Review on FPGA-based accelerators in convolutional neural network

Yiming Li

School of Microelectronics, Tianjin University, Tianjin, 300110, China

lym_cdbd@163.com

Abstract. In current research, an important means to realize artificial intelligence is artificial neural network. Many tedious problemscan be solved by artificial neural network, such as image classification, speech recognition, natural language processing. Among neural networks, although convolutional neural network has significantly better performance in the field of image recognition, it requires many parameters and a large amount of computation, and the number of network layers is gradually increasing with the progress of the algorithm, which leads to model sizes are getting larger and more difficult to handle, which has more stringent requirements on various aspects of hardware capabilities, such as computing power, data storage and memory bandwidth. Therefore, the research of acceleration is particularly important, especially in the field of hardware acceleration. And as an emerging hardware platform, field programmable gate array has the characteristics of rapid customization and programmability, which can greatly improve the efficiency of accelerators' design, implementation, and verification. As a result, it can be widely used in the hardware-accelerated design of neural networks. Although field programmable gate array itself has some defects, many research results have proved its huge advantages and development potential. This paper mainly summarizes the acceleration of convolutional neural network based on field programmable gate array platform, and discusses its characteristics and application space by comparing with other platforms, showing its high applicability to convolutional neural network, and finally looking forward to the progress of related research work.

Keyword: Convolutional Neural Network, FPGA, Hardware Acceleration, Deep Learning.

1. Introduction

The research of artificial neural network began with the concept proposed by Warren McCulloch and Walter Pitts in 1943. With the improvement of hardware conditions in the past 80 years, data and computing power have continued to increase. Neural networks have been developed step by step from perceptron and multi-layer perceptron to deep neural networks. Among the many classifications of neural networks, convolutional neural networks(CNNs) play an important role in the field of computer image processing and recognition [1]. It imitates the structure of biological visual system, and the common structure of its hidden layer includes convolution layer, pooling layer, and fully connected layer [2], as shown in Figure 1.

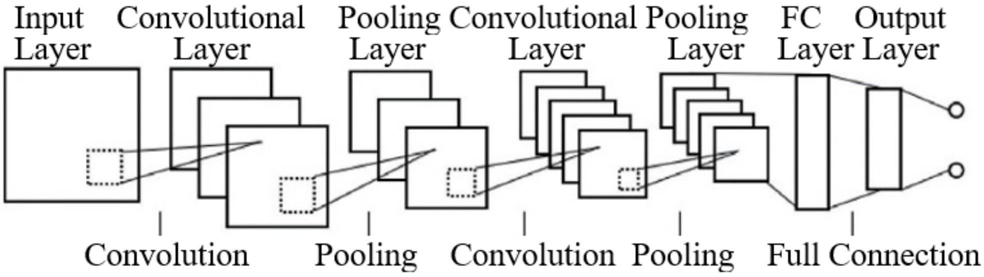

Figure 1. Illustration of convolutional neural network[2].

To improve the accuracy of CNN, the number of its layers is an important factor. Many studies have shown that the fitting ability of the CNN model will be significantly improved with the increase of the number of network layers. Table 1 shows the relevant information about the neural network models that won the ImageNet Large Scale Visual Recognition Challenge(ILSVRC) from 2012 to 2015. As the neural network increases from 8 layers to 152 layers, the error rate of Top-5 decreases from 15.30% to 3.57%. However, in the process, it can be found that while the number of layers increases, the model size and calculation amount of VGG are several times or even dozens of times larger than those of AlexNet, and ResNet compared to GoogLeNet, which all means that the requirements for hardware computing resources are also significantly increased. This phenomenon indicates that it is necessary to conduct research on the acceleration of CNNs, which is beneficial to its practical application in industrial fields and mobile devices.

Table.1 Information about models in ILSVRC

Network Model	Top-5 Error/%	Number of Layers	Model Size/MB	Times of Multiplication＆Addition Calculations(Operations)/Billion
AlexNet(ILSVRC'12)	15.30	8	240	0.72
VGG(ILSVRC'14)	7.30	19	500	19.6
GoogLeNet(ILSVRC'14)	6.70	22	24	1.55
ResNet(ILSVRC'15)	3.57	152	240	11.3

Field Programmable Gate Array (FPGA), is originally a semi-custom chip in the Application Specific Integrated Circuit (ASIC) world. Its key feature is reconfigurability, which gives FPGAs the advantage of being rapidly modifiable compared to custom circuits, and at the same time more reliable than previous programmable device gates. The principle of its reconfiguration is to use the compiled hardware program to control the interconnection form between various parts, thereby forming different logical structures. Researchers can recompile and simulate through the FPGA platform to quickly design and verify accelerators, making FPGA-based neural network acceleration one of the research hotspots.

In this paper, the basic structure of FPGA's common CNN hardware accelerator is summarized and studied, based on which FPGA is compared with other platforms. And its characteristics, advantages and disadvantages are discussed, to propose directions for possible future research.

2. FPGA Accelerator for CNNs

Computation and data transfer are two perspectives for acceleration of CNN based on FPGA [3]. High parallel computing design and optimized hardware structure are usually adopted to improve computing efficiency; meanwhile, data volume and memory access times are reduced to achieve fast data transmission.

This subsection will discuss the hardware structure of two common convolutional neural network FPGA accelerators.

The hardware structure shown in Figure 2 makes the computing throughput and memory bandwidth more matched, increasing the computing power of the CNN [4].

Its calculation is done in parallel by multiple computing engines, and different engines calculate the convolution of different convolution kernels, which is equivalent to multiple convolution kernels in parallel, performing convolution operations on the same input data. In the computing engine, the bottom layer is a multiplier, above which are multi-layer adders, forming a "tree" structure. The multiplication and addition operations between matrices are the essence of convolution operations, and the multiplication operations of the data at the corresponding positions of the matrix are independent of each other. So, the multiplication operations are performed in parallel by the bottom multipliers, and then the upper layer adders do addition in parallel. This hardware structure fully parallelizes the computations in the convolutional layers, increasing the computation speed.

In the data transmission part, since the convolution kernels are independent of each other, multiple parallel convolution operations can be performed only by reading the data once. This improves the low usage of local data and reduces the number of accesses to off-chip storage. At the same time, two buffers are added in the input and output parts, so that the data transmission time is covered under the calculation time, and the program execution time is reduced. This hardware structure can achieve the highest performance of 61.62GFLOPS at 100MHz frequency [4], which has significant advantages over other designs.

However, there are still some problems with this hardware structure(Figure 2). This structure is only for convolutional layers and does not consider fully connected layers that contain frequent data interactions. And the "tree" structure in its computing engine is only suitable for regular convolution calculations, and is not suitable for pruned sparse CNNs [5].

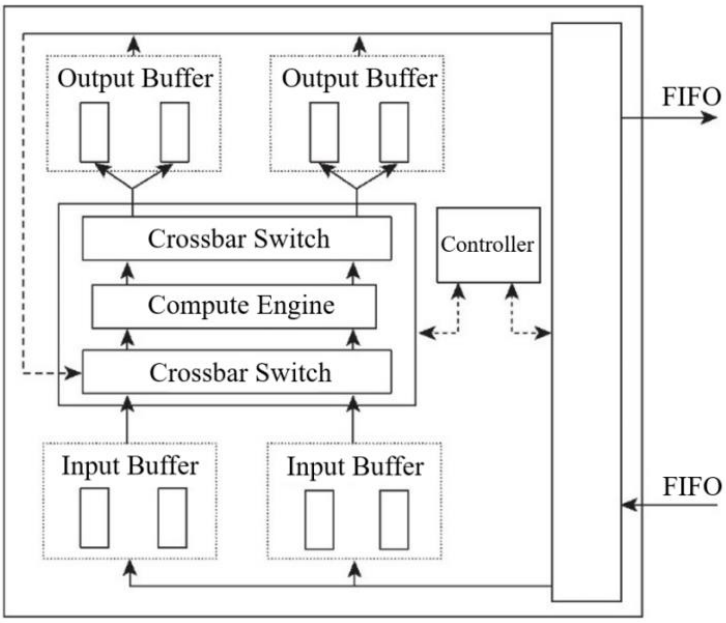

Figure 2. Parallel hardware structure[4].

Another FPGA accelerator shown in Figure 3 uses a binary neural network to reduce data accuracy and increase data transmission speed [6]. Binarized Neural Network (BNN) refers to a neural network that uses only two values of +1 and -1 to represent weights and activations. The number of data bits in the FPGA hardware design can be set by the developer, so the FPGA can support a variety of data types with different precisions. And it can be applied to low-precision data without modifying the overall structure. In the case of extremely low precision, such as when the number of data bits is 1, the multiplication and addition of data can be converted into simple bit operations, which can greatly reduce the computational difficulty. Therefore, the multipliers in this structure are replaced by the XNOR gates, which reduces the use of resources and improves computational efficiency. The only disadvantage is that this structure only optimizes the calculation process and ignores the data interaction part. If the input and output buffers mentioned above can be combined, the acceleration performance will go even further.

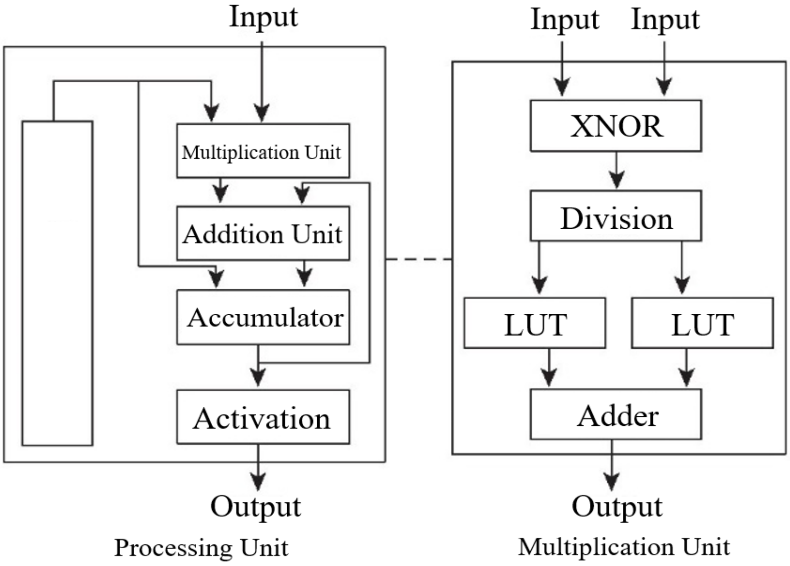

Figure 3. Hardware structure with binary neural network[6].

3. Comparison of FPGA Acceleration with GPU and ASIC Acceleration

3.1 FPGA acceleration VS GPU acceleration

3.1.1. Hardware structure. The hardware structure of GPU is fixed. Therefore, in order to obtain better acceleration performance, it is usually necessary to adjust the algorithm of the CNN to adapt to the hardware structure and model when designing the accelerator. In contrast, the reconfigurability and programmability of the FPGA make its hardware structure capable of adapting the algorithm, so the hardware structure can be designed according to the algorithm. This makes FPGA-based accelerators better suited to the rapid development of algorithms.

*3.1.2. Energy efficiency ratios.*FPGAs usually have better energy efficiency ratios. The computing power of the basic unit based on the lookup table in the FPGA is not as good as that of the arithmetic logic unit in the CPU and GPU, and the DSP resources used for floating-point calculation on the FPGA are also less than that of the GPU. Therefore, the floating-point computing power of the FPGA is weaker than that of the GPU. But FPGA technology is in a stage of rapid development, and its computing power is rapidly improving. For example, the 32-bit floating-point computing throughput of Intel Stratix 10 is expected to reach 9.2TFLOP/s, which is close to the peak 32-bit floating-point computing throughput of 11TFLOP/s provided by the latest Titan X Pascal GPU [7]. At the same time, FPGAs typically consume less power than GPUs. Combining the above two aspects, in normal circumstances, FPGA can achieve higher performance under unit power consumption [8]. For the binary neural network accelerator mentioned above, the energy efficiency ratio obtained in the VGG8 model at a frequency of 150 MHz is 10 times that of the GPU (GTX Titan X). For the inference stage of the CNN, the Microsoft team used an FPGA (Arria10GX1150) to achieve a performance of 233 photos per second with a power consumption of about 25 watts; and for the high-performance GPU implementation (Caffe+cuDNN), its acceleration performance is 500~824 photos per second, and the power consumption is 235 watts [9]. After comparison, it can be seen that the energy efficiency ratio of FPGA is 2~3 times that of GPU. This feature is very important for the application of CNNs with limited resources.

*3.1.3. Emerging convolutional neural networks.*FPGAs are more suitable for emerging CNNs. Since GPUs are dominated by arithmetic logic units, performance will drop significantly when accelerating sparse CNNs. And the GPU can only support 32-bit floating-point and 8-bit integer data. Its performance on data types with custom number of digits is far less than FPGA. It can be seen from the hardware structure analysis above that FPGA has good support capability for low-precision data. The experimental results show that the performance (TOP/sec) of FPGA (Stratix 10) in matrix multiplication operations of sparse deep neural network, deep neural network with 6-bit integer data and binary deep neural network is better than that of GPU (Titan X Pascal) increased by 10%, 50% and 5.4 times [7].

3.2. FPGA acceleration VS ASIC acceleration

ASIC acceleration generally outperforms FPGAs in performance. The TrueNorth chip designed by IBM Corporation consumes only 659mW when accelerating a typical complex neural network, and the corresponding computing power consumption is 46 billion synaptic operations per watt [10]. But ASIC is dedicated hardware designed for a certain type of application and the hardware structure cannot be changed after generation. However, FPGA is designed with universality and reconfigurability, which makes its development cycle shorter and less difficult to develop. FPGA acceleration can flexibly adapt to some widely used but immature algorithms.

4. Discussions

The most notable feature of an FPGA, and its most notable advantage, is its flexibility. Due to the reconfigurability of FPGA, it can realize rapid customization for different fields, different algorithms, and different requirements. It is also possible to perform fast software and hardware iterative optimization on the FPGA, so that the FPGA-based CNN hardware accelerator can be continuously upgraded. But the time consumed in the refactoring process cannot be ignored.

The reconfiguration of FPGA is divided into static and dynamic. Static reconfiguration is to change and fix the hardware logic form before the algorithm runs. While dynamic reconfiguration is to reconfigure the hardware structure and logic as needed during program operation. Although only a part of the hardware structure needs to be dynamically reconfigured, the time required for it far exceeds the computation time. For example, for the convolutional layer of the CNN, the reconfiguration time is 155ms, and the calculation time of this layer is only 2.7ms [11]. The relationship between calculation and reconfiguration needs to be reasonably arranged to cover the time required for reconfiguration and make the entire algorithm run more efficiently.

In addition, the FPGA customizable characteristics may also affect the final calculation accuracy. As mentioned above, one of the acceleration methods of FPGA for CNNs is to use lower bit-width data units, that is, to reduce data precision. This will increase the computing performance several times, but will also increase the computing error a lot. It is necessary to design the corresponding data bit width for different CNN algorithms, and find a balance between calculation speed and accuracy. Or adjust the computing core according to the bit width of the input data like BitFusion [12], so that the parallelism of the calculation is increased, which is also a good solution.

5. Conclusion

The high parallelization of FPGA is its main feature for accelerating CNNs. The use of convolution kernel parallelism, multiply-add parallelism and other strategies can effectively improve the efficiency of the algorithm [13]. With the continuous development of FPGA platform hardware, its reconfigurable characteristics make FPGA-based convolutional neural network accelerators have great development prospects compared to that based on ASIC, and improve the efficiency of researchers in designing, testing, and manufacturing accelerators. This means that the accelerator is highly adaptable to rapidly-changing CNNs and can be applied in many emerging fields. At the same time, FPGAs have higher energy efficiency than GPUs, laying the foundation for hardware acceleration in scenarios that require low power consumption, such as in the terminal or edge computing fields. However, the design needs to make more trade-offs in the consumption of its reconfiguration and the accuracy of the calculation.

From the current development trend, future research mainly in three aspects.

The first aspect is combining FPGA with other hardware platforms. It should be noted that the bandwidth problem of the FPGA is solved to increase the communication efficiency with other hardware and reduce the communication delay. Moreover, the dynamic data accuracy can be researched as it allows different parts of the CNN to use data with different numbers of bits, considering both computational efficiency and accuracy. Finally, FPGA cluster, achieve high performance in test due to the integrates multiple FPGA chips [14]. But there are few studies in this area. It is necessary to adjust the weight distribution between each chip to improve the storage and communication efficiency.

It is foreseeable that the FPGA-based convolutional neural network acceleration technology will continue to improve, and ultimately promote the transformation and development of the entire artificial intelligence field.

Reference

[1] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77, 354-377.

[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[3] Kala, S., & Nalesh, S. (2020). Efficient cnn accelerator on fpga. IETE Journal of Research, 66(6), 733-740.

[4] Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015, February). Optimizing FPGA-based accelerator design for deep convolutional neural networks. In Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays (pp. 161-170).

[5] Zhao, X., Zhang, X., Yang, F., Xu, P., Li, W., & Chen, F. (2021, August). Research on Machine Learning Optimization Algorithm of CNN for FPGA Architecture. In Journal of Physics: Conference Series (Vol. 2006, No. 1, p. 012012). IOP Publishing.

[6] Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., & Marr, D. (2016, December). Accelerating binarized neural networks: Comparison of FPGA, CPU, GPU, and ASIC. In 2016 International Conference on Field-Programmable Technology (FPT) (pp. 77-84). IEEE.

[7] Nurvitadhi, E., Venkatesh, G., Sim, J., Marr, D., Huang, R., Ong Gee Hock, J., ... & Boudoukh, G. (2017, February). Can FPGAs beat GPUs in accelerating next-generation deep neural networks?. In Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays (pp. 5-14).

[8] Qasaimeh, M., Denolf, K., Lo, J., Vissers, K., Zambreno, J., & Jones, P. H. (2019, June). Comparing energy efficiency of CPU, GPU and FPGA implementations for vision kernels. In 2019 IEEE international conference on embedded software and systems (ICESS) (pp. 1-8). IEEE.

[9] Lacey, G., Taylor, G. W., & Areibi, S. (2016). Deep learning on fpgas: Past, present, and future. arXiv preprint arXiv:1602.04283.

[10] Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., ... & Modha, D. S. (2015). Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. IEEE transactions on computer-aided design of integrated circuits and systems, 34(10), 1537-1557.

[11] Kästner, F., Janßen, B., Kautz, F., Hübner, M., & Corradi, G. (2018, May). Hardware/software codesign for convolutional neural networks exploiting dynamic partial reconfiguration on PYNQ. In 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (pp. 154-161). IEEE.

[12] Sharma, H., Park, J., Suda, N., Lai, L., Chau, B., Chandra, V., & Esmaeilzadeh, H. (2018, June). Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural network. In 2018 ACM/IEEE 45th Annual International Symposium on Computer

Architecture (ISCA) (pp. 764-775). IEEE.

[13] Blelloch, G. E. (1996). Programming parallel algorithms. Communications of the ACM, 39(3), 85-97.

[14] Geng, T., Wang, T., Sanaullah, A., Yang, C., Patel, R., & Herbordt, M. (2018, August). A framework for acceleration of CNN training on deeply-pipelined FPGA clusters with work and weight load balancing. In 2018 28th international conference on field programmable logic and applications (FPL) (pp. 394-3944). IEEE