World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:14, No:12, 2020

A Survey of Field Programmable Gate Array-Based Convolutional Neural Network Accelerators

Wei Zhang

***Abstract*—**With the rapid development of deep learning, neural network and deep learning algorithms play a significant role in various practical applications. Due to the high accuracy and good performance, Convolutional Neural Networks (CNNs) especially have become a research hot spot in the past few years. However, the size of the networks becomes increasingly large scale due to the demands of the practical applications, which poses a significant challenge to construct a high-performance implementation of deep learning neural networks. Meanwhile, many of these application scenarios also have strict requirements on the performance and low-power consumption of hardware devices. Therefore, it is particularly critical to choose a moderate computing platform for hardware acceleration of CNNs. This article aimed to survey the recent advance in Field Programmable Gate Array (FPGA)-based acceleration of CNNs. Various designs and implementations of the accelerator based on FPGA under different devices and network models are overviewed, and the versions of Graphic Processing Units (GPUs), Application Specific Integrated Circuits (ASICs) and Digital Signal Processors (DSPs) are compared to present our own critical analysis and comments. Finally, we give a discussion on different perspectives of these acceleration and optimization methods on FPGA platforms to further explore the opportunities and challenges for future research. More helpfully, we give a prospect for future development of the FPGA-based accelerator.

***Keywords*—**Deep learning, field programmable gate array, FPGA, hardware acceleration, convolutional neural networks, CNN.

I. Introduction

WITH the rapid development of deep learning, it speeded up the development of machine learning and artificial intelligence. Especially, CNNs has been widely used in image recognition [1], image classification [2], [3], object detection [4], [5], voice recognition [6], and autonomous driving technology [7], [8]. As it has advantages of excellent performance and high accuracy, a great number of research organizations consequently have been studying CNNs all over the world in recent years [9]. However, as practical applications place higher demands on accuracy and complexity, the scale of neural networks has exploded. Meanwhile, many application scenarios have strict requirements on the performance, low-power consumption, and real-time of hardware devices. Many hardware platforms can hardly meet the performance requirement on data computation. What is more, the realization of high-performance deep learning networks with low power consumption brings huge challenges, especially for large-scale deep learning neural network models. Therefore, it is particularly important to choose an appropriate computing platform for neural network applications.

So far, the state-of-the-art means for accelerating deep learning algorithms are FPGA, ASIC, GPU, and DSPs. Among these approaches, FPGA-based accelerators have attracted more and more attention of researchers because they have advantages of good performance, high energy efficiency, fast development round, and capability of reconfiguration [10]-[13]. So the primary survey here is FPGA-based acceleration of CNNs in this article over the last decade.

A. Overview of Deep Learning

In 2006, deep learning leader Hinton [14] proposed the training method of unsupervised deep confidence network. In 2013, deep learning ranked first among the top 10 breakthrough technologies. By March 2016, Alpha-Go [15] defeated the master of human Go. The history of deep learning development is shown in Fig. 1. The open circle in Fig. 1 represents the key turning point for the rise and fall of the depth of learning. The size of the solid circle indicates the depth of deep learning in this year. The oblique upward line indicates that the depth learning heat is rising, and the oblique downward line indicates that the deep learning heat is in the falling period.

Deep learning is an important part of machine learning, which can reduce the loss rate, improve the accuracy and enhance the robustness. In the last few years, deep learning has led to very good performance on a variety of problems.

Yuqingyang [16] summarized the research progress of deep learning at the current stage as follows: 1) Improving network training skills and improving network performance; 2) Development of network system: jump connection structure, stack self-coding network system; 3) New Learning mode - semi-supervised deep learning; 4) Deep reinforcement learning - artificial intelligence decision algorithm for cross-domain integration. He has made a relatively good review of the progress of deep learning research, but it is not comprehensive enough. Further speaking, it lacks an overview of the deep learning algorithm development environment and the learning algorithm framework. If it can be combined with the development of the algorithm optimization process, it will be more detailed and better.

B. Overview of CNNs [18]

Artificial neural networks are a typical machine learning method and an important form of deep learning. From 1943 McCulloch and Pitts [19] first proposed artificial neuron models (M-P neurons, as shown in Fig. 2 (a)) to 1958 Rosenblatt [20] designed the perceptron (Fig. 2 (b)). As the

Wei Zhang is with the Qihoo360 and Yangzhou University, China (e-mail: yidazhang1@gmail.com).

number of layers of the perceptron increases, there are deep neural networks as multiple hidden layers increase (Fig. 2 (c), (d)). In order to solve the problem of long calculation time and high power consumption, Hubel and Wiesel [21] proposed a CNN in the 1960s (as shown in Fig. 3: a representative CNN architecture). The characteristics of local perception and parameter sharing of CNNs enable it to effectively reduce the number of parameters and reduce the complexity of deep neural networks.

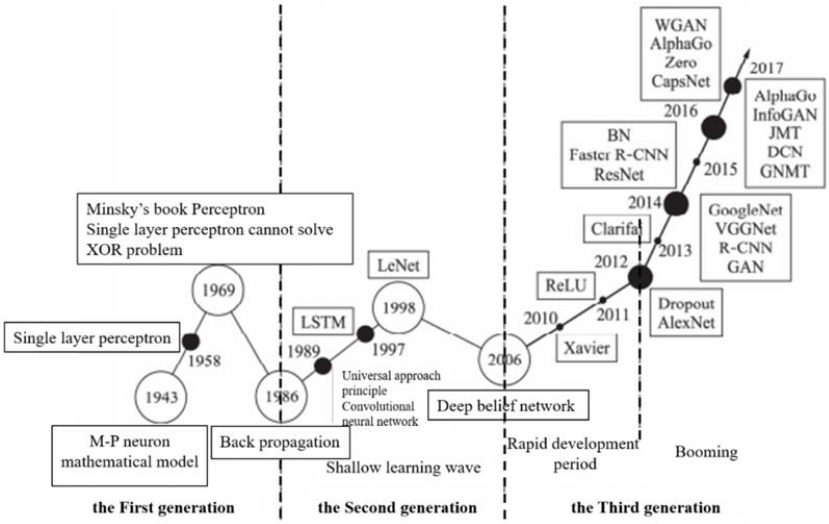


Fig. 1 The history of deep learning [17]

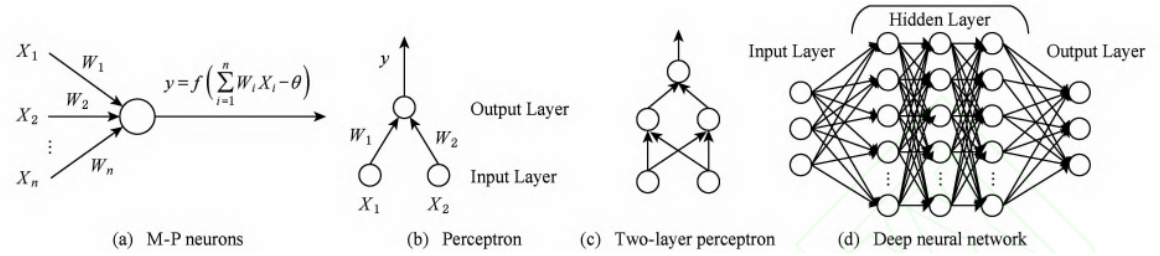


Fig. 2 The evolution of neural network [24]

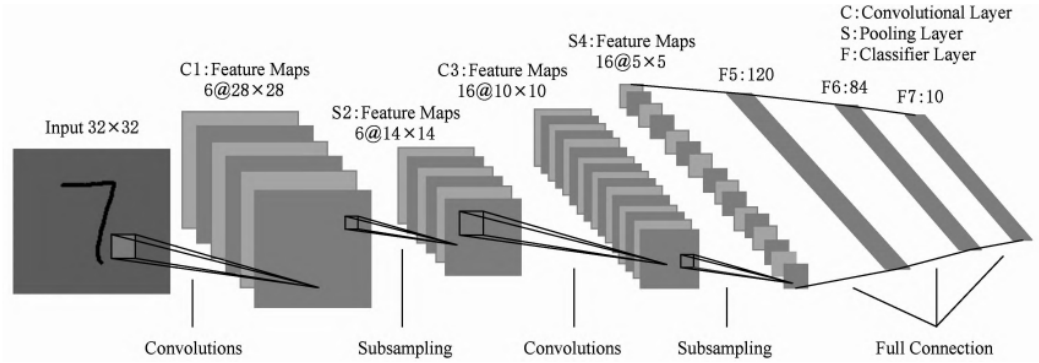


Fig. 3 A representative CNN architecture-LeNet5 [22]

CNN, as a well-known deep learning architecture extended from artificial neural network, has been extensively adopted in various applications, which include video surveillance, machine vision, image search engine in data centers, etc. [12], [23], [36].

C. Acceleration Methods of Deep Learning Algorithm

The acceleration of deep learning algorithms is mainly divided into software acceleration and hardware acceleration, but with the needs of practical applications, separate software and hardware acceleration is difficult to meet the practical application requirements. At present, hardware and software acceleration often develops synergistically. While improving and optimizing the deep learning algorithm architecture, it also improves the hardware platform, including software coordination and visual development of the development

process.

Major scientific research institutions have proposed their own accelerated structures, such as the DianNao family of the Tianshi Chen team [24]-[27] of the Chinese Academy of Sciences (based on RISC, a new dedicated instruction set architecture called Cambricon [28] is designed.), the TPU [29] (tensor process unit) and the Cloud TPU [30] of Google, the ScaleDeep [31] launched by Purdue University, the Eyeriss [32] proposed by MIT. The memristor-based ISAAC [33] proposed by the HP Laboratory and the University of Utah, and the compressed sparse convolutional neural network accelerator SCNN proposed by Parashar et al. [34], etc.

The research of existing neural network acceleration chips mainly focuses on four aspects [35]:

1) Starting from the computational structure of the neural network structure, it is studied how the tree structure and the array structure complete the convolution operation of the neural network in colleges and universities;
2) From the perspective of storage bottleneck, how to apply 3D storage technology to the design of the accelerator;
3) Starting from the exploration of new material devices, how to realize the integration of neural network processing and storage in new devices such as memristors;
4) From the perspective of data flow and optimization, how to maximize the partial reuse of various types of data in the network and the processing of sparse networks are studied.

II. Hardware Acceleration of CNNs [22]

The deep learning algorithm is based on data processing, which contains a large number of computational operations. At the same time, in the application field of deep learning, there are many application scenarios that have certain requirements on performance, power consumption, etc., and usually require a high-performance or energy-efficient solution. Therefore, people usually use other hardware to accelerate deep learning algorithms when they are applied. Currently, there are three types of mainstream hardware accelerators, GPUs, ASICs, and FPGAs [25].

A. Acceleration Platforms

1. GPUs

Different from the traditional CPU structure, the internal structure of the GPU contains a large number of logical computing units, and the data transfer speed between the computing unit and the shared memory is much faster than the global memory. Besides, the GPU has relatively high-speed global memory with relatively large memory bandwidth, such as GDDR5. At present, some deep learning frameworks (such as Caffe, TensorFlow, etc.) can be better applied to GPUs, and companies like NVIDI also provide a better deep learning environment for GPUs, such as the deep learning acceleration library cuDNN.

2. ASICs

Unlike the way in which the deep learning algorithm is adapted to the CPU or GPU hardware structure to achieve acceleration, the main way to accelerate the deep learning algorithm using ASIC is to customize the dedicated hardware acceleration algorithm, such as the accelerated design for CNN algorithms [36]-[40]. Since the ASIC is specifically tailored to accelerate one or a certain type of algorithm, the acceleration effect is usually good and the power consumption is low, but at the same time, the re-configurability is poor and the development cost is high. Hardware design and development cycle are longer.

3. DSPs

DSP chips provide powerful DSP capabilities and are an effective platform for neural network acceleration [35]. The four DSP IP vendors have also released DSP IPs that support neural networks, including Synopsys' EV6x (embedded vision) processor, CEVA's CEVA-XM6, VeriSilicon's VIP8000, and Cadence's Vision C5 DSP.

4. FPGAs

In addition to CPU and GPU, FPGA is gradually becoming a candidate platform for energy-efficient neural network processing. FPGAs can achieve high parallelism and simplify logic according to the calculation process of a neural network with the hardware design for specific models. Therefore, FPGAs can achieve higher energy efficiency than CPUs and GPUs. Especially, various accelerators for deep CNN have been proposed based on FPGA platform because it has advantages of high performance, re-configurability, and fast development round, etc. Yangjing et al. [41] proposed a flexible and adaptable pulse neural network accelerator architecture based on FPGA, which can support the flexible configuration of neural network topology and connection weights. Wang Siyang [42] proposed a method of accelerating CNN based on the unified rotation strategy (URS) of the combination of lookup table (LUT) and greedy strategy, which accelerates the iterative convergence process of traditional CORDIC algorithm. They used this method to identify and verify the ETL9B handwritten Japanese database, which achieved 99.7% recognition accuracy and reduced the time consumption by about 90%.

B. Conclusion

Compared with the CPU neural network acceleration structure, FPGA is closer to the bottom layer, and usually has better energy efficiency, which is more suitable for the reasoning stage of the algorithm and the emerging deep neural network [43]. While ASIC acceleration outperforms FPGAs in performance, FPGA acceleration offers greater flexibility, lower development thresholds, and fewer development cycles.

In general, the use of FPGA as a deep learning algorithm accelerator is divided into two design methods; one is to design the RTL-level circuit structure using hardware circuit description languages such as VHDL, Verilog, etc. The other method is to use Advanced Synthesis (HLS). The tool

integrates high-level languages such as C into a hardware circuit bit stream file that the FPGA can recognize. Moreover, the current development environment based on FPGA is becoming more and more lightweight and efficient, and the high-level language C/C++ and Python language bring great convenience to the development process.

TABLE I
COMPARISON OF HARDWARE ACCELERATION SCHEMES

	CPU	GPU	ASIC	FPGA
Architectural difference	70% of the transistors are used to build the Cache, and there are some control units with few calculation units, which are suitable for handling complex logic and operations	Most of the transistors are built into the calculation unit, which has low computational complexity and is suitable for massively parallel computing	Transistors are customized according to the algorithm, no redundancy, low power consumption, high computational performance, and high computational efficiency	Programmable logic, high computational efficiency, closer to the underlying IO, logic programmable through redundant transistors and writing
Chip process	22nm	28nm	65nm	8nm
Highest performance device	E5-2699 V3	Tesla K80	DianNao	Virtex7-690T
Single precision floating point peak computing capability	1.33 TFLOPS	8.74 TFLOPS	452 TFLOPS	1.8 TFLOPS
Power consumption	145W	300W	485mW	30W
Energy consumption ratio	9 GFLOPS/W	29 GFLOPS/W	932 GFLOPS/W	60 GFLOPS/W

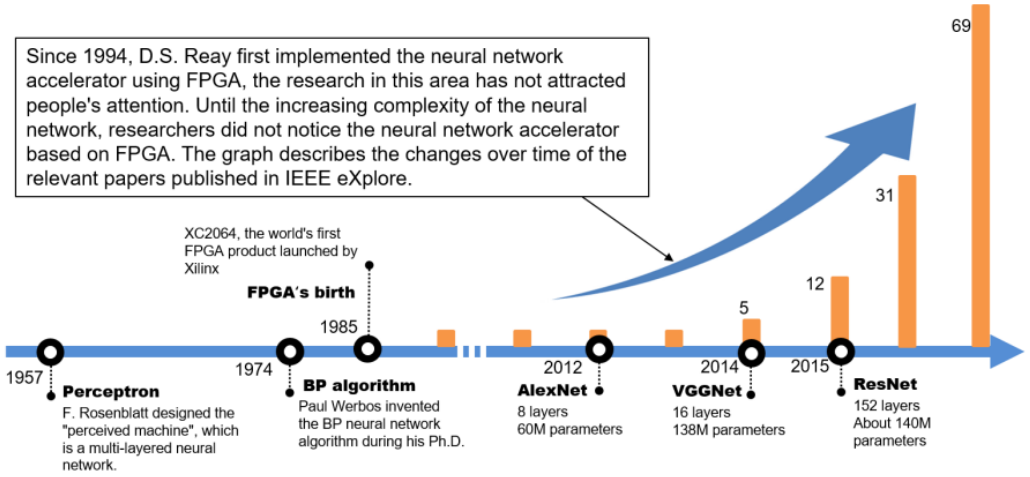


Fig. 4 Development history of the neural network accelerator based on FPGA [45]

In summary, due to the reconfigurable, customizable and energy-efficient features of FPGAs on acceleration of CNNs, it has become a research hot spot in research organizations at home and abroad.

III. The CNNs Hardware Accelerator Based on FPGA

A. Background

Back in the late 1990s when the FPGA were born, FPGAs were not originally developed for neural networks. Although in 1994 Reay [44] first used the FPGA to accelerate the neural network, due to the development of the neural network itself, it did not attract attention. Until the birth of AlexNet in ILSVRC 2012, the development direction of neural networks was clarified, and the research community is developing towards more in-depth and complex network research, such as late CNN, RNN, DNN and so on. Next is the generation of models such as VGGNet, GoogleNet, and ResNet, which fully marks the development trend of complex neural networks. Until December of 2020 the number of FPGA-based neural network accelerators published in the IEEE eXplore has reached 87 and is still on the rise. It is enough to illustrate the research trend in this direction.

The design of the CNN accelerator based on FPGA mainly follows some computational characteristics of the neural network training process, which should mainly consider the complex convolution calculation and the throughput of the whole system [46]. Farabet et al. [47] and Peemen et al. [48] mainly consider the bandwidth problem of on-chip storage and the correlation between different computing units. Sankaradas et al. [49] analyze how to maximize the use of on-chip stored data, including data storage addresses and multiple access issues. In general, CNNs require high memory bandwidth and a large amount of computing resources in hardware acceleration, so a good balance between the two is needed [50].

B. Research Status: Directions

Under the influence of Zhang et al. [51], according to the classification principle of accelerator architecture improvement and optimization, algorithm optimization and hardware performance improvement, the research status of an FPGA-based CNN acceleration problem at home and abroad

is reviewed. Among them, the hardware performance improvement includes the optimization of the development environment and the research of new materials and new processes.

1. Accelerator Architecture

Most research on FPGA-based CNN accelerators optimizes the architecture or optimizes the speed of one step or some steps in the architecture to increase the speed of the accelerator.

Since the traditional CNNs are computational-intensive and memory-intensive and unsuitable for the application in mobile edge computing scenarios, Ding et al. [52] presented a depth-wise separable CNNs and utilized a custom computing engine architecture based on Intel Arria 10 FPGA to process the dataflow between adjacent layers by using double-buffered memory channels. They had a performance of 98.9 GOP/s, which is 17.6 times faster and 29.4 times lower power consumption than CPU and GPU, respectively.

Zhang et al. [50] proposed an analytical design scheme using the roofline model to overcome the underutilization of either logic resource or memory bandwidth. They can identify the solution with best performance and lowest FPGA resource requirement through loop tiling and transformation, include memory access optimization, computation optimization, and design space (a block diagram of the proposed accelerator is shown in Fig. 5.)

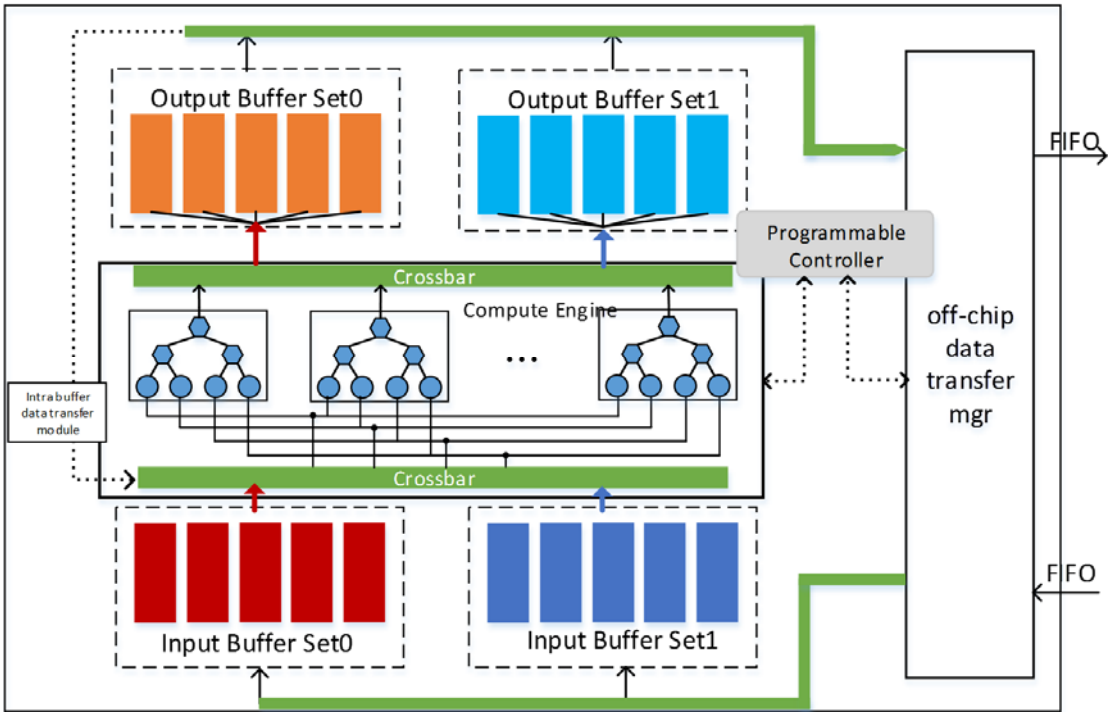


Fig. 5 Block diagram of proposed accelerator [50]

Liu and Liu [53] used the sparsity of CNN convolution calculation to convert CNN convolution calculation into matrix multiplication, and proposed an FPGA-based parallel matrix multiplication acceleration scheme. Simulation results on the Virtex-7 VC707 FPGA show that the design reduces computation time by 19% compared to traditional CNN accelerators.

2. Algorithm Optimization

Compared with the hardware acceleration method, the processing speed of the CNN is limited from the perspective of algorithm optimization. After all, the characteristics of CNNs have high requirements for computing resources and memory resources, and it is more difficult to optimize the network itself.

Suda et al. [54] proposed an extensible RTL compiler, named ALAMO, which analyzes algorithm structures and parameters, and automatically integrates a set of modular and scalable computational primitives to accelerate the operation of various deep learning algorithms on FPGAs.

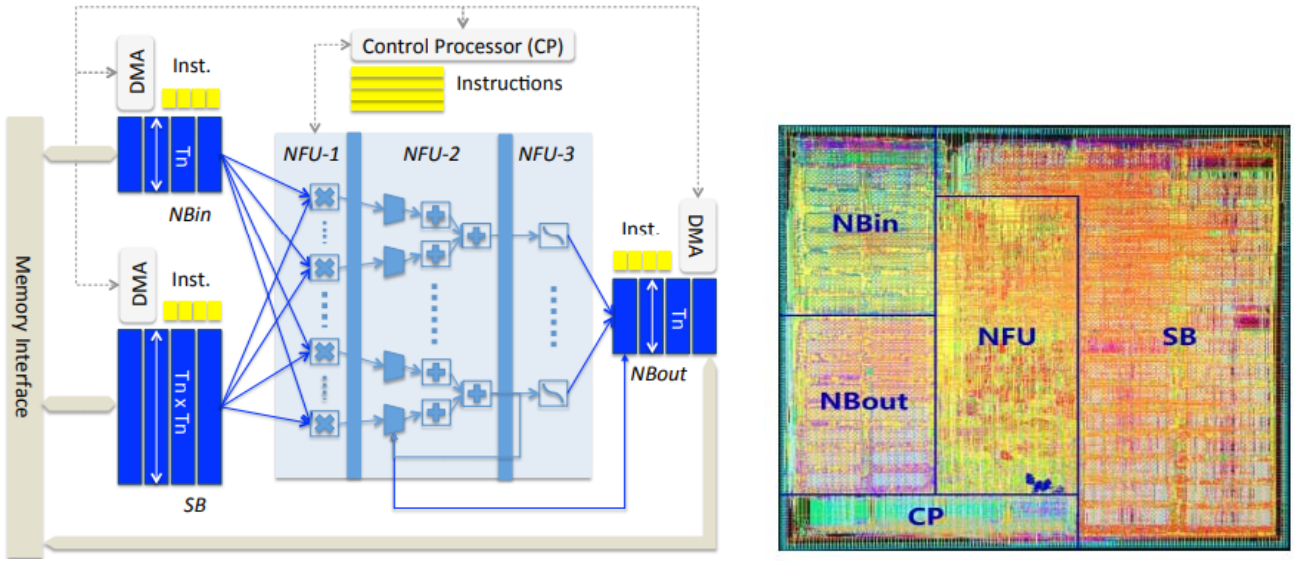
Yu et al. [55] studied the computational parallelism of CNN network structure, and designed an FPGA-based CNN accelerator to improve the data throughput rate through the pipeline architecture. And they perform parallel computation optimization on the convolution unit to improve the computational efficiency. Finally, the MINST handwritten numeric character library was used as the recognition object for experimental comparison. It was found that the accelerator can achieve the peak computing speed of the FPGA up to 0.676 GMAC/s under 75 MHz, which is four times faster than the general-purpose CPU platform and consumes only 2.68%.

3. Hardware Performance Improvement

The existing software implementation scheme is difficult to meet the requirements of the CNN for computing performance and power consumption. By improving the performance of

hardware accelerator and improving the parallelism of data processing, the accelerator is also greatly optimized.

Chen et al. [27] proposed a ubiquitous machine-learning hardware accelerator called DianNao, which launched the field of deep learning processors. It opens up new paradigms for machine learning hardware accelerators that focus on neural networks. But DianNao cannot adapt to different application demands because of its non-reconfiguration, which does not use FPGA or other reconfigurable hardware to achieve.



(a) (b)

Fig. 6 The block diagram (a) and the layout (65nm) (b) of DianNao accelerator [27], computational blocks: Neural Functional Unit, NFC, storage blocks: NBin, NBout, SB and control and code: CP

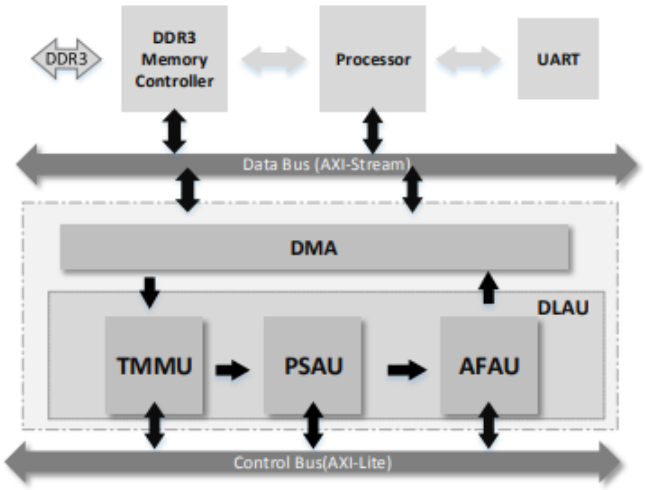


Fig. 7 DLAU Accelerator Architecture [60]

At present, many researchers have done a lot of work around FPGA acceleration researches. For the restricted Boltzmann machine (RBM), Ly and Chow [56] and Kim et al. [57] have proposed reliable and effective design solutions. The former has designed a dedicated hardware processing cores based on FPGA, which are optimized for the RBM algorithm. Similarly, the latter uses multiple RBM processing modules in parallel, with each module responsible for a relatively small number of nodes. Other similar works have also proposed FPGA-based neural network accelerators [58]. Yu et al. [59] proposed an FPGA-based accelerator, but it could not adapt to the ever-changing network scale and network topologies.

In order to improve the acceleration performance of FPGA-biased accelerator, Wang et al. [60] designed a scalable deep learning accelerator unit (DLAU) on FPGA, which uses three pipeline processing units (TMMU, PSAU and AFAU, as shown in Fig. 7) to increase throughput and explore the locality of deep learning applications using tile technology.

In conclusion, these studies focus on effectively implementing specific deep learning algorithms, but how to increase the size of neural networks and increase large networks through scalable and flexible hardware architectures (like FPGAs) has not been properly addressed [60].

C. Research Status: Challenges

Deploying FPGA system on accelerator for CNNs is still challenging. Some factors contribute to this situation, such as the large scale of dataflow, the computation-intensive requirement and frequent memory accesses. Although existing high-level synthesis tools for FPGAs (such as HLS and OpenCL) greatly reduce design and development time, the final implementation is still inefficient in achieving resource allocation to maximize parallelism and throughput. On the other hand, because the development speed of the current hardware platform does not match the development speed of the software development platform, there is no great FPGA hardware accelerated development environment, which brings great problems to the design of the neural network accelerator.

IV. Conclusion and Outlook

The birth of the world's first FPGA chip (1985) is more than 10 years after Gerald Estrin [62] proposed the concept of reconfigurable computing (the 1960s). At this time, although the FPGA platform has excellent parallelization and low power consumption characteristics, it has not attracted people's attention because of its high reconfiguration cost and high programming complexity. Software and hardware development conditions are still very poor. With the continuous development of deep learning, due to the high parallelism of its applications, more and more research institutions and companies are investing in FPGA-based deep learning accelerators research, which is also the trend of the times [46].

At present, the FPGA-based accelerator design has low reconfigurability and versatility, which will be the main direction of future FPGA-based accelerators design research. It is possible to improve versatility and reduce power consumption through the coupling of multiple acceleration platforms.

The future development of the accelerator has good opportunities, but at the same time it faces serious challenges. In this regard, we give a prospect for future development of FPGA-based accelerator.

1) The software development environment/EDA's property should be further optimized to improve communication issues between FPGA platforms.
2) Cloud services promote the acceleration performance of FPGAs. The rise and rapid development of cloud computing has brought new opportunities for the acceleration of neural networks. Virtualization of FPGA hardware resources, task migration and load balancing of virtualized FPGAs, and efficient parallel multi-machine FPGA heterogeneous acceleration architecture are worthy of further study.
3) The bottleneck of memory access should be solved. The access speed does not meet the requirements of increased computing speed, and it is still a difficult problem in future accelerator design [41].
4) After the algorithm model is compressed and then further accelerated by the FPGA, the cost of data transmission and data storage can be effectively reduced. In addition, the pipelined parallel acceleration method will also be a major direction of research [61].
5) Technological breakthroughs in multiple areas will also contribute to the improved performance of the accelerator. The technological revolution is often accompanied by a leap in different fields. New devices such as bio-inspired pulsed neural networks [42], quantum computers, and memristors are likely to provide a viable solution for future accelerator designs.
6) With the development of artificial intelligence chips, such as OPEN AI LAB's EAIDK-610 (RK3399), NVIDIA's Jetson Nano (CUDA-X), it is possible that FPGA-based neural network acceleration platforms will be more embedded, more lightweight and more portable in the future.
7) As AIoT's commercial applications continue to be landed, a large number of embedded hardware acceleration platforms are needed.

Acknowledgments

This work was supported in part by College Students' Innovative Entrepreneurial Training Plan Program of China x20180186 and 201911117138T. We thank Prof. Wang Zhongfeng of Nanjing University (NJU) for his directions in 2019 Winter Academic Program at NJU.

References

[1] Boukaye Boubacar Traore, Bernard Kamsu-Foguem, Fana Tangara, Deep convolution neural network for image recognition, in: Ecological Informatics, Volume 48,2018,Pages 257-268, ISSN 1574-9541,
[2] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[3] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556, 2014.
[4] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[5] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[6] Dawid Poap, Marcin Wozniak. Voice Recognition by Neuro-Heuristic Method (J). Tsinghua Science and Technology, 2019, 24(01):9-17.
[7] A. Ucar, Y. Demir, C. Guzelis, Object recognition and detection with deep learning for autonomous driving applications, (in English), Simul.-Trans. Soc. Model. Simul. Int. 93 (9) (Sep 2017) 759–769, doi:10.1177/0037549717709932.
[8] P. Pelliccione, E. Knauss, R. Heldal, et al., Automotive architecture framework: the experience of volvo cars, J. Syst. Archit. 77 (2017) 83–100. 06/01/ 2017 https://doi.org/10.1016/j.sysarc.2017.02.005.
[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521,no. 7553, pp. 436–444, 2015.
[10] D. Aysegul, J. Jonghoon, G. Vinayak, K. Bharadwaj,C. Alfredo, M. Berin, and C. Eugenio. Accelerating deep neural networks on mobile processor with embedded programmable logic. In NIPS 2013. IEEE, 2013.
[11] S. Cadambi, A. Majumdar, M. Becchi, S. Chakradhar, and H. P. Graf. A programmable parallel accelerator for learning and classification. In Proceedings of the 19th international conference on Parallel architectures and compilation techniques, pages 273{284. ACM, 2010.
[12] C. Farabet, C. Poulet, J. Y. Han, and Y. LeCun. Cnp: An fpga-based processor for convolutional networks. In Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on, pages 32{37. IEEE, 2009.
[13] M. Peemen, A. A. Setio, B. Mesman, and H. Corporaal. Memory-centric accelerator design for convolutional neural networks. In Computer Design (ICCD), 2013 IEEE 31st International Conference on, pages 13{19. IEEE, 2013.
[14] BY G. E. HINTON, R. R. SALAKHUTDINOV. Reducing the Dimensionality of Data with Neural Networks. SCIENCE28 JUL 2006 : 504-507. DOI: 10.1126/science.1127647
[15] Silver, D., Huang, A., Maddison, C. et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489 (2016). https://doi.org/10.1038/nature16961
[16] Hou Yuqingyang, Quan Jicheng, Wang Hongwei. Review of Deep Learning Development (J). Ship Electronic Engineering, 2017,37(04):5-9+111.
[17] Zhang Rong, Li Weiping, Mo Tong. Review of Deep Learning (J). Information and Control, 2018,47(04):385-397+410.
[18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, Tsuhan Chen, Recent advances in convolutional neural networks, Pattern Recognition, Volume 77, 2018, Pages 354-377, ISSN 0031-3203.

[19] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity (J). Bulletin of Mathematical Biophysics,1943,5(4): 115-133

[20] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain (J). Psychological Review, 1958,65(6):386-408

[21] Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex (J). Journal of Physiology, 1962,160(1):106-154

[22] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[23] Wu Yan-Xia, Liang Kai, Liu Ying, Cui Hui-Min. The Progress and Trends of FPGA-Based Accelerators in Deep Learning (J/OL). Chinese Journal of Computers, 2019:1-20 (2019-03-19). http://kns.cnki.net/kcms/detail/11.1826.TP.20190114.1037.002.html.

[24] Daofu Liu, Tianshi Chen, Shaoli Liu, Jinhong Zhou, et al. 2015. PuDianNao: A Polyvalent Machine Learning Accelerator. SIGARCH Comput. Archit. News 43, 1 (March 2015), 369–381. DOI:https://doi.org/10.1145/2786763.2694358

[25] Z. Du et al., "ShiDianNao: Shifting vision processing closer to the sensor," 2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA), Portland, OR, 2015, pp. 92-104, doi: 10.1145/2749469.2750389.

[26] Y. Chen et al., "DaDianNao: A Machine-Learning Supercomputer," 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture, Cambridge, 2014, pp. 609-622, doi: 10.1109/MICRO.2014.58.

[27] Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. 2014. DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning. SIGARCH Comput. Archit. News 42, 1 (March 2014), 269–284. DOI:https://doi.org/10.1145/2654822.2541967

[28] Liu Shaoli, Du Zidong, Tao Jinhua, et al. Cambricon: An instruction set architecture for neural networks (C). //Proc of the 43rd Int Symp on Computer Architecture. Piscataway, NJ:IEEE,2016:393-405

[29] Norman P. Jouppi, Cliff Young, Nishant Patil, et al. 2017. In-Datacenter Performance Analysis of a Tensor Processing Unit. SIGARCH Comput. Archit. News 45, 2 (May 2017), 1–12. DOI:https://doi.org/10.1145/3140659.3080246

[30] Google. Cloud TPUs: Google's second-generation tensor processing unit is coming to cloud (EB/OL). (2017-10-30). https://ai.Google/tools/cloud-tpus/

[31] Swagath Venkataramani, Ashish Ranjan, et al. 2017. ScaleDeep: A Scalable Compute Architecture for Learning and Evaluating Deep Networks. In Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA '17). Association for Computing Machinery, New York, NY, USA, 13–26. DOI:https://doi.org/10.1145/3079856.3080244

[32] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. 2016. Eyeriss: a spatial architecture for energy-efficient dataflow for convolutional neural networks. SIGARCH Comput. Archit. News 44, 3 (June 2016), 367–379. DOI:https://doi.org/10.1145/3007787.3001177

[33] Ali Shafiee, Anirban Nag, Naveen Muralimanohar, et al. 2016. ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars. SIGARCH Comput. Archit. News 44, 3 (June 2016), 14–26. DOI:https://doi.org/10.1145/3007787.3001139

[34] A. Parashar et al., "SCNN: An accelerator for compressed-sparse convolutional neural networks," 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), Toronto, ON, 2017, pp. 27-40, doi: 10.1145/3079856.3080254.

[35] Chen Guilin, Ma Sheng, Guo Yang. Survey on Accelerating Neural Network with Hardware (J/OL). Journal of Computer Research and Development, 2019(02) (2019-03-20). http://kns.cnki.net/kcms/detail/11.1777.TP.20190129.0940.004.html.

[36] Cavigelli L, Gschwend D, Mayer C, et al. Origami: A convolutional network accelerator // Proceedings of the Great Lakes Symposium on VLSI. Pittsburgh, USA, 2015: 199-204

[37] Chen Y-H, Krishna T, Emer J, et al. 14.5 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks // Proceedings of the 2016 IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, USA, 2016: 262-263

[38] Shafiee A, Nag A, Muralimanohar N, et al. ISAAC: A convolutional neural network accelerator with In-situ analog arithmetic in crossbars // Proceedings of the ISCA. Seoul, ROK, 2016: 14-26

[39] Andri R, Cavigelli L, Rossi D, et al. YodaNN: An ultra-low power convolutional neural network accelerator based on binary weights // Proceedings of the IEEE Computer Society Annual Symposium on VLSI. Pittsburgh, USA, 2016: 236-241

[40] Gokmen T, Vlasov Y. Acceleration of deep neural network training with resistive cross-point devices: design considerations. Front neurosci, 2016, 10(51): 333

[41] Shen Yangjing, Shen Juncheng, Ye Jun, Ma Qi. A FPGA Based Spiking Neuron Network Accelerator (J). Electronic Science and Technology, 2017, 30(10):89-92+96.

[42] Siyang Wang. FPGA based Convolutional Neural Network Accelerator Design and Realization (D). University of Electronic Science and Technology of China, 2017.

[43] Nurvitadhi E V G, Sim J, et al. Can FPGAs beat GPUs in accelerating next-generation deep neural networks? // Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. Monterey, USA, 2017: 5-14

[44] D. S. Reay, T. C. Green and B. W. Williams, "Field programmable gate array implementation of a neural network accelerator," IEE Colloquium on Hardware Implementation of Neural Networks and Fuzzy Logic, London, UK, 1994, pp. 2/1-2/3.

[45] Wang, T., Wang, C., Zhou, X., & Chen, H. (2018). A Survey of FPGA Based Deep Learning Accelerators: Challenges and Opportunities. arXiv preprint arXiv:1901.04988.

[46] C. Farabet, Y. LeCun, K. Kavukcuoglu, et al. Large-scale FPGA-based convolutional networks (J). In Scaling up Machine Learning: Parallel and Distributed Approaches eds Bekkerman, 2011, 399–419.

[47] C. Farabet, B. Martini, B. Corda, et al. Neuflow: A runtime reconfigurable dataflow processor for vision (C). In Computer Vision and Pattern Recognition Workshops, 2011, 109–116.

[48] M. Peemen, A. Setio, B. Mesman, et al. Memory-centric accelerator design for convolutional neural networks (C). IEEE International Conference on Computer Design, 2013, 13–19.

[49] M. Sankaradas, V. Jakkula, S. Cadambi, et al. A massively parallel coprocessor for convolutional neural networks (C). In Application Specific Systems, Architectures and Processors, 2009, 53–60.

[50] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, J. Cong, "Optimizing fpga-based accelerator design for deep convolutional neural networks", FPGA, 2015.

[51] Qianru Zhang, Meng Zhang, Tinghuan Chen, Zhifei Sun, Yuzhe Ma, Bei Yu, Recent advances in convolutional neural network acceleration, Neurocomputing, Volume 323, 2019, Pages 37-51,ISSN 0925-2312.

[52] Wei Ding, Zeyu Huang, Zunkai Huang, Li Tian, Hui Wang, Songlin Feng, Designing efficient accelerator of depthwise separable convolutional neural network on FPGA, Journal of Systems Architecture, 2018, ISSN 1383-7621.

[53] Liu Qinrang, Liu Chongyang. Calculation Optimization for Convolutional Neural Networks and FPGA-based Accelerator Design Using the Parameters Sparsity (J). Journal of Electronics & Information Technology, 2018,40(06):1368-1374.

[54] Yufei Ma, Naveen Suda et al., ALAMO: FPGA acceleration of deep learning algorithms with a modularized RTL compiler, Integration, the VLSI Journal (2017), https://doi.org/10.1016/j.vlsi.2017.12.009

[55] Yu Zijian, Ma De, Yan Xiaolang, Shen Juncheng. FPGA-based Accelerator for Convolutional Neural Network (J). Computer Engineering, 2017, 43(01):109-114+119.

[56] D. L. Ly and P. Chow, "A high-performance FPGA architecture for restricted Boltzmann machines," in Proc. FPGA, Monterey, CA, USA,2009, pp. 73–82.

[57] S. K. Kim, L. C. McAfee, P. L. McMahon, and K. Olukotun, "A highly scalable restricted Boltzmann machine FPGA implementation," in Proc. FPL, Prague, Czech Republic, 2009, pp. 367–372.

[58] J. Qiu et al., "Going deeper with embedded FPGA platform for convolutional neural network," in Proc. FPGA, Monterey, CA, USA, 2016,pp. 26–35.

[59] Q. Yu, C. Wang, X. Ma, X. Li, and X. Zhou, "A deep learning prediction process accelerator based FPGA," in Proc. CCGRID, Shenzhen, China, 2015, pp. 1159–1162.

[60] C. Wang, L. Gong, Q. Yu, X. Li, Y. Xie, X. Zhou, "DLAU: A scalable deep learning accelerator unit on FPGA", IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 36, no. 3, pp. 513-517, Mar. 2017.

[61] Chen Huang, Zhu Yong-xin, Tian Li, Wang Hui, Feng Song-lin. FPGA-based Design of Accelerator for Convolution Layer of Convolutional Neural Network (J). Microelectronics and Computer, 2018,35(10):85-88.

[62] G. Estrin, "The WEIZAC Years (1954-1963)," in Annals of the History of Computing, vol. 13, no. 4, pp. 317-339, Oct.-Dec. 1991, doi: 10.1109/MAHC.1991.10037.