Research on the Architecture of Convolutional Neural Network Accelerator

YingLiu*, HaitaoDing, JialiMa

*Information Engineering University, No.62, KexueAvenue, Zhengzhou450001, China

liuying930408@163.com

Abstract. Convolutional neural networks are widely used in human production and life, but due to their large amount of calculation and complex calculation mode, their calculation speed is slow, so it is necessary to design a dedicated hardware accelerator. This paper firstly analyzes the algorithm of the convolutional neural network and decomposes the algorithm into multiple basic operations. For the convolution operation with the largest amount of calculation and complex operation mode, a near calculation storage array is designed according to its operational characteristics. Furthermore, a convolutional neural network accelerator architecture is proposed to realize the fast operation of a convolutional neural network.

Keywords: Convolutional Neural Network, Hardware Design, Algorithm Acceleration.

1. Introduction
In recent years, CNN (Convolutional Neural Networks) has developed rapidly and has been widely used in image classification[1], face recognition[2], and speech recognition[3]. The vigorous development of CNN is not only related to its own excellent classification effect but also benefited from the development of semiconductor technology and cloud computing. The computing power has been greatly improved and computing resources are more easily obtained.

However, with the continuous deepening of research, CNN has introduced some problems while its classification accuracy has been continuously improved[4]. The network topology has become increasingly complex, and the amount of calculations and parameters of the algorithm has also increased[5]. These large models are becoming difficult to handle. At the same time, the development of semiconductor devices has gradually slowed down, and the increase in computing power of traditional CPUs cannot keep up with the expansion of neural network parameters.

In order to better apply CNN, it is necessary to design a dedicated convolutional neural network accelerator. FPGA is a kind of semi-custom integrated circuit, which provides many dedicated arithmetic calculation units, logic module resources, and on-chip storage resources on the chip. Through the hardware programming connection resources to conFig the organization and unite into a complete circuit, the programming is flexible and the product design cycle is relatively short. This article chooses to use FPGA as the acceleration device. Based on the analysis of the operational characteristics of convolutional neural networks, a convolutional neural network accelerator architecture is designed to realize the fast operation of convolutional neural networks.
1.1. Research on a convolutional neural network algorithm

Convolutional neural network algorithm is an algorithm with huge computational complexity and high computational complexity[6]. It includes a convolutional layer, pooling layer and fully connected layer[7]. These calculation layers have the same calculation parts, such as offset addition and activation function calculations, as well as their own unique calculations. In order to better accelerate the convolutional neural network algorithm in hardware, it is necessary to study the implementation details of each calculation layer.

(1) Convolutional layer

Convolutional layer operations include convolution operations, offset addition operations and activation functions, which are executed in sequence. The operation process is shown in Fig 1.

Convolution operation is a linear filtering method commonly used in image processing, which can perform de-noising and sharpening effects on images. For a two-dimensional image and a two-dimensional convolution kernel filter matrix, the definition of the convolution operation is shown in equation (1):

\[ S(i, j) = (I * K)(i, j) = \sum_{m} \sum_{n} I(m,n)K(i-m, j-n) \]

For each pixel in the picture, calculate the product of its neighboring pixels and the corresponding position of the filter matrix, and then add the product as the output value of this pixel position. At present, many convolutional neural networks use a 3×3 convolution kernel. The convolution kernel slides through the position of each pixel in the picture in turn, and performs a 9×9 multiplication and accumulation operation. The whole process is shown in Fig 2.
In order to ensure sufficient feature extraction in a convolutional neural network, multiple convolution kernels need to be used, and each convolution kernel generates a feature map, which is a different channel of the picture after filtering. The number of input channels of each layer is the number of output channels of the previous layer. If there is more than one input channel, the results obtained from different input channels need to be superimposed to obtain the result of one output channel. Fig. 3 shows how to perform multi-channel convolution in the convolution layer of the convolutional neural network.

![Multi-channel convolution operation](image1)

**Fig.3 Multi-channel convolution operation**

The offset addition operation is performed after the convolution operation. The two-dimensional feature map of each output channel is added to the corresponding offset data. One matrix is the feature map matrix and the other is the offset matrix.

The activation function operation is performed after the offset addition operation is completed. The ReLU function is currently the activation function operation used by most convolutional neural networks. Here, the realization of the ReLU function is considered, and its expression is shown in equation (2):

$$f(x) = \max(0, x)$$  \hspace{1cm} (2)

Analyzing the above three calculation operations, it can be obtained that the basic operations of the convolutional layer include: $9 \times 9$ multiplication and accumulation operations, addition operations, and compared with zero operations.

**2) Pooling layer**

The operation of the pooling layer is mainly a pooling operation. The pooling operations currently used by convolutional neural networks include max-pooling and mean-pooling. The pooling window has a size of $2 \times 2$, as shown in Fig. 4.

![Pooling operation](image2)

**Fig.4 Pooling operation**

The two pooling operations are actually taking the maximum value of the data in the pooling window and taking the average value of the data in the pooling window. The size of the pooling window is fixed. The pooling window includes four data, so these four data can be processed directly
when the pooling operation is performed, that is, the basic operation of the pooling layer is the maximum value of four elements and the average value of four elements.

(3) **Fully connected layer**

The operations of the fully connected layer include fully connected operations, offset addition, and activation function operations. In the fully connected operation, the output neuron and the input neuron are all connected with independent weight values, as shown in Fig.5.

![Fig.5 Fully connected operation](image)

The fully connected operation is essentially a matrix-vector multiplication operation, so the multiply-accumulate operation can be used as its basic operation. After the full connection operation is over, the offset addition operation and the activation function operation are also required, which is the same as the convolutional layer operation.

In summary, the analysis of the convolutional neural network algorithm can be decomposed into a series of basic operations: 9×9 multiplication and accumulation operations, addition operations, compared with zero, multiplication and accumulation operations, four-element maximum and four-element average. We will design a convolutional neural network accelerator based on the characteristics of these operations.

### 2. Convolution neural network accelerator architecture

In order to make better use of hardware resources and achieve the optimal performance of convolutional neural network accelerators, we design a dedicated hardware computing unit for the dedicated computing part to perform corresponding operations and design a general computing module for the general computing part to reuse between different computing layers. According to the previous analysis of the convolutional neural network algorithm, the design of the convolutional neural network accelerator framework is shown in Fig.6.

![Fig.6 Convolutional neural network accelerator overall architecture](image)
Control unit is the core of the accelerator and is responsible for controlling the execution process of the entire accelerator. According to the requirements of the convolutional neural network algorithm, it selects a certain functional component in the accelerator or combines several functional components to complete the corresponding function.

BRAM is the cache unit of the accelerator layer, responsible for the reading of weight data and bias data from off-chip storage to the on-chip arithmetic unit and the writing back of on-chip arithmetic results to off-chip memory.

Registers directly participate in the operation and play a function of selecting data in the BRAM.

Near calculation storage array is a storage structure designed according to the characteristics of the convolution operation, which can simultaneously realize the function of caching the input feature map data and the function of selecting the data involved in the calculation.

Multiply-accumulate tree is used to complete the $9 \times 9$ multiply-accumulate operation in the convolution operation. The basic operation of the convolution operation with the largest amount of calculation in the convolution neural network algorithm is the multiplication and accumulation operation, and the multiplication and accumulation tree operation unit is responsible for completing this part of the operation.

Integrated operation unit is used for other operational functions in the convolutional neural network, including addition operation, compared with zero, multiplication and accumulation operation, four-element value average, and four-element value maximum. These functional components are integrated into the comprehensive operation unit and combined with the multiply-accumulate tree unit to complete the operation process of the convolutional neural network.

3. Research and design of near calculation storage array
In the convolutional neural network algorithm, the convolution operation has the largest amount of calculation and the operation mode is the most complicated. The acceleration of this part of the operation is very important. The near calculation storage array designed in the accelerator architecture can make good use of the corresponding computing requirements. This chapter will introduce the method of the near calculation storage array participating in the convolution operation.

The convolution operation needs to select a sliding window in the feature map, and the sliding window slides on the feature map according to certain step size. Based on this observation, this paper proposes a new near calculation storage array structure. The register group is used as the feature map storage unit for buffering the input feature map data. The near calculation storage array realizes the sliding window to move at different lengths through a shift operation. The internal structure of the near calculation storage array is shown in Fig.7.
According to the operation logic of the convolution operation, it is known that the convolution window needs to slide horizontally on the feature map. In order to reduce the data selection of the near calculation storage array, the shift operation can be used to realize the sliding of the convolution window. Specifically, perform a horizontal left shift operation on the shift operation module (the number of bits shifted to the left is equal to the number of steps), and update the value of the register in the shift operation module to the value of the right register. In this way, the value of the leftmost 3×3 register group is updated to the value of the new convolution window.

When the convolution window moves to the end of the line, a line-feed operation is required. Corresponding to the operation in the near calculation storage array, the shift operation module and the memory module jointly perform a vertical upward shift operation (the number of bits shifted up is equal to the number of steps). The value of the three rows of registers in the shift operation module after the shift operation will be updated to its next row of data.

4. Accelerator execution flow
The operations that the convolution layer needs to perform include convolution operation, offset addition operation, and activation operation. The input data includes the input feature map data stored on the near calculation storage array and the convolution kernel weight data and offset stored on the BRAM unit. Data and temporary storage result data, the calculation result is written back to the temporary storage result data interval in the BRAM. The implementation in the accelerator is shown in Fig.8.
The operation that the pooling layer needs to perform is the average of the four-element value or the maximum of the four-element value. The input data is the input feature map data on the near calculation storage array, and the calculation result is stored in the result data interval of the BRAM. The implementation in the accelerator is shown in Fig.9.

The operations that the fully connected layer needs to perform include multiplication and accumulation operations, bias addition operations, and activation operations. The input data is one-dimensional input vector data, two-dimensional weight data, offset data, and temporary storage result data stored on the BRAM cell. The result is written back to the temporary storage result data interval in the BRAM, and the implementation in the accelerator is shown in Fig.10.

5. Conclusion
In recent years, the development of artificial intelligence technology represented by deep learning is in full swing. Among them, convolutional neural networks have received extensive attention due to their powerful recognition effects on image data. But the good performance of the convolutional neural network is based on its huge amount of calculation, and the computing power is the key to the practical application of today’s convolutional neural network. Starting from the convolutional neural network algorithm, this article studies the implementation details of each of the operational layers. On this basis, a hardware accelerator architecture that meets the characteristics of the convolutional neural network algorithm is designed, which accelerates the operation of the convolutional neural network and can better serve the human society.

References
[1] Hu J, Shen L and Sun G 2018 Squeeze-and-excitation networks Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition pp7132-7141
[2] Collobert R , Weston J and Bottou L 2011 Natural Language Processing (Almost) from Scratch Journal of Machine Learning Research 12 pp2493-2537
[3] Abdel-Hamid O , Mohamed A R and Jiang H 2014 Convolutional Neural Networks for Speech Recognition IEEE Transactions on Audio Speech & Language Proc 22 pp1533-1545
[4] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks Advances in Neural Information Proc. Systems pp1097-1105
[5] Silver D , Huang A and Maddison C J 2016 Mastering the game of Go with deep neural networks and tree search J. Nature 529 pp484-489
[6] Simonyan K and Zisserman A 2014 Very Deep Convolutional Networks for Large-Scale Image Recognition (Preprint arXiv:1409.1556)
[7] Szegedy C, Liu W and Jia Y 2015 Going deeper with convolutions Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition pp1-9