EasyChair Preprint

№ 3633

A Survey of FPGA Based CNNs Accelerators

Wei Zhang

June 17, 2020

A Survey of FPGA Based CNNs Accelerators

Abstract: With the rapid development of deep learning, neural network and deep learning algorithms play a significant role in various practical applications. Especially, the high accuracy and good performance of CNNs has become a research hot spot in research organizations for 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 is aimed to survey the recent advance in 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 GPUs, ASICs and DSPs are compared, which is 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 and to further explore the opportunities and challenges for future research. More helpfully, we give a prospect for future development of FPGA-based accelerator.

Keywords: Deep learning, field programmable gate array (FPGA), hardware accelerator, convolutional neural networks (CNN)

1 Introduction

With the rapid development of deep learning, it speeded up the development of machine learning and artificial intelligence. Especially, convolutional neural networks (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]. The high accuracy and good performance of CNNs has become a research hot spot in research organizations[9].

However, with the increasing accuracy requirements and complexity for the practical applications, the size of the neural networks becomes explosively large scale. 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's more, it poses significant challenges to implement high performance deep learning networks with low power cost, especially for large-scale deep learning neural network models. Therefore, it is particularly important to choose a moderate computing platform for neural network applications.

So far, the state-of-the-art means for accelerating deep learning algorithms are field programmable gate array (FPGA), application specific integrated circuit (ASIC), graphic processing unit (GPU), and digital signal processors (DSP). 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][11][12][13]. So the primary survey here is FPGA-based acceleration of CNNs in

this article about 4 past years.

1.1 Overview of Deep Learning

In 2006, deep learning leader Hinton 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 defeated the master of human Go. The history of deep learning development is shown in Fig. 1.1. The open circle in Figure 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。

Houyu Chunyang[14] 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.

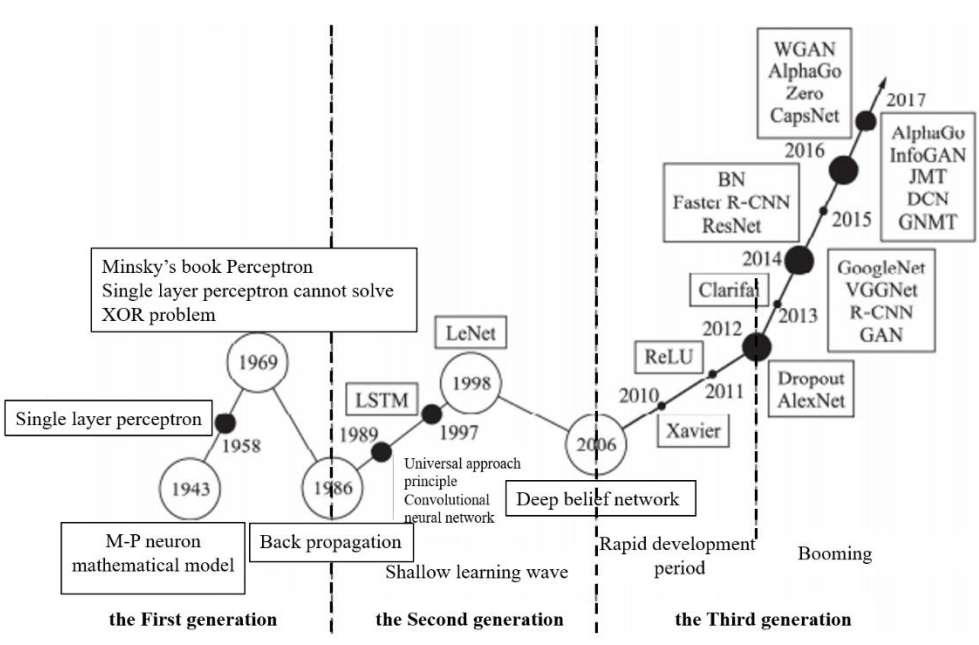

Fig. 1.1: the history of deep learning[15]

1.2 Overview of CNNs[16]

Artificial neural networks are a typical machine learning method and an important form of deep learning. From 1943 McCulloch and Pitts[17] first proposed artificial neuron models (M-P neurons, as shown in Fig. 1.2 (a)) to 1958 Rosenblatt[18] designed the perceptron (Fig. 1.2 (b)). As

the number of layers of the perceptron increases, there are deep neural networks as multiple hidden layers increase (Fig. 1.2 (c, d)). In order to solve the problem of long calculation time and high power consumption, Hubel and Wiesel[19] proposed a convolutional neural network in the 1960s (as shown in Fig. 1.3: A representative CNN architecture). The characteristics of local perception and parameter sharing of convolutional neural networks enable it to effectively reduce the number of parameters and reduce the complexity of deep neural networks.

Convolutional neural network (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][20][21]

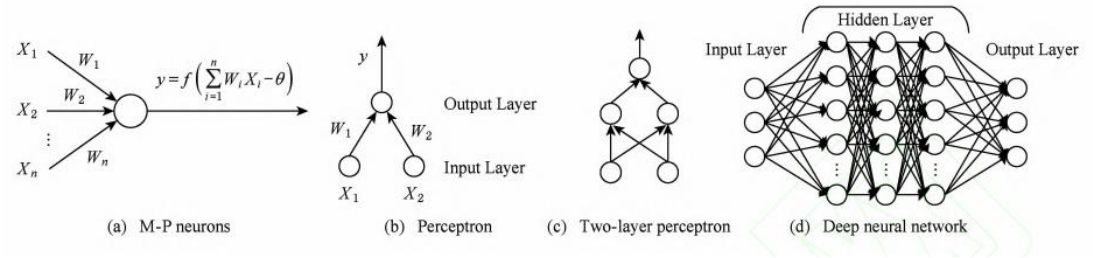

Fig. 1.2: the evolution of neural network[30]

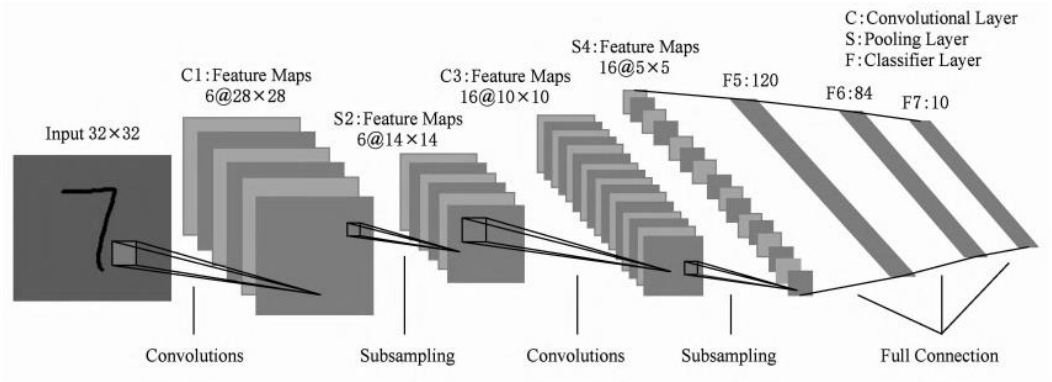

Fig. 1.3: A representative CNN architecture-LeNet5[30]

1.3 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 develop 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 Chen Tianshi team[22] of the Chinese Academy of Sciences (Based on RISC, a new dedicated instruction set architecture called Cambricon is designed.), the TPU (tensor process unit) and the cloud TPU of Google, the Scaledeep launched by Purdue University, the Eyeriss proposed by MIT, and the HP Lab. The memristor-based ISAAC jointly proposed with the University of Utah and the compressed sparse convolutional neural network accelerator SCNN proposed by Parashar et al. The research of existing neural network acceleration chips mainly focuses on 4 aspects[30]:

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.

2 Hardware Acceleration of CNNs[23]

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(Graphics Processing Units)， ASICs(Application-Specific Integrated Circuits)， FPGAs(Field-Programmable Gate Array).[24]

2.1 Acceleration platforms

2.1.1 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.

2.1.2 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 convolutional neural network algorithms[25][26][27][28][29]. 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.

2.1.3 DSPs

Digital signal processing (DSP) chips provide powerful digital signal processing capabilities and are an effective platform for neural network acceleration[30]. 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.

2.1.4 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. Shenyang Jing et al.[31] 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 et al.[32] proposed a method of accelerating convolutional neural network 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%.

2.2 Comparative analysis of hardware acceleration schemes

We compared the above architectural difference, chip process, highest performance device, single precision floating point peak computing capability, power consumption, energy consumption ratio etc. as shown in Table 2.1. Obviously, FPGA has its own advantages in terms of accelerated deep learning compared with GPU and ASIC. The unique advantage of using FPGA to accelerate deep learning algorithms has attracted the attention of scholars.

Table 2.1 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 wiring.
Chip process	22nm	28nm	65nm	8nm
Highest performance device	E5-2699 V3	Tesla K80	DianNao	Virtex7-690T
Single precision floating point peak computing capability	1.33TFLOPS	8.74 TFLOPS	452 GOPS	1.8 TFLOPS
Power consumption	145W	300W	485mW	30W
Energy consumption ratio	9 GFLOPS/W	29 GFLOPS/W	932 GFLOPS/W	60 GFLOPS/W

2.3 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[33]. 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.

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.

3 The convolutional neural networks hardware accelerator based on FPGA

3.1 Background

Back to 1990s when FPGAs were born, the FPGA was not developed for the neural network at first, but for the rapid development of electronic hardware prototype. The neural network has been explored for new improvements and applications since its inception, but it has not formed a unified research direction. Although in 1994 D.S. Reay 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 March of this year, 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.

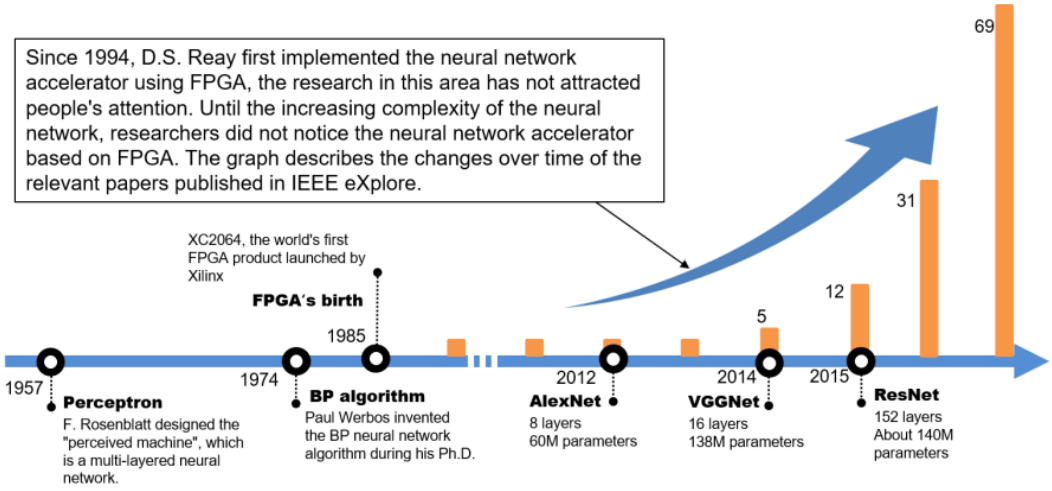

Fig. 3.1: Development history of the neural network accelerator based on FPGA.[34]

The design of the convolutional neural network 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[35]. C. Farabet and M. Peemen et al.[36][37] mainly consider the bandwidth problem of on-chip storage and the correlation between different computing units. M. Sankaradas et al.[38] analyzes how to

maximize the use of on-chip stored data, including data storage addresses and multiple access issues. In general, convolutional neural networks require high memory bandwidth and a large amount of computing resources in hardware acceleration, so a good balance between the two is needed[21].

3.2 Research Status: Directions

Under the influence of Qianru Zhang[23], according to the classification principle of accelerator architecture improvement and optimization, algorithm optimization and hardware performance improvement, the research status of FPGA-based convolutional neural network 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.

3.2.1 Accelerator architecture

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

Due to the traditional CNNs are computational-intensive and memory-intensive and unsuitable for the application in mobile edge computing scenarios, Wei Ding et al.[39] present a depth-wise separable CNNs and utilize a custom computing engine architecture on Arria 10 FPGA to handle the dataflow between adjacent layers by using double-buffering-based memory channels. They has a performance of 98.9 GOP/s and achieve up to 17.6× speed up and 29.4× low power than CPU and GPU implementations respectively.

Chen Zhang and Yijin Guan et al.[21] 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. (The Block diagram of proposed accelerator as shown Fig.3.2)

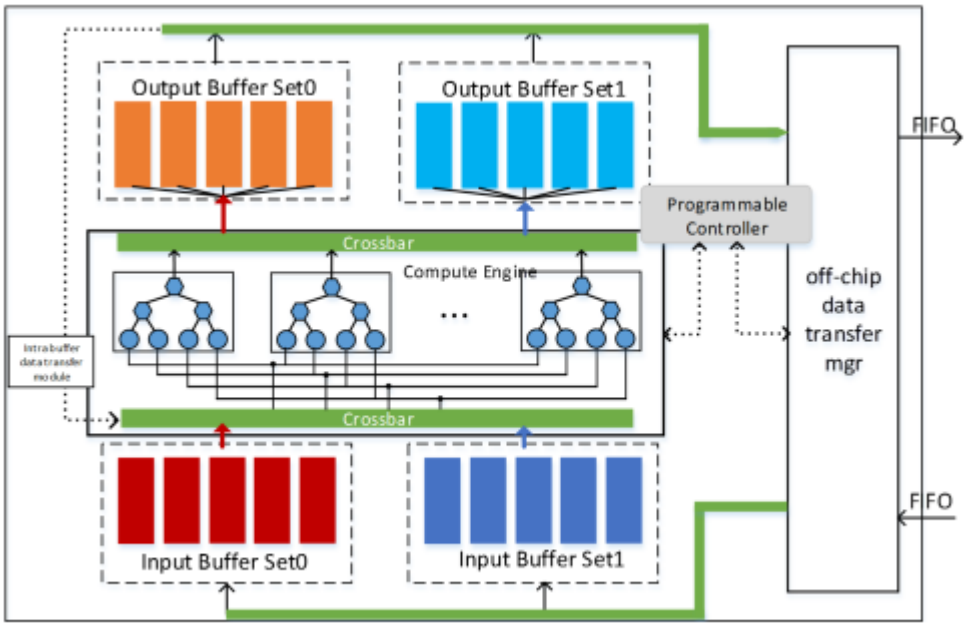

Fig.3.2: Block diagram of proposed accelerator[21]

Liu Qinrang and Liu Chongyang[40] 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.

3.2.2 Algorithm optimization

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

Yufei Ma, Naveen Suda et al.[41] 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. Integrating these modules together for end-to-end CNN implementations, this work quantitatively analyzes the complier's design strategy to optimize the throughput of a given CNN model under the FPGA Resource proposed.

Zijian Yu et al.[42] studied the computational parallelism of CNN network structure, and designed an FPGA-based convolutional neural network 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 4 times faster than the general-purpose CPU platform and consumes only 2.68%.

3.2.3 Hardware performance improvement

On the other hand, the existing software implementation scheme is difficult to meet the requirements of the convolutional neural network for computing performance and power consumption. By improving the hardware performance of the accelerator and improving the parallelism of data processing, the accelerator is also greatly optimized.

Chen et al.[43] presented a ubiquitous machine-learning hardware accelerator called DianNao, which initiated the field of deep learning processor. It opens a new paradigm to machine learning hardware accelerators focusing on neural networks. But DianNao is not implemented using reconfigurable hardware like FPGA, therefore it cannot adapt to different application demands.

Currently, around FPGA acceleration researches, Ly and Chow[44] designed FPGA-based solutions to accelerate the restricted Boltzmann machine (RBM). They created dedicated hardware processing cores which are optimized for the RBM algorithm. Similarly, Kim et al.[45] also developed an FPGA-based accelerator for the RBM. They use multiple RBM processing modules in parallel, with each module responsible for a relatively small number of nodes. Other similar works also present FPGA-based neural network accelerators[46]. Yu et al.[47] presented an FPGA-based accelerator, but it cannot accommodate changing network size and network topologies.

In order to improve the acceleration performance of FPGA-biased accelerator, Chao Wang et al.[48] design deep learning accelerator unit (DLAU), which is a scalable accelerator architecture

for large-scale deep learning networks using field-programmable gate array (FPGA) as the hardware prototype. The DLAU Accelerator uses three pipeline processing units to increase throughput and explore the locality of deep learning applications using tile technology.

To sum up, these studies focus on implementing a particular deep learning algorithm efficiently, but how to increase the size of the neural networks with scalable and flexible hardware architecture has not been properly solved[48].

3.3 Research Status: Challenges

Deploying FPGA system on accelerator for Convolutional Neural Networks (CNNs) is still challenging due to the large volume of data, the extensive amount of computation and frequent memory accesses. Although existing high-level synthesis tools (e.g. HLS, OpenCL) for FPGAs dramatically reduce the design time, the resulting implementations are still inefficient with respect to resource allocation for maximizing 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.

4 Conclusion and Outlook

As early as the 1960s, Gerald Estrin proposed the concept of reconfigurable computing. It was not until 1985 that the first FPGA chip was introduced by Xilinx. Although the parallelization and power consumption of the FPGA platform is excellent, it has not been paid attention to because of its cost of reconfiguration and high programming complexity. With the continuous development of deep learning, due to the high parallelism of its applications, more and more researchers are investing in the research of FPGA-based deep learning accelerators, which is also the trend of the times。[34]

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) Optimize the software development environment and improve communication problems 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) Solve the bottleneck of memory access. The access speed does not meet the requirements of increased computing speed, and it is still a difficult problem in future accelerator

design.[30]

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[49].
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[31], 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’s possible that embedded, lightweight and portable will be a research hot spot in the development of FPGA-based neural network acceleration platform in the future.

Refences

[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 Po?ap,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] 侯宇青阳,全吉成,王宏伟.深度学习发展综述[J].舰船电子工程,2017,37(04):5-9+111.

[15] 张荣,李伟平,莫同.深度学习研究综述[J].信息与控制,2018,47(04):385-397+410.

[16] 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,

[17] 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

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

[19] 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

[20] M. Sankaradas, V. Jakkula, S. Cadambi, S. Chakradhar,I. Durdanovic, E. Cosatto, and H. P. Graf. *A massively parallel coprocessor for convolutional neural networks.* In *Application-specific Systems, Architectures and Processors, 2009. ASAP 2009. 20th IEEE International Conference on*, pages 53{60. IEEE, 2009.

[21] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, J. Cong, "*Optimizing fpga-based accelerator*

design for deep convolutional neural networks", FPGA, 2015.
[22] 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
[23] 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,
[24] 吴艳霞,梁楷,刘颖,崔慧敏.深度学习 FPGA 加速器的进展与趋势[J/OL].计算机学报,2019:1-20[2019-03-19].http://kns.cnki.net/kcms/detail/11.1826.TP.20190114.1037.002.html.
[25] 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
[26] 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
[27] 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
[28] 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
[29] Gokmen T, Vlasov Y. *Acceleration of deep neural network training with resistive cross-point devices: design considerations*. Front neurosci, 2016, 10(51): 333
[30] 陈桂林,马胜,郭阳.硬件加速神经网络综述[J/OL].计算机研究与发展,2019(02)[2019-03-20].http://kns.cnki.net/kcms/detail/11.1777.TP.20190129.0940.004.html.
[31] 沈阳靖,沈君成,叶俊,马琪.基于 FPGA 的脉冲神经网络加速器设计[J].电子科技,2017,30(10):89-92+96.
[32] 王思阳. 基于 FPGA 的卷积神经网络加速器设计[D].电子科技大学,2017.
[33] 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
[34] 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.
[35] 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.
[36] C. Farabet, B. Martini, B. Corda, et al. *Neuflow: A runtime reconfigurable dataflowprocessor for vision*[C]. In Computer Vision and Pattern Recognition Workshops, 2011,109–116.
[37] M. Peemen, A. Setio, B. Mesman, et al. *Memory-centric accelerator design forconvolutional neural networks*[C]. IEEE International Conference on Computer Design,2013, 13–19.
[38] M. Sankaradas, V. Jakkula, S. Cadambi, et al. *A massively parallel coprocessor forconvolutional neural networks*[C]. In Application Specific Systems, Architectures andProcessors, 2009, 53–60.
[39] 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.
[40] 刘勤让,刘崇阳.利用参数稀疏性的卷积神经网络计算优化及其 FPGA 加速器设计[J].电子与信息学报,2018,40(06):1368-1374.
[41] Yufei Ma, Naveen Suda, Yu Cao, Sarma Vrudhula, and Jae Sun Seo. 2018. *ALAMO: FPGA acceleration of deep learning algorithms with a modularized RTL compiler*. Integration, the VLSI Journal. [doi>10.1016/j.vlsi.2017.12.009]
[42] 余子健,马德,严晓浪,沈君成.基于 FPGA 的卷积神经网络加速器[J].计算机工程,2017,43(01):109-114+119.
[43] T. Chen et al., "*DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning*," in Proc. ASPLOS, Salt Lake City, UT, USA, 2014, pp. 269–284.

[44] 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.
[45] 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.
[46] J. Qiu *et al.*, "*Going deeper with embedded FPGA platform for convolutional neural network*," in *Proc. FPGA*, Monterey, CA, USA, 2016,pp. 26–35.
[47] 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.
[48] 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.
[49] 陈煌,祝永新,田犁,汪辉,封松林.基于 FPGA 的卷积神经网络卷积层并行加速结构设计[J].微电子学与计算机,2018,35(10):85-88.