Research on Dynamic Reconfiguration Technology of Neural Network Accelerator Based on Zynq

Hao Lv1,2, *, Shengbing Zhang1, Xiaojian Liu2, Shuo Liu2, Yongqiang Liu2, Wei Han2 and Shaowei Xu2

1School of Computer Science and Engineering, Northwestern Polytechnical University, Xi’an, China
2Xi’an Aeronautics Computing Technique Research Institute, Aviation Industry Corporation of China, Xi’an, China

*Corresponding author e-mail: lvhao20032003@qq.com

Abstract. Target detection based on convolutional neural network is a research hotspot in the field of computer vision. Conventional neural network (CNN) accelerators use the time division multiplexing method, and different network layers use the same accelerator, and their adaptability and resource utilization are not high. How to combine the dynamic reconfigurable characteristics of FPGA so that the calculation of each layer can be matched with the corresponding accelerator architecture at the cost of a certain configuration delay, and to improve the utilization efficiency of computing resources is a research hotspot. This article takes the YOLOv2 target detection algorithm widely used in the industry as an example, and uses Xilinx’s Zynq as the platform to describe the process of mapping the CNN model to the FPGA. Combined with the dynamic reconfigurability of FPGA, the calculation of each layer can be matched with the reconstructed accelerator architecture at the cost of a certain configuration delay, and the reconstruction delay can be shared by batch data multiplexing accelerator architecture, which effectively improves In order to improve the accelerator performance, the convolutional layer and the cascaded maximum pooling layer are merged to reduce the memory access delay. Experiments and evaluations were carried out on the accelerator architecture combined with dynamic reconfigurable characteristics, and the performance of 30.35GOP/s was obtained on the Zynq platform. Provide a reference for the application and optimization of CNN on embedded platforms.

Keywords: Reconfiguration, FPGA, Zynq, CNN

1. Introduction
In recent years, deep learning algorithms represented by Convolutional Neural Networks have made great breakthroughs in many computer vision tasks, such as image classification, target detection, image quality enhancement, etc. [1-2]. However, with the improvement of recognition rate, the computational complexity and memory requirements of deep learning algorithms have also increased sharply. Current general-purpose processors cannot meet the computational requirements of multi-
objective constraints such as power consumption, size, performance, and real-time performance. In the real-time application field of embedded platforms, storage space and computational overhead have become obstacles to the development of convolutional neural networks. Compared with GPUs, FPGAs have low power consumption and low latency, suitable for small batch streaming applications [3], and are very suitable for inference phase. Compared with ASIC, FPGA can re-change the hardware structure through configuration and has reconfigurable characteristics. FPGA is a very potential choice.

The current research on FPGA-based neural network accelerators mainly focuses on matrix multiplication optimization, network optimization, compression and encoding, bandwidth and architecture optimization, etc. Because the dynamic reconfigurability of FPGA requires delay consumption, there are not many studies on the combination of CNN and this aspect. Venieris SI. et al. proposed fpgaConvNet: Analyze the network structure of a given CNN model and map it to the FPGA. The highlight is that for the first time the possibility of combining the dynamic reconfigurability of FPGA and the CNN accelerator is proposed. Although the entire FPGA is reconstructed every time, which greatly increases the reconstruction delay, it is still of great significance for the study of combining the dynamic reconfigurable characteristics of FPGA with the neural network accelerator [4-5]. Kanstner F et al. are based on FPGA-based partial reconfigurable technology to reduce configuration file size and configuration delay, and at the same time accelerate CNN forward inference through software and hardware coordination. When performing FPGA logic configuration, the CPU side is also performing calculations to cover the configuration delay. Compared with pure CPU, the performance is improved by 1.54-2.24 times. Although the reconstruction logic only accounts for close to 1/3 of the total logic, the configuration delay still takes about 120-150ms, and the cost of the reconstruction delay is still very high.

In this paper, combined with Xilinx's Zynq FPGA+CPU heterogeneous platform, taking the YOLOv2 target detection model as an example, a CNN accelerator is designed, and the complete process of mapping the CNN model to the FPGA is introduced. Then combined with FPGA dynamic reconfigurable technology, the FPGA dynamic reconfigurable CNN accelerator is further designed and evaluated. Discuss the possibility and main problems of the dynamic reconfigurable characteristics of FPGA and CNN accelerator. The cost of reconstruction delay is shared by the batch data multiplexing accelerator architecture, the convolutional layer and the cascaded maximum pooling layer are combined to reduce the memory access delay, and the accelerator architecture combined with dynamic reconfigurability is experimented and evaluated.

2. Algorithm Model

2.1. YOLOv2 Neural Network Algorithm

The convolutional neural network (Convolutional Neural Network, CNN) structure model is relatively fixed, usually including an input layer, an output layer and several hidden layers, among which the hidden layer consists of several convolutional layers, pooling layers, normalization layers and activation layers One or more of them are cascaded. Facebook's AI Lab proposed a lightweight model YOLO [6] (You Only Look Once) based on regression methods. YOLO only uses a neural network to complete end-to-end target detection. The target detection is regarded as a regression problem, and the target detection frame and the corresponding class probability are obtained. Currently on Titan X GPU, YOLO detects images at a speed of 45 frames per second, which can be used for real-time detection. The fastest version of fast YOLO uses a simpler convolutional neural network, and the test speed reaches 155 frames per second.

YOLOv2 was proposed by Redmon J in 2016. It incorporates a series of the most advanced algorithms and processing methods in the field of target detection and recognition, such as standardized operations, joint training methods, and multi-scale data set expansion. It is comparable in detection speed and accuracy. Compared with SSD, it has been further improved. Figure 1 is a simplified YOLOv2 model network structure.
2.2. Zynq and Reconfigurable Computing

The hardware platform used in this paper is based on Xilinx’s zynq-7000soc [7]. Zynq series connects the traditional embedded ARM + peripheral controller (PS) and FPGA logic (PL) through some interfaces and interconnection routes. Therefore, developers can design acceleration logic corresponding to specific application in PL side, which is controlled by arm, and computationally intensive tasks are executed by PL side, and PS side is mainly responsible for control and interrupt processing. The PL side FPGA of zynq has dynamic reconfigurability, which can be controlled by PS side.

Through the reconfiguration of FPGA (write configuration stream file to FPGA through specific port), the digital logic in FPGA can be changed to make it suitable for new applications. This reconfiguration of the whole FPGA is called full reconfiguration. Zynq has the characteristic of partially reconfigurable (PR). In logic design, FPGA can be divided in advance, and the design is placed in the divided FPGA area. After the layout and routing is completed, the partial configuration stream file that only configures the FPGA area is obtained. This configuration method is called partial reconfigurable.

After full reconfiguration of FPGA with complete configuration stream file, the partition logic can be modified through partial configuration stream file without affecting the logic running on other regions. However, both full reconfiguration and partial reconfiguration have the cost of configuration delay in reconfiguration of FPGA. In the stage of FPGA configuration, FPGA itself can not work, which leads to the difficulty of adding dynamic reconfigurable features to FPGA design which is very sensitive to time delay.

3. Research and Design of Reconfigurable Accelerator

A typical accelerator processes each layer in turn according to the network sequence. A single accelerator on FPGA can only process a certain layer at a time. The input feature image pixel blocks and convolution kernel weights are read from the off chip storage, and then processed by PE array to
the final result, write it back out of the film. For a given network, each layer in the network is mapped to the same accelerator in sequence, which is time division multiplexing of the same accelerator. At this time, the hardware logic of FPGA is initialized only once, and then remains unchanged. Due to the different "shapes" of the dimensions of each layer, when mapping different convolution layers to the same architecture, the dynamic utilization of computing cells is often low. Figure 2 shows a typical accelerator without dynamic reconfiguration.

The reconfigurable accelerator is different from the typical accelerator. Considering the complexity of the current neural network structure and the different convolution "shape" of each layer, the whole network is generally divided into several subgraphs (which can be simplified as strings). The row cascade structure or single-layer accelerator structure can also be customized for various complex structures [8], which greatly improves the dynamic utilization of the calculation unit for each subgraph. The accelerator architecture of each subgraph can be limited to serial cascade structure and single-layer accelerator structure. In fact, it is a simple flow structure, that is, only the first layer and the last layer have read-write interaction with the outside of the chip, and the results of the intermediate layer are retained on the FPGA chip, and the final results are transmitted to the next layer after calculation. If the convolution parameters are small enough, they can be stored in the corresponding layer accelerator, otherwise additional memory access is needed.

In addition, since accelerators are customized for each subgraph, each subgraph has a corresponding configuration stream file (under some conditions, the accelerator architecture of the previous subgraph will be used directly, and no refactoring is required at this time).

Compared with typical accelerators, the advantages of reconfigurable accelerators are obvious: under the given hardware system conditions, each subgraph can make full use of hardware resources, which greatly improves the performance of each subgraph. In addition, when hierarchical streaming architecture is used, each layer can interact with each other directly on FPGA chip without memory access, which greatly reduces memory access delay and power consumption. Figure 3 shows the relationship between reconfigurable accelerator and multi mapping.

![Reconfigurable accelerator and multi mapping](image)

Figure 3. Reconfigurable accelerator and multi mapping

There are four kinds of layers in yolov2 network: convolution layer, pooling layer, routing layer and reordering layer. Convolution layer takes up most of the computation of the model, and existing research shows that it is very effective to accelerate convolution with FPGA [9]. Pooling is similar to convolution, using comparison instead of multiplication and addition, so the processing of convolution layer and pooling layer is put on FPGA. The routing layer can be realized by setting the memory offset address in advance. The reordering layer is also processed on the FPGA side.

In addition to the routing layer, the output of the previous layer is taken as the input of the current layer, and the output result is taken as the input of the next layer, which is processed sequentially. After the function of routing layer is realized by setting access address offset in advance, FPGA
accelerator only needs to read, process and write data back to memory according to relevant parameters.

Each time multiple blocks of data are read or written, the next read is performed after multiplexing the data blocks cached on FPGA chip, so as to ensure that each block of data is read only a small number of times from outside the chip, so as to reduce the number of memory accesses and the amount of data transferred. For each block data fetched, the control flow processing: read the block data out of the chip to the input cache, and send it to the conv, pool and reorg modules according to the layer category. The processed data is written back to the output buffer and written back to off chip storage after the final result is obtained.

The accelerator has several Axi master interfaces and an axilite slave interface. Read and write the control, data and status registers through the axilite slave interface. Multiple Axi master interfaces simultaneously read input characteristic graph and write back output characteristic graph, and one Axi master interface is responsible for reading the weight parameters of each layer.

4. Testing and analysis

4.1. Experimental environment

This paper takes AES and 3DES as examples to integrate with the design, tests the reconstruction process of encryption and decryption algorithm, analyzes the relevant indicators of the reconstruction process, and verifies the correctness of using different encryption and decryption algorithms to process data. Zynq-xc7z020 is used to build the hardware test platform, dual core arm-a9, clock frequency 667MHz, memory 1GB. Using Logitech c210 camera, the maximum resolution is 640×480, and the maximum is 30F/s.

The acquired image is obtained from USB camera and stored in memory. After preprocessing by arm, it is handed over to yolov2 accelerator on FPGA for target detection. The detected data is still stored in memory. After arm post-processing, the image with detection category and position is written back to an address in memory, and sent to the HDMI controller on the FPGA side to display on the display screen [10]. Figure 4 shows the structure of the test platform.

The verification platform is divided into two parts: PL module and PS system. Use vivado HLS 2018.2 HLS for accelerator design, vivado v2018.2 synthesis and layout. Firstly, the hardware system and hardware modules are built by vivado, and the bit netlist file is generated. Then, it is packaged into hardware package and imported into vivado SDK. Then, the PS terminal project is established based on the hardware environment of bit file. Then, the convolution kernel parameter file and H header file after fixed-point number conversion are imported into the SDK to realize the construction of the whole verification platform.

![Test platform structure of target detection system](image-url)
4.2. Performance Evaluation

The yolov2 model consists of 23 convolution layers, 5 pooling layers and 1 rearrangement layer. Because of the fusion of the largest pooling layer and convolution layer, there are 25 layers (4 fusion layers, 19 convolution layers, 1 pooling layer and 1 reordering layer). The main error of this yolov2 model is still concentrated in the transmission delay, which is about 10%. The total output delay is about 99.504ms, which is far less than the total input delay (634.379ms).

CPU platform: server CPU Intel Xeon e5-2620 V4 (8 cores) working frequency 2.1GHz [11]. As shown in Table 1, compared with CPU, CPU + FPGA heterogeneous system is about 86 times more energy efficient than dual core arm-a9 and 120.4 times higher than Xeon. The speed is 112.9 times of dual core arm-a9 and 7.3 times of Xeon. At this time, the power consumption is low, but due to the weak computing power, the calculation delay is too long and the energy efficiency is low. Xeon, on the contrary, has strong computing performance, which makes the computing delay very short. However, due to the high power consumption, it leads to low energy efficiency. Different from the former two, the power consumption of CPU + FPGA heterogeneous system is slightly higher than that of dual core arm-a9. However, with FPGA acceleration, the computing performance is greatly improved, and good performance in computing delay and energy efficiency is achieved.

| Device | CPU E5-2620v4 (8 cores) | CPU ARM-A9 (dual-core) | CPU+FPGA XC7Z020 |
|--------|-------------------------|------------------------|------------------|
| Technology (nm) | 14 | 28 | 28 |
| Clock (Hz) | 2.1G | 666.67M | 666.67M+150M |
| Memory (MB) | 257842 | 512 | 512 |
| Peak System Power(W) | 85 | 3.96 | 5.21 |
| Performance (GOP/s) | 4.11 | 0.27 | 30.35 |
| Latency per img (s) | 7.17 | 110.36 | 0.99 |
| Power Efficiency (GOP/s/W) | 0.05 | 0.07 | 5.83 |

Based on the same zynq-xc7z020 platform, a test comparison scheme is set up, and FPGA is adopted to accelerate core time-sharing multiplexing (ARM program call), so it is not applicable to reconfigurable technology. As shown in Table 2, the FPGA based dynamic reconfigurable accelerator allocates the reconstruction delay effectively by processing batch data, which greatly improves the throughput. However, in terms of the delay of processing each image, the single-layer accelerator architecture still has the advantage because the reconstruction delay is too high. Therefore, in practical application, we should analyze the specific problems, explore and design the appropriate accelerator architecture combined with the current application situation.

| CNN model | Non dynamic reconfigurable | Dynamic reconfigurable |
|-----------|---------------------------|------------------------|
| FPGA Board | XC7Z020 | XC7Z020 |
| Clock(MHz) | 150 | 150 |
| Precision | Fixed-16 | Fixed-16 |
| DSP(Used/Total) | 153/220 | 211/220 |
| Operations(GOP) | 29.47 | 29.47 |
| Latency(s/per img) | 0.98 | 1.12 |
| Throughput(GOP/s) | 30.35 | 33.49 |
5. Conclusion
The common CNN accelerator is based on time-sharing multiplexing technology, and the whole process of calculation calls the same FPGA acceleration core, while the dynamic reconfigurable accelerator dynamically reconstructs between different subgraphs based on the partition of model task graph. The whole process of designing dynamic reconfigurable accelerator is given in this paper. Taking YOLOv2 as an example, the reconfigurable accelerator is designed and evaluated. The reconfiguration delay is shared by using the accelerator architecture of batch data time-sharing and each subgraph. The memory access delay is reduced by fusing the cascaded maximum pooling layer and convolution layer, which effectively improves the performance of the neural network accelerator based on FPGA dynamic reconfiguration.

References
[1] RUSSAKOVSKY O, DENG J, SU H, et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 2015, 115 (3): 211-252.
[2] LIN T Y, MAIRE M, BEONGLONGE S, et al. Microsoft coco: common objects in context. European Conference on Computer Vision. Springer, Cham, 2014: 740-755.
[3] CHOUDHURY T, CLARKSON B, JEBARA T, et al. Multimodal person recognition using unconstrained audio and video. AVBPA 99, Washington DC, 1999: 176-181.
[4] Venieris S I, Bouganis C S. fpgaConvNet: A framework for mapping convolutional neural networks on FPGAs. 2016 IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). Piscataway: IEEE, 2016: 40-47.
[5] Venieris S I, Bouganis C S. Latency-driven design for FPGA-based convolutional neural networks. 2017 27th International Conference on Field Programmable Logic and Applications (FPL). Piscataway: IEEE, 2017: 1-8.
[6] J. Redmon, S. Divvala, R. Girshick, et al. You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016.
[7] Yan MinJia. (2019) Design and Implementation of CNN Object Detection System on Embedded Platform. Master’s Thesis, University of Electronic Science and Technology of China.
[8] Venieris S I, Bouganis C S. fpgaConvNet: Mapping Regular and Irregular Convolutional Neural Networks on FPGAs. IEEE transactions on neural networks and learning systems, 2018 (99): 1-17.
[9] Guan Y, Liang H, Xu N, et al. FP-DNN: An automated framework for mapping deep neural networks onto FPGAs with RTL-HLS hybrid templates. 2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). Piscataway: IEEE, 2017: 152-159.
[10] Chen Chen, et al. Design and implementation of FPGA-based deep learning object detection system. Plication of Electronic Technique, 2019.
[11] Chen Chen. Research of Scalability on FPGA-based Neural Network Accelerator. Master’s Thesis, Jiangnan University, China, 2019.