tasks through dedicated circuitry, enabling them to execute data-intensive operations such as matrix operations and convolution computations more efficiently than CPUs [8].The fundamental principles of hardware accelerators encompass four aspects: parallel computing, specialized compute units, memory optimization, and low-precision computation.In parallel computing, hardware accelerators such as GPU, FPGA, and ASIC significantly boost computational speed by increasing the number of processing units that handle data in parallel. For instance, GPU leverage thousands of small processing cores to execute computational tasks simultaneously, enabling large-scale parallelization of matrix multiplication and convolution operations in deep learning. This accelerates both training and inference processes.

In mathematical models, the hardware acceleration process can be abstracted as:

$$T_{CPU} = \frac{O\left(n^3\right)}{f_{CPU}} \ , \ T_{ACC} = \frac{O\left(n^3\right)}{f_{ACC*p}} \tag{1}$$

Here, T_{CPU} and T_{ACC} represent the execution times for the CPU and accelerator, respectively, f denotes the clock frequency, and p indicates the degree of parallelism. Clearly, under high parallelism conditions, $T_{ACC} \ll T_{CPU}$.

Secondly, hardware accelerators such as FPGAs and ASICs execute specific tasks through custom hardware computing units. These units can provide specialized optimizations for particular algorithms—such as convolutional neural networks—thereby achieving optimal results in both energy efficiency and computational speed. For example, Google's Tensor Processing Unit is an ASIC optimized for deep learning tasks, accelerating matrix operations through specialized hardware design.It is worth noting that memory bandwidth and access efficiency are critical to the performance of hardware accelerators. GPUs, FPGAs, and ASICs all employ specialized memory architectures and optimized algorithms to reduce memory

bottlenecks.GPUs reduce memory access latency through shared memory and cache mechanisms, while FPGAs optimize data transfer via custom data paths and memory access methods. These optimizations accelerate data-intensive tasks and minimize computational latency.

With the advancement of deep learning algorithms, low-precision computing—such as using FP16, INT8, and similar formats—has been widely adopted to accelerate neural network inference processes.Low-precision computation can significantly reduce computational load and memory bandwidth requirements. Moreover, in most applications, the accuracy loss incurred by low-precision computation has a negligible impact on results.Hardware accelerators (such as TPUs and NVIDIA's Tensor Cores) significantly enhance energy efficiency and processing speed through hardware optimizations for low-precision computations.

2.3 Performance Evaluation Metrics

The performance evaluation of hardware accelerators typically covers key metrics such as throughput, latency, energy efficiency, flexibility and computational efficiency. Throughput refers to the amount of data processed within a unit of time, usually measured in terms of the number of images processed per second or the number of inference requests. It is an important indicator for evaluating processing capabilities, and it significantly impacts the response speed and overall efficiency of large-scale data processing and real-time inference. Latency refers to the time required to complete a single task. It is crucial in real-time systems such as autonomous driving. Accelerators like FPGAs effectively reduce latency by performing parallel processing and customizing data paths. Energy efficiency reflects the computing performance provided by each watt of power consumption, measured in trillion operations per second per watt. It is crucial for embedded devices and edge computing, enabling power reduction and extended battery life. GPUs and ASICs demonstrate outstanding energy efficiency in data center environments. Flexibility reflects the accelerator's ability to adapt to multiple tasks and algorithm updates. GPUs, with their support for both software and hardware collaboration, can handle various models and thus have a high degree of flexibility. In contrast, FPGAs and ASICs are highly computationally efficient but require reconfiguration or redesign to accommodate new requirements. Computational efficiency refers to the ratio of actual performance to theoretical peak performance, and is used to evaluate the performance of hardware in actual tasks. For example, TPU and GPU have high computational efficiency in deep learning, but may be limited by bandwidth in memory-intensive operations.

2.4 An Overview of the Development of Hardware Accelerators

The hardware accelerator has undergone continuous development from general computing to dedicated architectures and then to heterogeneous integration, closely linked to the rapid iteration of artificial intelligence technology. Initially, GPUs were mainly used for graphics rendering while CPUs were used for computations. However, as the scale of deep learning models rapidly expanded, traditional CPUs gradually failed to meet the demands of large-scale parallel computing. At this point, the GPU, with the help of the CUDA programming framework, was endowed with general computing capabilities, thus becoming the main platform for deep learning training. By 2006, NVIDIA had launched the CUDA platform, enabling developers to directly access the GPU and perform large-scale parallel core operations for matrix calculations and convolution computations. This has shortened the training time of deep learning from weeks to days, accelerating and popularizing the research of artificial intelligence. As AI applications move towards edge devices and embedded scenarios, energy efficiency and low latency have gradually become constraints. At this point, FPGA is widely used for inference acceleration due to its hardware reconfigurable feature. By means of customized data paths and pipeline design, FPGA demonstrates high energy efficiency and low latency in real-time tasks such as object detection and speech recognition. Through this, Microsoft's Project Brainwave also utilized FPGA clusters to achieve online inference services. However, the development of FPGAs has a high threshold and a long design cycle, which still imposes certain limitations on their large-scale deployment. To further enhance its performance and reduce its power consumption, some companies have developed application-specific integrated circuits. Among the products developed in this research are Google TPU, the Hummingbird chip from HuaWei, and the Ascend series from HuaWei. These products directly implement matrix multiplication, activation functions, and convolution operations as hardware circuits in deep learning, significantly increasing throughput and reducing energy consumption, and becoming the core computing units for cloud-based inference and large-scale training. Similarly, as the complexity of AI models and the diversity of application scenarios increase, a single architecture is no longer sufficient to meet all requirements. Some industries have gradually adapted to the development trend of heterogeneous computing and high integration, by deploying CPUs, GPUs, FPGAs and ASICs on the same platform in a coordinated

ISSN 2959-409X

manner, to achieve unified scheduling of computing, storage and communication resources. This approach not only retains the flexibility of general processing but also takes into account the high performance of dedicated computing. In the future, with the development of technologies such as on-chip networks, Chiplets, and 3D packaging, hardware accelerators will achieve higher integration and modularization goals, enabling functions such as on-demand combination and dynamic optimization. This will allow artificial intelligence systems to achieve a more ideal balance between performance, energy efficiency, and scalability.

3. Case Study

3.1 Applications of GPUs in Deep Learning

Training

The parallel computing capability of GPUs is one of the core reasons they have become the preferred hardware for deep learning training. Unlike the sequential computation of CPUs, GPUs process different computational tasks simultaneously using hundreds or thousands of small computational units. This design is particularly well-suited for computationally intensive tasks such as matrix operations and convolution operations. Deep learning models, particularly those like convolutional neural networks and recurrent neural networks , commonly involve extensive matrix multiplication and element-wise computation tasks. These operations can achieve significantly accelerated processing speeds through highly parallelized execution on GPUs.

Table 1. CPU Platform and GPU Platform Configurations (Data Source: [9])

	CPU platform	GPU platform
Model	I5 3570K	Tesla K20
Memory	16 GB DDRIII	5 GB GDDR5
Software	Matlab 2012a R27	CUDA 5.5
Library	Intel(R) MKL V10.3.5	cuBLAS V2

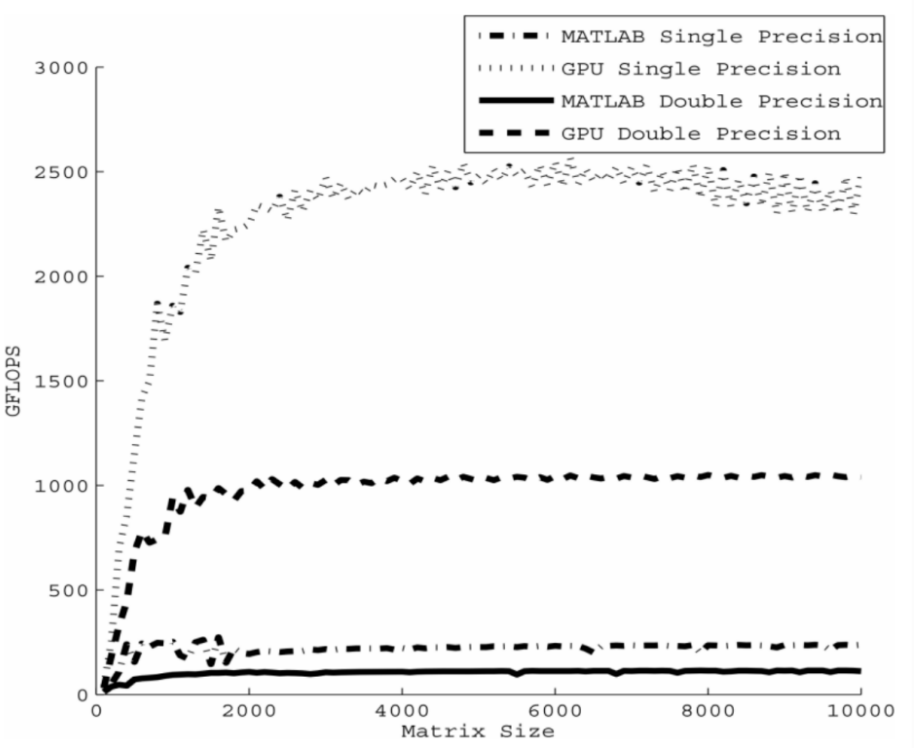

Fig. 1 GFLOPS evaluation of n×n matrix multiplication using single-precision and double-precision floating-point operations (Data source:[9])

Table 2. MNIST: Average execution time across three components within one learning cycle (Data source: [9])

	FP	BP	WU	Overall
CPU	7.4648	3.8836	3.3239	14.6724
GPU	0.6126	0.4014	0.2981	1.3120
Speedup	12.19	9.68	11.15	11.18

Relevant experiments have demonstrated that GPU acceleration significantly outperforms CPU acceleration. This experiment utilizes the MNIST handwritten digit database to evaluate acceleration performance. Table 1 lists the configurations of the CPU and GPU platforms employed in the simulation. Figure 1 illustrates the GFLOPS results when multiplying matrices of varying sizes using Matlab on the CPU and cuBLAS on the GPU. Table 2 records the average time required for each of the three components within each cycle, with time measured in seconds. The final row of the table shows the acceleration ratio of GPU relative to CPU. It is clearly evident that GPU-based computation achieves speeds far exceeding those of CPUs, ranging from 9 to 12 times faster. This confirms that GPU-based computational acceleration is an economical and efficient method for training deep learning networks.

Another significant advantage of GPUs is their high-bandwidth memory design (sliced flattened butterfly topology) [10], enabling rapid transfer of large data volumes. Deep learning models require frequent loading of data and intermediate results during training, and the enhanced memory bandwidth of GPUs makes these operations more efficient. Furthermore, GPUs optimize data transfer speeds through shared memory, registers, and caching mechanisms, further reducing computational bottlenecks.

3.2 Using FPGAs for AI Model Inference on Low-End Embedded Platforms

FPGAs are reconfigurable and possess parallel and real-time processing capabilities. Due to their extremely low power consumption, they are well-suited for low-end, power-constrained embedded platforms. Relevant experiments have compared Raspberry Pi with FPGAs for CNN model inference, with test results shown in Table 3.The test results confirm that the FPGA achieves significantly faster inference speeds compared to the Raspberry Pi. This is clearly demonstrated in Figure 2, where the CNN inference speed on the FPGA is 14 times faster than that on the Raspberry Pi. Furthermore, performing CNN inference on the FPGA substantially reduces power consumption, while the slight decrease in accuracy is negligible. This experiment sufficiently demonstrates the significant advantages of FPGAs in low-end, power-constrained embedded platforms. They can be applied to AI tasks requiring low-power real-time processing, such as autonomous driving and facial recognition.

Table 3. Independent Sample Test of Inference Time Achieved by Raspberry Pi and FPGA (Data Source: [11])

Test cases	Raspberry Pi			FPGA Implementation		
	Inference Time(ms)	Power Consumption(W)	Accuracy (%)	Inference Time(ms)	Power Consumption (W)	Accuracy(%)
1	45.4	4.5	98.5	3.2	2.5	97.0
2	44.8	4.4	98.8	3.1	2.4	97.3
3	46.0	4.6	97.0	3.3	2.6	97.1
4	45.5	4.3	97.3	3.0	2.3	97.2
5	44.9	4.5	97.6	3.2	2.4	97.4
6	46.2	4.7	97.1	3.1	2.5	96.9

ISSN 2959-409X

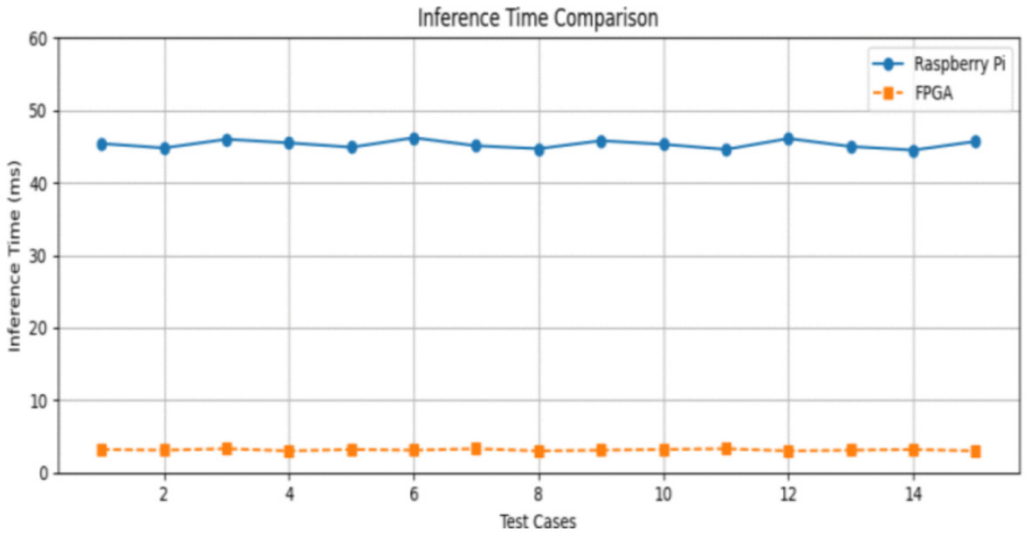

Fig. 2 shows the inference time comparison between the CNN on Raspberry Pi and that on FPGA. (Data source: [11])

3.3 Research on Hardware Accelerators in Different AI Application Scenarios

Hardware accelerators play a crucial role in various artificial intelligence applications. Different scenarios demand distinct levels of computational power, latency, and energy efficiency, driving continuous advancement in targeted research. In computer vision, convolutional neural networks and object detection models are widely used for image classification, semantic segmentation, and industrial inspection. Their core computations involve massive matrix operations and convolutional operations, placing extremely high demands on parallelism and memory bandwidth.Research indicates that GPUs and TPUs can significantly boost throughput and shorten training cycles for such tasks, reducing training time from weeks to days or even hours [9]. Simultaneously, to address power constraints on edge devices, researchers deployed lightweight CNNs—pruned and quantized—on FPGAs. By implementing customized data paths and pipeline optimizations, they achieved low-latency inference with over 10x acceleration gains in tasks like drone navigation and industrial defect detection [11]. In the field of natural language processing, large-scale language models represented by the Transformer, BERT, and GPT series impose increasingly stringent demands on computational and storage resources. Related research has proposed hybrid-precision training and operator fusion methods based on TPUs and ASICs, which not only enhance the utilization rate of matrix multiplication units but also reduce memory access overhead, achieving efficient scaling for large-scale distributed training [6]. Autonomous driving and robotics demand even stricter real-time performance, requiring perception, decision-making, and control to be completed within milliseconds. In medical and IoT scenarios, power consumption and size constraints are even more stringent. Existing work utilizes low-power FPGAs and dedicated ASICs to accelerate medical image analysis and real-time monitoring in wearable devices, reducing system power consumption by over 50% and significantly extending battery life. Overall, current research trends focus on two primary paths: first, enhancing hardware throughput and energy efficiency through high-bandwidth on-chip memory, low-precision computing units, and heterogeneous interconnects; second, achieving coordinated optimization of software and hardware via model compression, operator fusion, and dynamic scheduling to fully unlock hardware potential. Looking ahead, advancements in chiplets, on-chip networks, and adaptive computing architectures will enable hardware accelerators to achieve automatic optimization and resource scheduling across more scenarios. This will provide efficient, low-power computing support for next-generation artificial intelligence systems.

4. Limitations and Future Prospects

Hardware accelerators still have certain limitations. For instance, ASICs involve lengthy development cycles and high costs, and hardware accelerators require optimization based on the characteristics of different algorithms. There

are significant migration costs when dealing with different models. As AI models continue to expand, existing hardware accelerators may face bandwidth bottlenecks.

However, in the future, hardware accelerators can still leverage heterogeneous computing architectures to combine the strengths of CPUs, GPUs, FPGAs, and ASICs, forming highly efficient collaborative computing platforms. By utilizing reconfigurable computing, the adaptability of FPGA-like accelerators to algorithm updates can be enhanced, enabling them to meet computational demands while optimizing energy consumption control and reducing power usage.

5. Conclusion

This paper conducts a systematic study on the application of hardware accelerators in artificial intelligence systems, covering their fundamental principles, classification, performance evaluation, and typical application cases. Research indicates that accelerators customized or optimized for specific tasks can achieve a significant balance between performance and energy efficiency, demonstrating immense potential in areas such as deep learning training and AI model inference. Looking ahead, the integration of heterogeneous computing architectures and green computing principles will elevate hardware accelerators to a more pivotal role within the AI ecosystem. This research provides a reference for hardware selection and optimization in AI systems while laying the groundwork for continued innovation in accelerator architecture.

References

[1] Dally W J. High-performance hardware for machine learning[J]. Advances in Computers, 2020, 117: 1-70.

[2] Chen Y, Emer J, Sze V. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks[J]. IEEE Journal of Solid-State Circuits, 2017, 52(1): 127-138.

[3] Zhu J, Zhang W, Liu S, et al. Energy-efficient hardware accelerators for machine learning in embedded systems[J]. Electronics, 2022, 11(6): 945.

[4] Li Z, Zhang C, Chen X, et al. Hardware acceleration for deep neural networks: A survey[J]. Wireless Personal Communications, 2024, 133: 2665-2691.

[5] Sze V, Chen Y H, Yang T J, et al. Efficient processing of deep neural networks: A tutorial and survey[J]. Proceedings of the IEEE, 2017, 105(12): 2295-2329.

[6] Jouppi N P, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit[C]//Proceedings of the 44th Annual International Symposium on Computer Architecture. ACM, 2017: 1-12.

[7] Chen T, Du Z, Sun N, et al. DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning[C]//Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2014: 269-284.

[8] Jouppi N P, Yoon D H, Kurian G, et al. TPUv4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings[C]//Proceedings of the 50th Annual International Symposium on Computer Architecture. ACM, 2023: 1-14.

[9] Chen Zhilu, et al. „A fast deep learning system using GPU.“ 2014 IEEE international symposium on circuits and systems (ISCAS). IEEE, 2014.

[10] G. Kim, M. Lee, J. Jeong and J. Kim, „Multi-GPU System Design with Memory Networks,“ 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture, Cambridge, UK, 2014, pp. 484-495,

[11] Premalatha, S., et al. „FPGA Based AI Inference Accelerator for Low-End Embedded Systems.“ 2025 3rd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA). IEEE, 2025.